```
?USER = SITE

        PACKET   335
        INPUT     13 CARDS FROM CRA
        TIME    1559
        DATE   77280 FRIDAY, 10/07/77



 *** BURROUGHS B5700 DCMCP MARK XVI.0.178  AND INTRINSICS MARK XVI.0.132 ***

     #NO MESSAGES TODAY

15:59:44  ?USER= SITE
15:59:44  ?EXECUTE PATCH/MERGE
15:59:44  ?DATA CARD
15:59:45          4:PATCH/MERGE= 1 BOJ 1559 05/12/77
15:59:46          CDA IN CARD:PATCH/MERGE= 1
15:59:47          PRD0336 OUT 011 LINE:PATCH/MERGE= 1
15:59:48          DKA OUT SER MASTERF SITE:PATCH/MERGE= 1
15:59:49          DKA IN SER PATCH ALGOL:PATCH/MERGE= 1
15:59:49          DKA OUT SER PATCHDE SITE:PATCH/MERGE= 1
16:00:24          CDA REL CARD:PATCH/MERGE= 1
16:00:24  ?END
16:00:55          DKA REL PATCHDE SITE:PATCH/MERGE= 1
16:00:55          DKA REL PATCH ALGOL:PATCH/MERGE= 1
16:00:56          PRD0336 REL 011 LINE 1471:PATCH/MERGE= 1
16:00:56          DKA REL MASTERF SITE:PATCH/MERGE= 1
16:00:57          PATCH/MERGE= 1 EOJ 1600
16:00:58          FOR PATCH/MERGE= 1: PROCESS= 54 SECS, IO= 36 SECS, OLAY= 0
16:00:58          PKT#0335 REMOVED
```

```
 LABEL  000000000LINE   00177280?EXECUTE PATCH/MERGE                                                    PATCH  /MERGE
                BURROUGHS B-5700 PATCH/MERGE PROGRAM MARK XVI.0.104     FRIDAY, 10/07/77,   3:59 PM.



**********      INPUT      ********************************************************************************

$@CARD MERGE ZIP LISTI                                                          C       0000000/CARD
$, 22 PATCHES FOR ALGOL                                                         C

                                CARD INPUT IS CARD
                                PATCHES/ALGOL   IS NOT ON DISK
                                PATCH  /ALGOL   WILL BE MERGED

$*COMPILE ALGOL/NUDISK ALGOL LIBRARY                                            C
$*ALGOL STACK = 1000                                                            C
$*ALGOL FILE TAPE = SYMBOL/ALGOL DISK SERIAL                                    C
$*FILE LINE = LINE PRINT OR BACK UP                                             C
$*DATA CARD                                                                     C
$=  ***** YHIS LISTING SHOWS PATCHES IMPLEMENTED AT UCSC  *****                 C
$SET TAPE LIST SINGLE FORMAT PRT XREF                                           C

$#PATCH NUMBER 101 FOR ALGOL CONTAINS 2 CARDS                                   P  101 PATCH  /ALGOL

      IF T = INFO[GT1, GT2] THEN BEGIN                          02877010      P  101
          T:= 0; GO TO COMPLETE END;                            02877020      P  101
$;THIS PATCH ELEMINATES A COMPILER LOOP CAUSED WHEN A FORMAL PARAMETER          P  101
$; IN A PROCEDURE DECLARATION IS NOT INDICATED IN  THE SPECIFICATION           P  101
$; LIST.                                                                       P  101
$;*************************************************************************    P  101


$#PATCH NUMBER 102 FOR ALGOL CONTAINS 1 CARD                                    P  102

   IF BUP ≠ BUP:=BUP^TAKE(BUP + 1),PURPT THEN                   14088000      P  102
$; THIS PATCH ELEMINATES A COMPILER  LOOP CAUSED WHEN THE FORMAL                P  102
$; PARAMETER IN A PROCEDURE DECLARATION IS FOLLOWED BY A COMMA.                 P  102
$;*************************************************************************    P  102


$#PATCH NUMBER 103 FOR ALGOL CONTAINS 2 CARDS                                   P  103

FOUND;                                                          02315000      P  103
```

```
     IF OPINX +1>OPARSIZE THEN FLAG(602) ELSE % TOO MANY USER OPTIONS   02316000  P  103
$: THIS PATCH CORRECTS AN INVALID INDEX CONDITION CAUSED WHEN TOO MANY             P  103
$: USER OPTIONS HAVE BEEN SPECIFIED.                                               P  103
$:*************************************************************************       P  103



$#PATCH NUMBER 104 FOR ALGOL CONTAINS 4 CARDS                                      P  104

FLAG (120);                                                          07025000  P  104
FCR:= (LCR:=MKABS(CBUFF[9]))-9;                                      07025010  P  104
IF LISTER THEN PRINTCARD;                                            07025020  P  104
FCR:= (LCR:=MKABS(TBUFF[9]))-9  END;                                 07025030  P  104
$:THIS PATCH CORRECTS A PROBLEM WHERE A PATCH CARD IS LOST WHEN BEGIN               P  104
$:END PAIRS ARE NOT MATCHED AND PATCH CARD SEQUENCE NUMBERS ARE GREATER            P  104
$:THAN THE SEQUENCE NUMBER OF THE "END."CARD IN THE SOURCE FILE.                   P  104
$:*************************************************************************       P  104



$#PATCH NUMBER 105 FOR ALGOL CONTAINS 13 CARDS                                     P  105

BEGIN                                                                02198755  P  105
LABEL ENDREADTAPE, EOFT;                                             02198760  P  105
READ (TAPE, 10, TBUFF[*])[EOFT];                                     02201750  P  105
MAXLCR:=LCR:=MKABS(TBUFF[9]);                                        02202000  P  105
GO TO ENDREADTAPE;                                                   02202010  P  105
EOFT:                                                                02202020  P  105
DEFINEARRAY[25]:="ND;END."& "E"[1:43:5];                             02202030  P  105
DEFINEARRAY[34]:="9999" & "9999"[1:25:23];                           02202040  P  105
TLCR:= MKABS(DEFINEARRAY[34]);                                       02202050  P  105
PUTSEQNO (DEFINEARRAY[33],TLCR-8);                                   02202060  P  105
TURNONSTOPLIGHT("%", TLCR-8);                                        02202070  P  105
ENDREADTAPE:                                                         02202080  P  105
END;                                                                 02202090  P  105
$:THIS PATCH CORRECTS AN EOF NO LABEL ENCOUNTERED WHEN THE SOURCE                  P  105
$: "END." CARD IS PATCHED OVER AND THE PATCH DECK CONTAINS CARD SEQUENCE           P  105
$:NUMBERS GREATER THAN THE SEQUENCE NUMBER OF THE "END." CARD IN THE               P  105
$:SOURCE FILE.                                                                     P  105
$:*************************************************************************       P  105



$# PATCH NUMBER 106 FOR ALGOL CONTAINS 114 CARDS.                         00000001  P  106

        CODEFILEBIT = 29#,                                               01001172  P  106
        USEROPINX   = 30#;                                               01001173  P  106
        CODEFILE    = OPTIONWORD,[CODEFILEBIT:1]#,                       01001463  P  106
                      LENGTH6,LENGTH7,LENGTH8,LENGTH9,WHATISIT;          02365000  P  106
LENGTH8;                                                                 02574000  P  106
    IF Q ="8CODEF" THEN                                                  02574100  P  106
        BEGIN SWITCHIT(CODEFILEBIT); GO AGAIN; END;                      02574200  P  106
    GO WHATISIT;                                                         02574300  P  106
INTEGER PROCEDURE MOVEANDBLOCK(FROM,SIZE,NAME);                          02927100  P  106
   VALUE SIZE,NAME; REAL SIZE,NAME; ARRAY FROM[0,0];                     02927110  P  106
```

```
BEGIN                                                                  02927120  P  106
INTEGER NSEGS,I,J,K;                                                   02927130  P  106
ARRAY A[0:14];                                                         02927140  P  106
SWITCH FORMAT FMT :=                                                   02927150  P  106
   (/,"FILE PARAMETER BLOCK IS CODE FILE SEGMENT",I5,/),               02927160  P  106
   (/,"SEGMENT DICTIONARY IS CODE FILE SEGMENT",I5,/),                 02927170  P  106
   (/,"PROGRAM-LINE DICTIONARY IS CODE FILE SEGMENT",I5,/),            02927180  P  106
   (/,"PROGRAM REFERENCE TABLE IS CODE FILE SEGMENT",I5,/),            02927190  P  106
   (/,"SEGMENT-LINE DICTIONARY IS CODE FILE SEGMENT",I5,/),            02927200  P  106
   (/,"POWER OF TEN ARRAY IS CODE FILE SEGMENT",I5,/),                 02927210  P  106
   (/,"SEGMENT ZERO",I*,/),                                            02927220  P  106
   (/,"SEGMENT NUMBER",I5," IS CODE FILE SEGMENT",I5,/));              02927230  P  106
STREAM PROCEDURE OCTALWORDS(N,W,S,D); VALUE N,W;                        02927240  P  106
   BEGIN                                                               02927250  P  106
   DI:=D; DS:="LIT" ";                                                 02927260  P  106
   SI:=LOC N; SI:=SI+6;                                                02927270  P  106
   4(DS:=3 RESET; 3(IF SB THEN DS:=SET ELSE DS:=RESET; SKIP SB));      02927272  P  106
   DI:=DI-4; DS:=3 FILL;                                               02927280  P  106
   DI:=D; DI:=DI+5; DS:=4 LIT" ";                                      02927290  P  106
   SI:=S;                                                              02927300  P  106
   W-2(8(DS:=3 RESET;                                                  02927310  P  106
         3(IF SB THEN DS:=SET ELSE DS:=RESET; SKIP SB);                02927320  P  106
        );                                                             02927330  P  106
       DS:="LIT" ");                                                   02927340  P  106
     DS:=2 LIT" ");                                                    02927350  P  106
   END OF OCTALWORDS;                                                  02927360  P  106
%**********     S T A R T     **********                                02927370  P  106
NSEGS:=(SIZE+29) DIV 30;                                               02927380  P  106
IF DA DIV CHUNK < T:=(DA+NSEGS) DIV CHUNK THEN                         02927390  P  106
   DA:=CHUNK*T;                                                        02927400  P  106
MOVEANDBLOCK:=DA;                                                      02927410  P  106
IF CODEFILE THEN                                                       02927420  P  106
   IF NAME≥0 THEN                                                      02927430  P  106
      WRITE(LINF,FMT[NAME],DA)                                         02927440  P  106
   ELSE                                                                02927450  P  106
      WRITE(LINF,FMT[7],ABS(NAME),DA);                                 02927460  P  106
IF SIZE≠0 THEN                                                         02927470  P  106
   BEGIN                                                               02927480  P  106
   FOR J:=0 STEP 30 WHILE J < SIZE DO                                  02927490  P  106
      BEGIN                                                            02927500  P  106
      IF (K:=(128-(J MOD 128))) < 30 THEN                              02927510  P  106
         BEGIN                                                         02927520  P  106
         MOVE(K,FROM[J DIV 128,J MOD 128],CODE(0));                    02927530  P  106
         MOVE(30-K,FROM[(J DIV 128)+1,0],CODE(K));                     02927540  P  106
         END                                                           02927550  P  106
      ELSE                                                             02927560  P  106
         MOVE(30,FROM[J DIV 128,J MOD 128],CODE(0));                   02927570  P  106
      IF J+30 > SIZE THEN % ZERO OUT UNUSED SECTION                    02927580  P  106
         BEGIN                                                         02927590  P  106
         K:=0;                                                         02927600  P  106
         MOVE(1,K,CODE(SIZE-J));                                       02927610  P  106
         IF (SIZE-J) < 29 THEN % MORE THAN ONE WORD                    02927612  P  106
            MOVE(29-SIZE+J,CODE(SIZE-J),CODE(SIZE-J+1));               02927620  P  106
         END;                                                          02927630  P  106
      IF CODEFILE THEN                                                 02927640  P  106
         BEGIN                                                         02927650  P  106
         FOR K:=0 STEP 5 WHILE K≤25 AND (J+K)≤SIZE DO                  02927660  P  106
```

```
                BEGIN                                                          02927670  P  106
                BLANKET(14,A);                                                 02927680  P  106
                OCTALWORDS(J+K,IF (J+K+5)≤SIZE THEN 5 ELSE                     02927690  P  106
                      SIZE-J-K,CODE(K),A);                                     02927700  P  106
                WRITE(LINE,15,A[*]);                                           02927710  P  106
                END;                                                           02927720  P  106
              WRITE(LINE);                                                     02927722  P  106
              END;                                                             02927730  P  106
            WRITE(CODE[DA]); DA:=DA+1;                                         02927740  P  106
            END;                                                               02927750  P  106
        END;                                                                   02927760  P  106
    END OF MOVEANDBLOCK;                                                       02927770  P  106
                    "8CODEF",0,          % 56,57                             09027058  P  106
$ VOIDT 09332001                                                               09319000  P  106
          IDARRAY[4]:=MOVEANDBLOCK(FDOC,GT1,0);                                09396000  P  106
          IDARRAY[0]:=MOVEANDBLOCK(SEGDICT,SGAVL,1);                           09399000  P  106
                                (LDICT,SGAVL,2)[18:33:15];                     09399150  P  106
          IDARRAY[2]:=MOVEANDBLOCK(PRT,PRTIMAX,3);                             09402000  P  106
          GT1:=DA; DA:=0; MOVE(30,IDARRAY[0],PRT[0,0]);                        09407000  P  106
          GT2:=MOVEANDBLOCK(PRT,30,6); DA:=GT1;                                09407010  P  106
          IF CODEFILE THEN WRITE(LINE);                                        09407020  P  106
PROCEDURE SEGMENT(SIZE,NO,NOO);                                                13634000  P  106
    VALUE SIZE,NO,NOO;                                                         13635000  P  106
    REAL  SIZE,NO,NOO;                                                         13636000  P  106
    BEGIN                                                                      13637000  P  106
    INTEGER DUMMY; % THIS IS HERE SO THAT OUR CODE SEGMENT                     13637100  P  106
                   % IS NOT TOO BIG                                            13637200  P  106
    PDPRT[PDINX.[37:5],PDINX.[42:6]] :=                                        13638000  P  106
         SIZE & NO[28:38:10] &                                                 13639000  P  106
         MOVEANDBLOCK(ENDOC,ABS(SIZE),-ABS(NO))[13:33:15] &                    13640000  P  106
         REAL(SAVEPRTOG)[3:47:1];                                              13641000  P  106
       PDINX:=PDINX+1;  SIZE:=ABS(SIZE);                                       13642000  P  106
       IF SIZE>SEGSIZEMAX THEN SEGSIZEMAX:=SIZE;                               13643000  P  106
       AKKUM:=AKKUM+SIZE;                                                      13644000  P  106
       IF SAVEPRTOG THEN AUXMEMREQ:=AUXMEMREQ+16*(SIZE.[38:6]+1);              13645000  P  106
       IF LISTER OR SEGSTOG THEN                                               13646000  P  106
            BEGIN                                                              13647000  P  106
            IF NOHEADING THEN DATIME;                                          13648000  P  106
            IF SINGLTOG THEN WRITE(LINE,PRINTSIZE,NO,SIZE,NOO)                 13649000  P  106
                 ELSE WRITE(LINE,PRINTSIZE,NO,SIZE,NOO);                       13650000  P  106
            END;                                                               13651000  P  106
       LDICT[NO,[38:3],NO,[41:7]] :=                                           13652000  P  106
            IF BUILDLINE THEN                                                  13653000  P  106
                 MOVEANDBLOCK(ENIL,ENILPTR+1,4) & SIZE[18:33:15]               13654000  P  106
            ELSE -1;                                                           13655000  P  106
       END OF SEGMENT;                                                         13656000  P  106
$ VOIDT 13696001                                                               13657000  P  106
$! BY KFK                                                                      99999900  P  106
$! DATE 1/22/76                                                               99999901  P  106
$! MSA CENTRAL - DETROIT                                                      99999902  P  106
$!    THIS PATCH ADDS A NEW DOLLARSIGN OPTION "CODEFILE".                      99999903  P  106
$! WHEN THIS OPTION IS SET AN OCTAL LISTING OF ANYTHING                        99999904  P  106
$! WRITTEN TO THE CODE FILE IS PRODUCED.  THIS IS A REWRITE OF                 99999905  P  106
$! THE "TABLES" PATCH THAT LISTED SEGMENT ZERO, THE PRT AND OTHER             99999906  P  106
$! ITEMS.  "TABLES" IS NO LONGER VALID AS A COMPILER DOLLARSIGN               99999907  P  106
$! CARD OPTION.  THIS PATCH IS A COMPILER DEBUGGING AID.                      99999908  P  106
$!*************************************************************************99999909  P  106
```

```
$#PATCH NUMBER 107 FOR ALGOL CONTAINS 142 CARDS.                                00000001  P  107

 612      INCLUDECARD; TOOMANY NESTED INCLUDES.                                00412000  P  107
 613      INCLUDECARD; MISSING FILE NAME ON INCLUDE CARD.                      00413000  P  107
 614      INCLUDECARD; ENDING SEQUENCE NUMBER MISSING.                         00414000  P  107
 615      INCLUDECARD; COPY MISSING ON INCLUDE CARD.                           00415000  P  107
 616      INCLUDECARD; MORE THAN ONE FILE NAME ON INCLUDE CARD                 00416000  P  107
 617      INCLUDECARD; + COPY CAN NOT BE USED UNLESS $ IS IN COLUMN ONE        00417000  P  107
DEFINE                                                                        01561500  P  107
    INSERTMAX = 20#, % CHANGE THIS IF YOU NEED MORE LEVELS OF INCLUDES        01561510  P  107
    INSERTCOP = INSERTINFO[INSERTDEPTH,4]#, % = 1 IF COPY TO NEWTAPE          01561520  P  107
    INSERTMID = INSERTINFO[INSERTDEPTH,0]#, % MFID OF THE LIBRARY FILE        01561530  P  107
    INSERTFID = INSERTINFO[INSERTDEPTH,1]#, % FID  OF THE LIBRARY FILE        01561540  P  107
    INSERTINX = INSERTINFO[INSERTDEPTH,2]#, % POINTER TO THE RECORD           01561550  P  107
    INSERTSEQ = INSERTINFO[INSERTDEPTH,3]#; % LAST SEQUENCE TO BE INCLUD       01561560  P  107
INTEGER SAVECARD, INSERTDEPTH;                                                01561570  P  107
ARRAY INSERTINFO[0:INSERTMAX,0:4];                                            01561580  P  107
FILE LIBRARYFIL DISK RANDOM(2,10,30);                                         01561590  P  107
DEFINE LF = LIBRARYFIL#;                                                      01561600  P  107
SAVE ARRAY LBUFF[0:9]; % INPUT BUFFER                                         01561610  P  107
REAL STREAM PROCEDURE CMPD(A,B);                                              01561620  P  107
    BEGIN                                                                     01561630  P  107
    SI:=A; DI:=B;                                                             01561640  P  107
    IF 8 SC ≥ DC THEN                                                         01561650  P  107
        BEGIN                                                                 01561660  P  107
        SI:=SI-8; DI:=DI-8; TALLY:=2;                                         01561670  P  107
        IF 8 SC = DC THEN TALLY:=1;                                           01561680  P  107
        END;                                                                  01561690  P  107
    CMPD:=TALLY;                                                              01561700  P  107
    END CMPD;                                                                 01561710  P  107
    IF INSERTDEPTH > 0 AND INSERTCOP=1 OR INSERTDEPTH=0 THEN                  02193800  P  107
    LABEL COPYLIB, COPYEOF;                                                   02210600  P  107
                          LIBCLAST, LIBTLAST, COPYLIB;                        02211000  P  107
    GO COMPAR;                                                                02224010  P  107
COPYLIB;                                                                      02224020  P  107
    READ(LF[INSERTINX:=INSERTINX+1],10,LBUFF[*])[COPYEOF];                    02224030  P  107
    READ SEEK(LF[INSERTINX+1]);                                               02224032  P  107
    IF(CMPD(INSERTSEQ,LBUFF[9]) = 0) THEN GO COPYEOF;                         02224040  P  107
    LCR:=MKABS(LBUFF[9]);                                                     02224050  P  107
    GO TO EXIT;                                                               02224060  P  107
COPYEOF;                                                                      02224070  P  107
    CLOSE(LF,RELEASE);                                                        02224080  P  107
    IF((INSERTDEPTH:=INSERTDEPTH-1)  =  0) THEN                               02224090  P  107
        BEGIN LASTUSED:=SAVECARD; MEDIUM:=MEDIUM.[24:12];                     02224100  P  107
        GO USETHESWITCH;                                                      02224102  P  107
        END;                                                                  02224104  P  107
    FILL LF WITH INSERTMID, INSERTFID;                                        02224110  P  107
    GO COPYLIB;                                                               02224120  P  107
PROCEDURE INCLUDECARD;                                                        02238100  P  107
    BEGIN                                                                     02238110  P  107
    REAL V;                                                                   02238112  P  107
    LABEL EEXIT,AGAIN,GETEM,EOF,EXIT,DONTSCAN;                                02238120  P  107
    REAL STREAM PROCEDURE SCNN(A,B); VALUE B;                                 02238130  P  107
        BEGIN                                                                 02238140  P  107
        SI:=A; DI:=LOC SCNN; DS:=8 LIT"0        ";                            02238150  P  107
```

```
        DI:=DI"7; SI:=SI+3; DS:=B CHR;                                  02238160  P  107
        END;                                                            02238170  P  107
    STREAM PROCEDURE MVE(A,B,C,D); VALUE B,C;                           02238180  P  107
        BEGIN                                                           02238190  P  107
        SI:=A; SI:=SI+3; DI:=D; C(DS:=LIT"0"); DS:=B CHR;               02238200  P  107
        END;                                                            02238210  P  107
STREAM PROCEDURE MVEWD(A,B); VALUE A;                                   02238212  P  107
    BEGIN SI:=A; DI:=B; DS:=10 WDS; END;                                02238214  P  107
    DEFINE SKAN = BEGIN                                                 02238220  P  107
                  COUNT:=RESULT:=ACCUM[0]:=0;                           02238230  P  107
                  SCANNER;                                              02238240  P  107
                  V:=SCNN(ACCUM[1],MIN(COUNT,7));                       02238250  P  107
                  END#;                                                 02238260  P  107
    DEFINE ERR(ERR1) = BEGIN FLAG(ERR1); GO TO FEXIT; END#;             02238270  P  107
    IF((INSERTDEPTH:=INSERTDEPTH+1) > INSERTMAX) THEN ERR(612);         02238280  P  107
    INSERTMID:=INSERTFID:=INSERTINX:=INSERTCOP:=0;                      02238290  P  107
    INSERTSEQ:="9999"&"9999"[1:23];                                     02238300  P  107
AGAIN;                                                                  02238330  P  107
    SKAN;                                                               02238340  P  107
DONTSCAN;                                                               02238342  P  107
    IF V="%       " THEN GO GETEM;                                      02238350  P  107
    IF V="/       " THEN GO AGAIN;                                      02238360  P  107
    IF RESULT=3 THEN % SEQ RANGE                                        02238370  P  107
        BEGIN                                                           02238380  P  107
        MVE(ACCUM[1],COUNT:=MIN(COUNT,8),8-COUNT,INSERTINX);            02238385  P  107
        SKAN;                                                           02238390  P  107
        IF V="-       " THEN                                            02238400  P  107
            BEGIN                                                       02238410  P  107
            SKAN;                                                       02238420  P  107
            IF RESULT # 3 THEN ERR(614);                                02238430  P  107
            MVE(ACCUM[1],COUNT:=MIN(COUNT,8),8-COUNT,INSERTSEQ);        02238440  P  107
        END ELSE GO TO DONTSCAN;                                        02238450  P  107
        GO AGAIN;                                                       02238460  P  107
        END; % SEQ RANGE                                                02238470  P  107
    IF V="+       " THEN % WE HAVE COPY FORM                            02238480  P  107
        BEGIN                                                           02238490  P  107
        SKAN;                                                           02238500  P  107
        IF V="COPY    " THEN                                            02238510  P  107
            IF EXAMIN(LCR-9)="$" THEN                                   02238512  P  107
                INSERTCOP:=INSERTINFO[INSERTDEPTH-1,4]                  02238514  P  107
            ELSE ERR(617)                                               02238520  P  107
        ELSE ERR(616);                                                  02238522  P  107
        GO AGAIN;                                                       02238530  P  107
        END;                                                            02238540  P  107
    IF INSERTMID=0 THEN INSERTMID:=V                                    02238550  P  107
    ELSE IF INSERTFID=0 THEN INSERTFID:=V ELSE ERR(616);                02238552  P  107
    GO AGAIN;                                                           02238555  P  107
GETEM;                                                                  02238560  P  107
    IF NOT BOOLEAN(INSERTCOP) AND NEWTOG THEN                           02238570  P  107
    IF EXAMIN(FCR) = "$" THEN % ONLY IF "$" IS IN COLUMN ONE            02238572  P  107
    IF BOOLEAN(INSERTINFO[INSERTDEPTH-1,4]) THEN % ONLY IF LAST HAD COPY02238574  P  107
        BEGIN MVEWD(FCR,LBUFF[0]);                                      02238580  P  107
        PUTSEQNO(LBUFF[9],MKABS(INFO[LASTSEQROW,LASTSEQUENCE]));        02238582  P  107
        WRITE(NEWTAPE,10,LBUFF[*]);                                     02238590  P  107
        END;                                                            02238600  P  107
    IF INSERTMID=0 THEN ERR(613);                                       02238602  P  107
    IF INSERTFID=0 THEN INSERTFID:=TIME("1);                            02238610  P  107
    IF INSERTFID=0 THEN                                                 02238620  P  107
```

```
        BEGIN INSERTFID:=INSERTMID; INSERTMID:=0; END;                                  02238630  P  107
    IF INSERTDEPTH > 1 THEN CLOSE(LF,RELEASE);                                         02238640  P  107
    FILL LF WITH INSERTMID,INSERTFID;                                                  02238650  P  107
    READ(LF[0],10,LBUFF[*])[EEXIT];     % DO THE FOLLOWING SO THAT                    02238652  P  107
    INSERTMID:=LF.MFID;                 % IF THE OPERATOR IL-ED US                     02238654  P  107
    INSERTFID:=LF.FID;                  % WE WILL HAVE THE PROPER NAMES.               02238656  P  107
    V:=-1;                                                                             02238658  P  107
    IF INSERTINX > 0 THEN                                                              02238660  P  107
        BEGIN                                                                          02238670  P  107
        DO READ(LF[V:=V+1],10,LBUFF[*])[EEXIT]                                         02238680  P  107
        UNTIL CMPD(INSERTINX,LBUFF[9]) ≤ 1;                                            02238690  P  107
        V:=V-1;                                                                        02238700  P  107
        END;                                                                           02238702  P  107
    INSERTINX:=V;                                                                      02238704  P  107
    IF INSERTDEPTH = 1 THEN                                                            02238710  P  107
    BEGIN SAVECARD:=LASTUSED; LASTUSED:=7; MEDIUM:="L "& MEDIUM[24:12];                02238720  P  107
    END;                                                                               02238730  P  107
    GO TO EXIT;                                                                        02238760  P  107
EEXIT;                                                                                 02238770  P  107
    IF((INSERTDEPTH:=INSERTDEPTH-1)  >  0) THEN                                        02238780  P  107
        BEGIN                                                                          02238790  P  107
        CLOSE(LF,RELEASE);                                                             02238800  P  107
        FILL LF WITH INSERTMID,INSERTFID;                                              02238810  P  107
        END;                                                                           02238820  P  107
EXIT;                                                                                  02238830  P  107
    Q:="1%0000";                                                                       02238832  P  107
    END;                                                                               02238840  P  107
    IF Q = "7INCLU" THEN                                                               02570000  P  107
        BEGIN INCLUDECARD; GO EXIT; END;                                               02571000  P  107
                "7INCLU",0,         % 54,55                                            09027056  P  107
            INSERTCOP:=1;                                                              09274100  P  107
$! BY KFK                                                                              99999900  P  107
$! DATE 3/14/75                                                                       99999901  P  107
$! MSA CENTRAL - DETROIT                                                              99999902  P  107
$!                                                                                     99999903  P  107
$!    THIS PATCH WILL ALLOW THE ALGOL COMPILER TO "INCLUDE" SOURCE                     99999904  P  107
$! CODE FROM DISK FILES AS IN FORTRAN.  THIS IS DONE WITH THE USE OF                   99999905  P  107
$! THE NEW DOLLAR SIGN CARD OPTION INCLUDE.                                           99999906  P  107
$!     THE SYNTAX FOR THE $ INCLUDE CARD IS:                                          99999907  P  107
$!                                                                                     99999908  P  107
$! $ INCLUDE <COPY PART> <FILE PART> <SEQUENCE PART>                                  99999909  P  107
$!                                                                                     99999910  P  107
$! <COPY PART> ::= <EMPTY> / + COPY                                                   99999911  P  107
$!                                                                                     99999912  P  107
$! <FILE PART> ::= <MULTI-FILE ID>/<FILE ID>    /                                    99999913  P  107
$!                 <MULTI-FILE ID>                                                    99999914  P  107
$!                                                                                     99999915  P  107
$! <MULTI-FILE ID> ::= [ALPHANUMERIC STRING OF 7 OR FEWER CHARACTERS]                 99999916  P  107
$!                                                                                     99999917  P  107
$! <FILE ID> ::= <EMPTY> / <ALPHANUMERIC STRING>                                      99999918  P  107
$!                                                                                     99999919  P  107
$! <SEQUENCE PART> ::= <STARTING SEQUENCE NUMBER> <ENDING SEQUENCE                    99999920  P  107
$!                       NUMBER> / <EMPTY>                                            99999921  P  107
$!                                                                                     99999922  P  107
$! <STARTING SEQUENCE NUMBER> ::= <UNSIGNED INTEGER>                                  99999923  P  107
$!                                                                                     99999924  P  107
$! <ENDING SEQUENCE NUMBER> ::= <EMPTY> / - <UNSIGNED INTEGER>                        99999925  P  107
```

```
$!                                                                                99999926  P  107
$! SOME EXAMPLES ARE:                                                            99999927  P  107
$!                                                                              99999928  P  107
$!   $ INCLUDE A/B 1213-99932                                                   99999929  P  107
$!   $ INCLUDE A 12321-77651                                                    99999930  P  107
$!   $ INCLUDE+COPY SPECIAL/FILE 76333-124457                                   99999931  P  107
$!   $ INCLUDE A 12223                                                          99999932  P  107
$!   $ INCLUDE A                                                                99999933  P  107
$!   $ INCLUDE + COPY IT                                                        99999934  P  107
$!                                                                              99999935  P  107
$! INCLUDE INSTRUCTS THE COMPILER TO COMPILE THE SOURCE CODE ON THE             99999936  P  107
$! DISK FILE <FILE PART> OVER THE RANGE <SEQUENCE PART> AS PART OF THE          99999937  P  107
$! ENTIRE PROGRAM.  IN THIS MANNER, THE USER CAN COMPILE ALL OR PART OF         99999938  P  107
$! AN AUXILLARY FILE(S) INTO HIS PROGRAM.  IF THE <FILE ID> IS NOT              99999939  P  107
$! PRESENT, THE USERCODE IS USED AS THE <FILE ID>.                              99999940  P  107
$!                                                                              99999941  P  107
$! THE STARTING AND ENDING SEQUENCE UNMBERS ARE INCLUSIVE.  IF THE              99999942  P  107
$! <SEQUENCE PART> IS EMPTY, THE ENTIRE FILE IS USED.  IF ONLY THE              99999943  P  107
$! STARTING SEQUENCE NUMBER IS PRESENT, THE FILE FROM THAT SEQUENCE             99999944  P  107
$! NUMBER TO THE END OF THE FILE IS USED.  IF BOTH SEQUENCE NUMBERS ARE         99999945  P  107
$! PRESENT, THE FILE FROM THE STARTING SEQUENCE TO ENDING SEQUENCE,             99999946  P  107
$! INCLUSIVE, IS USED.  IF A NEW FILE IS BEING MADE, AND THE                    99999947  P  107
$! COPY PART IS EMPTY, ANY IMBEDDED $ INCLUDE CARDS WILL BE WRITTEN ON          99999948  P  107
$! THE NEW FILE, BUT NOT THE INCLUDED FILES THEMSELVES.  THIS PROVIDES          99999949  P  107
$! THAT THE NEW FILE, WHEN IT ITSELF IS COMPILED, WILL INCLUDE THE              99999950  P  107
$! FILES.  WHILE AT THE SAME TIME ALLOWING THE INCLUDED FILES TO BE             99999951  P  107
$! UPDATED INDEPENDENTLY OF THE NEW FILE.                                       99999952  P  107
$!                                                                              99999953  P  107
$! IF A NEW FILE IS BEING MADE AND THE COPY PART IS PRESENT, THE                99999954  P  107
$! IMBEDDED $ INCLUDE CARDS WILL NOT BE WRITTEN OUT ON THE NEW FILE,            99999955  P  107
$! BUT RATHER THE INCLUDED RECORDS THEMSELVES WILL BE COPIED ONTO THE           99999956  P  107
$! NEW FILE.  THE COPY PART IS IGNORED IF A NEW FILE IS NOT BEING MADE.         99999957  P  107
$! NOTE THAT INCLUDED FILES CAN HAVE $ INCLUDE CARDS IMBEDDED WITHIN           99999958  P  107
$! THEM, AND THUS RECURSION ON THE $ INCLUDE CARDS CAN OCCUR.                   99999959  P  107
$!*****************************************************************************99999960  P  107



$#PATCH NUMBER 108 FOR ALGOL CONTAINS 17 CARDS                                           P  108

          IF BUILDLINE.[45:1] THEN                                            02331050  P  108
             BUILDLINE.[47:1]:=SEQXEQTOG:=FALSE;                              02331060  P  108
INTEGER SAVEINX;                                                              02365200  P  108
              IF BUILDLINE.[45:1] THEN                                        02525000  P  108
                BEGIN                                                         02525001  P  108
                  IF XMODE = 0 THEN                                           02525003  P  108
                    BEGIN                                                     02525004  P  108
                      OPTIONWORD:=BOOLEAN(0);                                 02525005  P  108
                      FOR SAVEINX:=1 STEP 2 UNTIL OPARSIZE DO                 02525006  P  108
                         OPTIONS[SAVEINX]:=0;                                 02525007  P  108
                      BUILDLINE.[47:1]:=SEQXEQTOG:=FALSE;                     02525008  P  108
                      IF LASTUSED < 5 THEN LASTUSED:=1;                       02525009  P  108
                      XMODE:=1;                                               02525010  P  108
                    END;                                                      02525011  P  108
                  SEQXEQTOG:=XMODE ≠ 2 AND XMODE ≠ 4;                         02525012  P  108
                  BUILDLINE.[47:1]:=SEQXEQTOG;                                02525013  P  108
                END;                                                          02526000  P  108
```

```
$!THIS PATCH MAKES IT POSSIBLE TO SET OR RESET COMPILER OPTION                    P  108
$!"SEQXEQ" ANY NUMBER OF TIMES BEFORE THE FIRST BEGIN.                           P  108
$! AFTER THAT ITS CONDITION WILL REMAIN UNCHANGE. IT STILL                       P  108
$! MAY NOT BE POPPED (POP WILL ACT AS A RESET).                                  P  108
$!*************************************************************************   P  108



$#PATCH NUMBER 109 FOR ALGOL CONTAINS 19 CARDS.                          00000000  P  109

                        (GIT(HOLE)),FUNCMONFILE);                        07422000  P  109
                        EMITNUM(1&CARDNUMBER[1:4:44]);                   07422100  P  109
                        SWITMONFILE);                                    07516000  P  109
                        EMITNUM(0&CARDNUMBER[1:4:44]);                   07516100  P  109
                        EMITV(GNAT(                                      07516200  P  109
                                PASSMONFILE(RR1);                        07619000  P  109
                                EMITNUM(0&CARDNUMBER[1:4:44]));          07619100  P  109
          FILEIDENT);                                                    10891000  P  109
          EMITNUM(FORMATTYPE&CARDNUMBER[1:4:44]);                        10891100  P  109
          EMITV(GNAT(PRINTI));                                           10891200  P  109
EMITNUM(1&CARDNUMBER[1:4:44]);                                           15011000  P  109
                          EMITNUM(5&CARDNUMBER[1:4:44]);                 15271000  P  109
                          EMITN(GNAT(PRINTI));                           15271100  P  109
                          EMITNUM((IF SPCLMON THEN 3 ELSE 2)             15326000  P  109
                                &CARDNUMBER[1:4:44]);                    15327000  P  109
                          EMITV(GNAT(PRINTI));                           15328000  P  109
                                EMITNUM(5&CARDNUMBER[1:4:44]);           15359000  P  109
                       EMITNUM(5&CARDNUMBER[1:4:44]);                    15374000  P  109
                            EMITV(GNAT(PRINTI));                         15374100  P  109
$! BY KFK - MSC DETROIT                                                   99990000  P  109
$! DATE 06/06/75                                                         99990100  P  109
$! THIS IS THE ALGOL HALF OF THE PATCH THAT WILL CAUSE THE               99990200  P  109
$! SEQUENCE NUMBER OF THE CARD IMAGE TO BE PRINTED ALONG WITH            99990300  P  109
$! THE MONITOR INFORMATION FOR AN ITEM BEING MONITORED.                  99990400  P  109
$! IN ADDITION, INTRINSICS PATCH 116 MUST BE USED WITH THIS              99990500  P  109
$! PATCH.                                                               99990600  P  109
$!*************************************************************************99990700  P  109



$#PATCH NUMBER 110 FOR ALGOL CONTAINS 2 CARDS.                           00000001  P  110

        XMODE:=1; IF LASTUSED < 5 AND LASTUSED # 3 THEN LASTUSED:=1;     02332000  P  110
    IF LASTUSED # 1 THEN GO TO AGAIN;                                    02437500  P  110
$! DATE 9/2/75                                                           99990000  P  110
$! BY JTC - MSA CENTRAL                                                  99990100  P  110
$! THIS PATCH CORRECTS A PROBLEM THAT WOULD OCCUR IF A                   99990200  P  110
$! "$ TAPE" CARD WERE INCLUDED IN THE TAPE OR CAST FILES                 99990300  P  110
$! INPUT TO AN ALGOL COMPILE. THE COMPILER WOULD HANG OR                 99990400  P  110
$! ABORT. NOW THE "$ TAPE" OPTION MAY ONLY BE SET FROM                   99990500  P  110
$! THE "CARD" INPUT FILE.                                               99990600  P  110
$!*************************************************************************99999999  P  110
```

```
$#PATCH NUMBER 111 FOR ALGOL CONTAINS 24 CARD.                                  00000001  P  111

 254          SCANNER: STRING, OCTAL, OR HEX CONSTANT HAS FLAG BIT SET.        00185000  P  111
        COUNT := 0; T := IF STREAMTOG THEN 63 ELSE 8;                           02691000  P  111
        IF NOT STREAMTOG AND COUNT=8 AND BOOLEAN(ACCUM[1],[18:1]) THEN          02697500  P  111
          BEGIN Q := ACCUM[1]; FLAG(254); GO TO SCANAGAIN; END;                 02697600  P  111
        IF COUNT < 8 OR (COUNT = 8 AND NOT BOOLEAN                              02703000  P  111
          (ACCUM[1],[18:1])) THEN % FLAG BIT NOT SET, FULL WORD CONST.          02703050  P  111
$ VOIDT 02781000                                                                02776000  P  111
            T.INCR := COUNT := (C×COUNT-1)DIV 6 + 1; % # OF CHARS.              02776100  P  111
            T.CLASS:= STRNGCON;                                                 02776200  P  111
            MOVECHARACTERS(1,ACCUM[4],0,ACCUM[1],3);                            02776300  P  111
          IF BOOLEAN(ACCUM[1],[18:1]) THEN % FLAG BIT SET.                      02776400  P  111
            IF STREAMTOG THEN                                                   02776500  P  111
                T.CLASS := STRING                                               02776600  P  111
              ELSE                                                              02776700  P  111
                FLAG(254)                                                       02776800  P  111
            ELSE                                                                02776900  P  111
              C := ACCUM[4]; % GET FULL WORD EQUIVALENT OF STRING.               02777000  P  111
            MOVECHARACTERS(COUNT,ACCUM[4],8-COUNT,ACCUM[1],3);                  02777050  P  111
            GO TO COMPLETE;                                                     02777100  P  111
            IF STEPI ≠ STRING AND ELCLASS ≠ STRNGCON AND                        16384000  P  111
              ELCLASS ≠ LITNO AND ELCLASS ≠ NONLITNO THEN                       16384100  P  111
              IF ELCLASS = LITNO OR ELCLASS = NONLITNO THEN                     16384700  P  111
                MOVECHARACTERS(COUNT:=IF ADDR < 8 THEN ADDR ELSE 8,             16384800  P  111
                  C,8-COUNT,ACCUM[1],3);                                        16384900  P  111
$; DATE 9/11/75                                                                 99990000  P  111
$; BY JTC - MSA CENTRAL.                                                        99990100  P  111
$; THIS PATCH ALTERS THE ALGOL COMPILER SO THAT STRING, HEX OR                  99990200  P  111
$; OCTAL CONSTANTS CAN CONTAIN 8, 12, OR 16 CHARACTERS RESPECTIVELY            99990300  P  111
$; AS LONG AS THE CONSTANTS DO NOT HAVE THE FLAG BIT SET.                      99990400  P  111
$; IF THE CONSTANT HAS THE FLAG BIT SET THEN A NEW ERROR MESSAGE               99990500  P  111
$; IS ELICITED, ERROR 254.                                                     99990600  P  111
$;     IN ADDITION, THE SYNTAX FOR THE "DS := N LIT <LITERAL>"                 99990700  P  111
$; STATEMENT IN A STREAM PROCEDURE HAS BEEN ALTERED SO THAT THE                99990800  P  111
$; HEX, OCTAL, REAL OR INTEGER CONSTANTS MAY BE USED IN PLACE OF               99990900  P  111
$; A STRING. IN THE CASE OF THE HEX OR OCTAL CONSTANTS, AN EQUIVA-             99991000  P  111
$; LENT SIX BIT STRING OF THE SAME LENGTH AS THE CONSTANT IS                   99991100  P  111
$; CONSTRUCTED AND USED. IN THE CASE OF THE REAL OR INTEGER                    99991200  P  111
$; CONSTANTS, THE FULL WORD EQUIVALENT IS CONSTRUCTED AND USED AS              99991300  P  111
$; AN 8 CHARACTERS STRING WITH THE EXCEPTION THAT IF N IS LESS                 99991400  P  111
$; THAN 8 THEN THE RIGHT MOST N CHARACTERS OF THE FULL WORD                    99991500  P  111
$; CONSTANT ARE USED AS THE LITERAL.                                           99991600  P  111
$;      SOME EXAMPLES:                                                        99991700  P  111
$;                                                                            99991800  P  111
$;          DS := 2 LIT 3"3714";        PUT LEFT ARROW (OCTAL 37) AND         99991900  P  111
$;                                        QUESTION MARK (OCTAL 14) IN         99992000  P  111
$;                                        DESTINATION STRING.                 99992100  P  111
$;                                                                            99992200  P  111
$;          DS := 8 LIT 3"2122";        PUT THE STRING "ABABABAB" INTO        99992300  P  111
$;                                        DESTINATION STRING.                 99992400  P  111
$;                                                                            99992500  P  111
$;          DS := 8 LIT 7;              PUT THE STRING "00000007" INTO        99992600  P  111
$;                                        DESTINATION STRING.                 99992700  P  111
$;                                                                            99992800  P  111
$;          DS := 4 LIT 76;             PUT THE STRING "00₁₂" INTO            99992900  P  111
$;                                        DESTINATION STRING.                 99993000  P  111
$;                                                                            99993100  P  111
```

```
$!          DS := 32 LIT 2.345;          PLACE THE FULL WORD CONSTANT      99993200  P  111
$!                                         2.345 INTO DESTINATION          99993300  P  111
$!                                         STRING 4 TIMES.                 99993400  P  111
$!*************************************************************************99999999  P  111



$#PATCH NUMBER 112 FOR ALGOL CONTAINS 1 CARD.                                00000001  P  112

        IF LASTUSED = 1 THEN MEDIUM := "C ";                                 02013566  P  112
$! DATE 9/23/75                                                             99990000  P  112
$! BY JTC - MSA CENTRAL.                                                    99990100  P  112
$! THIS PATCH CORRECTS A MINOR PROBLEM IN ALGOL WHICH                        99990200  P  112
$! CAUSED THE CARD IMAGES TO BE MARKED AS "0C" INSTEAD                       99990300  P  112
$! OF "C" AFTER A $CAST CALL.                                               99990400  P  112
$!*************************************************************************99999999  P  112



$#PATCH NUMBER 113 FOR ALGOL CONTAINS 6 CARDS.                               00000000  P  113

BOOLEAN PROCEDURE SWITCHGEN(BEFORE,PD);                                      10954000  P  113
  VALUE BEFORE; BOOLEAN BEFORE; REAL PD;                                     10954100  P  113
                PUT(TAKE(LASTINFO)&(PD:=PROGDESCBLDR(ADES,TL,PD))            10978500  P  113
                    [16:37:11],LASTINFO); % GET PRT LOC AND SAVE             10978600  P  113
                IF FORMALF := SWITCHGEN(TB1,GT1)                             14223000  P  113
$ VOID                                                                       14226000  P  113
$! DATE 9/24/75                                                             99990000  P  113
$! BY JTC - MSA CENTRAL                                                     99990100  P  113
$! THIS PATCH MAKES RECURSIVE SWITCH LABELS WORK RIGHT.                      99990200  P  113
$! PREVIOUSLY, EXTRA PRT LOCATIONS WERE BEING GENERATED                      99990300  P  113
$! AND NOT FIXED UP CAUSING SYSTEM HANGS AND VARIOUS                         99990400  P  113
$! OTHER NASTIES.                                                           99990500  P  113
$!*************************************************************************99999999  P  113



$# PATCH NUMBER 114 FOR ALGOL CONTAINS 13 CARDS.                             00000001  P  114

                  EDITLINE(LIN,FCR,L,[36:10],                                02182750  P  114
                           SGNO,L,[45:2],MEDIUM,OMITTING);                   02182760  P  114
STREAM PROCEDURE EDITLINE(LINE,NCR,R,S,L,SYMBOL,OMIT);                       02183500  P  114
   VALUE NCR,R,S,L,SYMBOL,OMIT;                                              02183750  P  114
    IF SC=" " THEN DS:=12 LIT" " ELSE                                        02186000  P  114
        BEGIN                                                                02186250  P  114
        S1:=LOC S; DS:=4 DEC; DS:=LIT "!";                                   02186300  P  114
        S1:=LOC R; DS:=4 DEC; DS:=LIT "!";                                   02186400  P  114
        S1:=LOC L; DS:=1 DEC; DS:=LIT " ";                                   02186500  P  114
        END;                                                                 02186600  P  114
   OMIT(DI := DI - 12; DS := 12 LIT "  !OMIT:    "; DI := LINE;              02186750  P  114
        DS := 8 LIT "  !OMIT!");                                             02186760  P  114
                  EDITLINE(LIN,FCR,"    ",0,0,MEDIUM,0);                     05038000  P  114
$! BY KFK - MSC DETROIT                                                     99990000  P  114
$! DATE 01/22/76                                                           99990100  P  114
```

```
$!    THIS PATCH ADDS THE SEGMENT AND SYLLABLE NUMBER TO                  99990200  P  114
$! THE OUTPUT PRODUCED ON THE LINE PRINTER FILE.                          99990300  P  114
$!*********************************************************************  99990400  P  114



$#PATCH NUMBER 116 FOR ALGOL CONTAINS 515 CARDS                           00000000  P  116

$ VOIDT 01001771                                                         01001750  P  116
%*************************************************************************01007005  P  116
%              X R E F   S T U F F                                       01007010  P  116
%*************************************************************************01007015  P  116
%                                                                        01007020  P  116
ARRAY                                                                    01007025  P  116
  XREFAY2[0:29],        % ARRAY OF ONE WORD REFERENCE RECORDS.           01007030  P  116
                        % THE LAYOUT OF EACH WORD IS                    01007035  P  116
                        %                                               01007040  P  116
                        %         .[1:5]  TYPE OF REFERENCE             01007045  P  116
                        %                  = 0 FOR FORWARD DECL         01007050  P  116
                        %                  = 1 FOR LABEL OCCURENCE      01007051  P  116
                        %                  = 2 FOR NORMAL DECL          01007055  P  116
                        %                  = 4 FOR NORMAL REFERENCE     01007060  P  116
                        %                  = 5 FOR ASSIGNMENT           01007065  P  116
                        %                                               01007070  P  116
                        %                   NOTE: THE LOWER ORDER BIT   01007075  P  116
                        %                         OF THIS FIELD IS ON   01007080  P  116
                        %                         IF YOU WANT STARS     01007085  P  116
                        %                         AROUND THIS REFERENCE 01007090  P  116
                        %                         IN THE XREF           01007095  P  116
                        %                                               01007100  P  116
                        %         .[6:15] IDENTIFIER ID. NO.            01007105  P  116
                        %                  THIS IS A UNIQUE NUMBER THAT 01007110  P  116
                        %                  IS ASSIGNED WHEN THE         01007115  P  116
                        %                  IDENTIFIER IS ENCOUNTERE     01007120  P  116
                        %                  FOR THE FIRST TIME.          01007125  P  116
                        %                                               01007130  P  116
                        %         .[21:27] SEQUENCE NUMBER              01007135  P  116
                        %                                               01007140  P  116
  XREFAY1[0:9],         % RECORD BUFFER AREA FOR WRITING OUT THE        01007145  P  116
                        % NAME INFORMATION RECORDS. ONE RECORD          01007150  P  116
                        % IS WRITTEN FOR EACH IDENTIFIER IN THE SYMBOL  01007155  P  116
                        % TABLE WHEN THE IDENTIFIER IS PURGED FROM THE  01007160  P  116
                        % SYMBOL TABLE, I.E., WHEN LEAVING THE BLOCK    01007165  P  116
                        % IN WHICH THE IDENTIFIER IS DECLARED.          01007170  P  116
                        %                                               01007175  P  116
                        % THE LAYOUT OF EACH RECORD IS:                 01007180  P  116
                        %                                               01007185  P  116
                        %       WORDS 0-7   THE IDENTIFIER WITH BLANK   01007190  P  116
                        %                     FILL ON THE RIGHT         01007195  P  116
                        %                                               01007200  P  116
                        %       WORD 8                                  01007205  P  116
                        %         .[21:12]  SEGMENT NUMBER IN WHICH     01007210  P  116
                        %                     THIS IDENTIFIER WAS DECLARED01007215  P  116
                        %                                               01007220  P  116
                        %         .[33:15]  IDENTIFIER ID. NO.          01007225  P  116
                        %                                               01007230  P  116
                        %       WORD 9      FLOAT WORD                  01007235  P  116
```

```
  XINFO[0:31,0:127];  % THIS ARRAY CONTAINS ONE ENTRY FOR EACH ENTRY      01007240  P  116
                      % IN THE INFO TABLE. IF YOU HAVE THE INDEX          01007245  P  116
                      % OF THE ELBAT WORD FOR AN IDENTIFIER IN            01007250  P  116
                      % THE INFO TABLE YOU CAN FIND THE XINFO WORD        01007255  P  116
                      % FOR THE IDENTIFIER  BY REFERRING TO:              01007260  P  116
                      %                                                   01007265  P  116
                      %      XINFO[INDEX.LINKR,INDEX.LINKC DIV 2]         01007270  P  116
                      %                                                   01007275  P  116
                      % EACH ENTRY CONTAINS:                             01007280  P  116
                      %                                                   01007285  P  116
                      %            .[21:12]  SEGMENT NUMBER IN WHICH      01007290  P  116
                      %                        THIS IDENTIFIER WAS DECL   01007295  P  116
                      %                                                   01007300  P  116
                      %            .[33:15]  IDENTIFIER ID. NO.           01007305  P  116
                      %                        IF THIS ID. NO. IS ZERO    01007310  P  116
                      %                        THEN XREF WAS NOT ON       01007315  P  116
                      %                        AT THE TIME THE IDENT      01007320  P  116
                      %                        WAS DECLARED AND ALL       01007325  P  116
                      %                        FUTURE REFERENCES WILL     01007330  P  116
                      %                        BE DISCARDED.              01007335  P  116
                      %                                                   01007340  P  116
  INTEGER             %                                                   01007345  P  116
    XREFPT,           % CONTAINS INDEX OF NEXT AVAILABLE SLOT IN          01007350  P  116
                      % XREFAY2, WHEN THIS BECOMES GREATER                01007355  P  116
                      % THAN 30 THE CURRENT ARRAY IS DUMPED TO DISK       01007360  P  116
                      % AND XREFPT IS RESET TO ZERO.                      01007365  P  116
                      %                                                   01007370  P  116
    XLUN;             % THIS VARIABLE CONTROLS THE ASSIGNING OF           01007375  P  116
                      % ID. NO. TO IDENTIFIERS. IT IS INCREMENTED         01007380  P  116
                      % EACH TIME A NEW IDENTIFIER IS ENCOUNTERED.        01007385  P  116
                      %                                                   01007390  P  116
  DEFINE              %                                                   01007395  P  116
    SEGNOF = [21:12]#,    % FIELDS IN XINFO ENTRIES AND WORD 8 OF         01007400  P  116
    IDNOF  = [33:15]#,    % IDENTIFIER RECORDS.                           01007405  P  116
                          %                                               01007410  P  116
    TYPEREF = [1:5]#,     % FIELDS OF REFERENCE WORDS                     01007415  P  116
    REFIDNOF =[6:15]#,    %                                              01007420  P  116
    SEQNOF = [21:27]#,    %                                              01007425  P  116
                          %                                               01007430  P  116
    XREFIT(INDEX,SEQNO,REFTYPE) = % DEFINE TO ADD INFO TO REF TABLE       01007435  P  116
      BEGIN IF XREF THEN CROSSREFIT(INDEX,SEQNO,REFTYPE); END#,           01007440  P  116
                          %                                               01007445  P  116
    XMARK(REFTYPE) =      % DEFINE TO CHANGE LAST ENTRY IN REF TABLE TO A 01007450  P  116
      BEGIN IF XREF THEN XREFAY2[XREFPT-1].TYPEREF := REFTYPE END#,       01007455  P  116
                          %                                               01007460  P  116
    XREFDUMP(INDEX) =     % DEFINE TO DUMP SYMBOL TABLE INFO FOR IDENTIFIER01007465  P  116
      BEGIN IF DEFINING.[1:1] THEN CROSSREFDUMP(INDEX); END#,             01007470  P  116
                          %                                               01007475  P  116
    XREFINFO[INDEX] =     % DEFINE TO TRANSLATE INFO ROW AND COLUMN TO    01007480  P  116
      XINFO[(INDEX).LINKR,(INDEX).LINKC DIV 2]#, % XINFO ROW AND COL      01007481  P  116
                          %                                               01007482  P  116
    FORWARDREF = 0#,      % DEFINES FOR DIFFERENT REFERENCE TYPES         01007483  P  116
    LBLREF     = 1#,      %                                              01007485  P  116
    DECLREF    = 2#,      %                                              01007486  P  116
    NORMALREF  = 4#,      %                                              01007490  P  116
    ASSIGNREF  = 5#;      %                                              01007495  P  116
                          %                                               01007500  P  116
```

```
        FILE DSK1 DISK SERIAL [20:816](2,10,30);                                01561085  P  116
        FILE DSK2 DISK SERIAL [20:450](2,30,30);                                01561087  P  116
%***************************************************************************02001605  P  116
%                                                                               02001610  P  116
%       MISCELLANEOUS CROSS REFERENCE PROCEDURES                                02001615  P  116
%                                                                               02001620  P  116
$ VOIDT 02001831                                                                02001625  P  116
%***************************************************************************02001630  P  116
%                                                                               02001635  P  116
PROCEDURE CROSSREFIT(INDEX,SEQNO,REFTYPE);                                      02001640  P  116
  VALUE INDEX,SEQNO,REFTYPE;                                                    02001645  P  116
  REAL  INDEX,SEQNO,REFTYPE;                                                    02001650  P  116
BEGIN                                                                           02001655  P  116
 IF XREFINFO[INDEX].IDNOF ≠ 0 THEN % SAVE                                       02001660  P  116
 BEGIN                                                                          02001665  P  116
  IF XREFPT > 29 THEN % NO SLOTS LEFT IN ARRAY. WRITE IT OUT.                   02001670  P  116
    BEGIN                                                                       02001675  P  116
      WRITE(DSK2,30,XREFAY2[*]);                                                02001680  P  116
      XREFPT := 0;                                                              02001685  P  116
    END;                                                                        02001690  P  116
  XREFAY2[XREFPT] := SEQNO & REFTYPE TYPEREF & XREFINFO[INDEX]                   02001695  P  116
                                    REFIDNOF;                                   02001700  P  116
  XREFPT := XREFPT + 1; % EVEN THOUGH THE ARRAY MAY BE FULL NOW WE              02001705  P  116
                        % CANT WRITE IT OUT BECAUSE SOME ROUTINES               02001710  P  116
                        % WILL LOOK BACK AT THE ENTRY WE JUST PUT               02001715  P  116
                        % IN AND FIX IT UP.                                     02001720  P  116
 END;                                                                           02001725  P  116
END OF CROSSREFIT;                                                              02001730  P  116
%                                                                               02001735  P  116
PROCEDURE CROSSREFDUMP(INDEX);                                                  02001740  P  116
  VALUE INDEX;                                                                  02001745  P  116
  REAL INDEX;                                                                   02001750  P  116
BEGIN                                                                           02001755  P  116
  STREAM PROCEDURE MOVEXREFINFO(S,D,N);                                         02001760  P  116
    VALUE N;                                                                    02001765  P  116
  BEGIN                                                                         02001770  P  116
    SI := D; DI := D; DS := 8 LIT " "; DS := 7 WDS; % BLANK RECORD              02001775  P  116
    SI := S; SI := SI + 3; DI := D; DS := N CHR; % MOVE IDENTIFIER              02001780  P  116
  END OF MOVEXREFINFO;                                                          02001785  P  116
  %                                                                             02001790  P  116
 IF XREFINFO[INDEX].IDNOF ≠ 0 THEN % DUMP IT                                    02001795  P  116
 BEGIN                                                                          02001800  P  116
  MOVEXREFINFO(INFO[INDEX.LINKR,INDEX.LINKC+1],XREFAY1[*],                      02001805  P  116
                    TAKE(INDEX+1),[12:6]);                                      02001810  P  116
  XREFAY1[8] := XREFINFO[INDEX];                                                02001815  P  116
  XREFAY1[9] := TAKE(INDEX); % ELBAT WORD                                       02001820  P  116
  WRITE(DSK1,10,XREFAY1[*]);                                                    02001821  P  116
  XREFINFO[INDEX] := 0;                                                         02001822  P  116
 END;                                                                           02001825  P  116
END OF CROSSREFDUMP;                                                            02001830  P  116
$VOIDT 02882921                                                                 02882099  P  116
        IF GT1 ≠1 AND NOT MACROID THEN % NOT RESERVED WORD                      02882100  P  116
          XREFIT(T,LINK,CARDNUMBER,NORMALREF); % BUILD XREF ENTRY               02882200  P  116
$VOID                                                                           07596500  P  116
            XMARK(LBLREF); % THIS WILL SORT AHEAD OF DECLARATION                07598100  P  116
                           % WHEN WE GET AROUND TO THE XREF.                    07598200  P  116
    XMARK(ASSIGNREF); % FILL STATEMENT                                          07687600  P  116
```

```
                    XMARK(ASSIGNREF); % SEARCH STATEMENT                    07835500  P  116
                   XMARK(ASSIGNREF); % FOR STATEMENT                        08178100  P  116
        XREFDUMP(POINTER); % DUMP XREF INFO                                 13283500  P  116
$ VOIDT 13301731                                                            13301099  P  116
         IF XREF THEN % MAKE DECLARATION REFERENCE                          13310050  P  116
          IF (ACCUM[0].CLASS ≠ DEFINEDID OR NOT                             13310075  P  116
              BOOLEAN(ACCUM[0].FORMAL)) THEN % NOT DEFINE PARAMETER         13310080  P  116
$ VOIDT 13310601                                                            13310099  P  116
            BEGIN                                                           13310100  P  116
              XREFINFO[NEXTINFO] :=                                         13310200  P  116
                IF SPECTOG THEN                                             13310300  P  116
                  XREFINFO[ELBAT[I]]                                        13310350  P  116
                ELSE                                                        13310400  P  116
                  ((XLUN := XLUN + 1) & SGNO SEGNOF);                       13310450  P  116
              IF SPECTOG THEN % JUST GO BACK AND FIX UP XREF ENTRY           13310500  P  116
                XMARK(DECLREF)                                              13310525  P  116
              ELSE                                                          13310550  P  116
                XREFIT(NEXTINFO,CARDNUMBER,IF PTOG AND NOT STREAMTOG        13310575  P  116
                       THEN NORMALREF ELSE DECLREF);                        13310580  P  116
            END                                                             13310600  P  116
           ELSE % DEFINE PARAMETERS - DONT CROSS REF.                       13310700  P  116
            XREFINFO[NEXTINFO] := 0                                         13310750  P  116
          ELSE                                                              13310800  P  116
            IF DEFINING.[1:1]  THEN % WE ARE DOING XREFING                  13310900  P  116
              XREFINFO[NEXTINFO] := 0;                                      13310950  P  116
             IF XREF AND NOT SPECTOG THEN % ERASE PREVIOUS XREF ENTRY.      13348100  P  116
                   ACCUM[0] := 0 & DEFINEDID CLASS & 1 FORMAL;              14259090  P  116
$VOIDT                                                                      14259092  P  116
       XMARK(DECLREF); % PROCEDURE DECLARED FORWARD, MARK LAST               14310500  P  116
                       % XREF ENTRY AS A DECLARATION.                       14310501  P  116
         XREFDUMP(J); % DUMP XREF INFO                                      14445100  P  116
          XREFIT(PROINFO,0,FORWARDREF); % WE NEED THIS SO WE CAN FIND       14469100  P  116
                        % THE FORWARD DECL. DURING XREF                    14469101  P  116
       REAL REMEMBERSEQNO; % REMEMBERS SEQUENCE NUMBER OF VARIABLE          15075550  P  116
                        % ON LEFT HAND SIDE OF ASSIGNMENT SO WE            15075551  P  116
                        % CAN XREF IT CORRECTLY.                           15075552  P  116
            REMEMBERSEQNO := CARDNUMBER;                                    15085100  P  116
       XMARK(ASSIGNREF); % ASSIGNMENT TO SIMPLE VARIABLE.                   15091100  P  116
$VOIDT                                                                      15092010  P  116
                        BEGIN                                               15113100  P  116
                          BEGIN                                             15115000  P  116
                            ERR(201); % PARTIAL WORD NOT LEFT-MOST          15115100  P  116
                            GO TO EXIT;                                     15115200  P  116
                          END;                                              15115300  P  116
                          XREFIT(TALL,REMEMBERSEQNO,ASSIGNREF);             15116000  P  116
                          GO TO L1;                                         15116100  P  116
                        END;                                                15116200  P  116
                    XREFIT(TALL,REMEMBERSEQNO,ASSIGNREF); % ASSIGNMENT TO15301100  P  116
                                % SUBSCRIPTED VARIABLE.                     15301200  P  116
                    IF STEPI = ASSIGNOP THEN                                15342000  P  116
                      IF P1 = FS THEN % PARTIAL WORD IS LEFT-MOST           15342100  P  116
                        BEGIN                                               15342200  P  116
                          XREFIT(TALL,REMEMBERSEQNO,ASSIGNREF); % PARTIAL15342300  P  116
                                % WORD ASSIGNMENT TO SUBSCR. VAR.           15342400  P  116
                          GO TO LAST;                                       15342500  P  116
                        END                                                 15342600  P  116
         XMARK(LBLREF); % MARK LABEL OCCURENCE FOR XREF                      16159100  P  116
```

```
                XMARK(ASSIGNREF);                                          16318500  P  116
IF (XREF OR DEFINING.[1:1]) AND XLUN > 0 THEN                              17001000  P  116
    DEFINE XREFINFO[INDEX] = INFO[((INDEX).CF DIV 2).[33:7],               17002005  P  116
                                  ((INDEX).CF DIV 2).LINKC]#,              17002006  P  116
           CF = [33:15]#,                                                  17002007  P  116
           FF = [16:15]#,                                                  17002008  P  116
           NEWID[INDEX] = (IF BOOLEAN(INDEX) THEN XREFINFO[INDEX].FF       17002009  P  116
                           ELSE XREFINFO[INDEX].CF)#;                      17002010  P  116
    ARRAY TIMINGS[0:2,0:3];                                                17002012  P  116
    PROCEDURE SAVETIMES(I);                                                17002015  P  116
      VALUE I; INTEGER I;                                                  17002020  P  116
    BEGIN                                                                  17002025  P  116
      INTEGER J;                                                           17002030  P  116
      FOR J := 1 STEP 1 UNTIL 3 DO                                         17002035  P  116
        TIMINGS[I,J] := TIME(J);                                           17002040  P  116
    END;                                                                   17002045  P  116
    PROCEDURE UPDATETIMES(I);                                              17002050  P  116
      VALUE I; INTEGER I;                                                  17002055  P  116
    BEGIN                                                                  17002060  P  116
      INTEGER J;                                                           17002065  P  116
      FOR J := 1 STEP 1 UNTIL 3 DO                                         17002070  P  116
        TIMINGS[I,J] := TIME(J) - TIMINGS[I,J];                            17002075  P  116
    END;                                                                   17002080  P  116
    SAVETIMES(0); % SAVE TIMES FOR START OF IDENTIFIER SORT.               17002525  P  116
    TOTALNO := XLUN; % REMEMBER NUMBER OF IDENTIFIERS.                     17004500  P  116
      FOR I:= 0 STEP 1 UNTIL 8191 DO                                       17004700  P  116
          XREFINFO[I] := 0;                                                17004710  P  116
              BEGIN                                                        17022000  P  116
                REWIND(DSK1);                                              17022100  P  116
                UPDATETIMES(0); % UPDATE TIMES FOR IDENTIFIER SORT.        17022200  P  116
                TIMINGS[0,0] := XLUN; % NUMBER OF IDENTIFIERS SORTED.      17022300  P  116
              END                                                          17022400  P  116
              IF BOOLEAN(A[8]) THEN                                        17025000  P  116
                XREFINFO[A[8]].FF := XLUN := XLUN + 1                      17025100  P  116
              ELSE                                                         17025200  P  116
                XREFINFO[A[8]].CF := XLUN := XLUN + 1;                     17025300  P  116
              A[8].IDNOF := XLUN;                                          17025400  P  116
          IF 63 SC < DC THEN                                               17033000  P  116
            TALLY := 1                                                     17033100  P  116
          ELSE                                                             17033200  P  116
            BEGIN                                                          17033300  P  116
              SI := A;                                                     17033400  P  116
              DI := B;                                                     17033500  P  116
              IF 63 SC = DC THEN                                           17033600  P  116
                TALLY := 2;                                                17033700  P  116
            END;                                                           17033800  P  116
          DS := 8 LIT 3"777777777"; % ID.NO. AND SEG.NO. FIELDS            17041100  P  116
              IF REAL(COMP1:=COMPS1(A,B)) = 2 THEN % IDS EQUAL             17042300  P  116
                COMP1 := A[8].IDNOF < B[8].IDNOF;                          17042350  P  116
        SORT(OUTPUT1,INPUT1,0,HV1,COMP1,10,IF TOTALNO < 1000 THEN          17045000  P  116
                7000 ELSE 10000);                                          17045100  P  116
$ VOIDT 17069001                                                           17047001  P  116
    ARRAY IDTYPE[0:(IDMAX+3)*4-1];                                         17047100  P  116
    STREAM PROCEDURE SETUPHEADING(S,D,SEG,SEQNO,FWDTOG,LBLTOG,             17047200  P  116
                                  FWDSEQNO,TYPE,OWNTOG,PARAMTOG,           17047300  P  116
                                  VALTOG);                                 17047350  P  116
      VALUE SEG,SEQNO,FWDTOG,LBLTOG,FWDSEQNO,OWNTOG,PARAMTOG,              17047400  P  116
```

```
    VALTOG;                                                                  17047450  P  116
BEGIN                                                                        17047500  P  116
  SI := S;                                                                   17047700  P  116
  DI := D;                                                                   17047800  P  116
  63 (IF SC = " " THEN JUMP OUT ELSE DS := CHR);                             17047900  P  116
  DS := 6 LIT "  --  ";                                                      17048000  P  116
  OWNTOG (DS := 4 LIT "OWN ");                                               17048100  P  116
  SI := TYPE;                                                                17049300  P  116
  32 (IF SC = "," THEN JUMP OUT ELSE DS := CHR);                             17049400  P  116
  PARAMTOG (DS := 6 LIT "  --  ";                                           17049410  P  116
            DS := 4 LIT "NAME";                                              17049420  P  116
            VALTOG (DI := DI - 4; DS := 5 LIT "VALUE");                      17049430  P  116
            DS := 10 LIT " PARAMETER");                                      17049440  P  116
  DS := 26 LIT  "  --  DECLARED IN SEGMENT ";                               17049500  P  116
  SI := LOC SEG;                                                             17049600  P  116
  S := DI;                                                                   17049700  P  116
  DS := 4 DEC; DI := DI - 4; DS := 3 FILL; % CONV AND ZERO SUPPR             17049800  P  116
  DI := DI + 8; % TO FORCE STORE OF LAST WORD                                17049900  P  116
  SI := S;                                                                   17050000  P  116
  DI := S;                                                                   17050100  P  116
  4(IF SC # " " THEN DS:= CHR ELSE SI := SI + 1);                            17050200  P  116
  DS := 4 LIT " AT ";                                                        17050300  P  116
  SI := LOC SEQNO;                                                           17050400  P  116
  DS := 8 DEC;                                                               17050500  P  116
  FWDTOG (DS := 17 LIT "  --  FORWARD AT ";                                  17050600  P  116
          SI := LOC FWDSEQNO;                                                17050700  P  116
          DS := 8 DEC);                                                      17050800  P  116
  LBLTOG (DS := 16 LIT "  --  OCCURS AT ";                                   17050900  P  116
          SI := LOC FWDSEQNO;                                                17051000  P  116
          DS := 8 DEC);                                                      17051100  P  116
END OF SETUPHEADING;                                                         17051200  P  116
                                                                             17051300  P  116
STREAM PROCEDURE ADDASEQNO(SEQNO,N,STARS,D);                                 17051400  P  116
  VALUE SEQNO,N,STARS;                                                       17051500  P  116
BEGIN                                                                        17051600  P  116
  DI := D;                                                                   17051700  P  116
  DI := DI + 8;                                                              17051800  P  116
  N (DI := DI + 10);                                                         17051900  P  116
  STARS(DI := DI - 1; DS := LIT "*");                                        17052000  P  116
  SI := LOC SEQNO;                                                           17052100  P  116
  DS := 8 DEC;                                                               17052200  P  116
  DS := LIT " ";                                                             17052300  P  116
  STARS (DI := DI - 1; DS := LIT "*");                                       17052400  P  116
END;                                                                         17052500  P  116
STREAM PROCEDURE BLANKET(D);                                                 17052600  P  116
BEGIN                                                                        17052700  P  116
  DI := D;                                                                   17052800  P  116
  DS := 8 LIT " ";                                                           17052900  P  116
  SI := D;                                                                   17053000  P  116
  DS := 16 WDS;                                                              17053100  P  116
END OF BLANKET;                                                              17053200  P  116
PROCEDURE PRINTXREFSTATISTICS;                                               17053300  P  116
BEGIN                                                                        17053400  P  116
  SWITCH FORMAT STATS :=                                                     17053500  P  116
    (///, "CROSS REFERENCE STATISTICS", /,                                   17053600  P  116
          "------ --------- ----------", /),                                17053700  P  116
    ("PHASE ONE - SORT",I6," IDENTIFIERS"),                                  17053800  P  116
```

```
    ("PHASE TWO - SORT",I7," REFERENCES"),                                  17053900  P  116
    ("PHASE THREE - PRINT CROSS REFERENCE (",I7," LINES)"),                 17054000  P  116
    (X5,I4,":",211," ELAPSED TIME (MIN:SEC)"),                              17054100  P  116
    (X5,I4,":",211," PROCESSOR TIME"),                                      17054200  P  116
    (X5,I4,":",211," I/O TIME",/));                                         17054300  P  116
  INTEGER I,J,K;                                                            17054400  P  116
  WRITE(LINE,STATS[0]);                                                     17054500  P  116
  FOR I := 0 STEP 1 UNTIL 2 DO                                              17054600  P  116
    BEGIN                                                                   17054700  P  116
      WRITE(LINE,STATS[I+1],TIMINGS[I,0]);                                  17054800  P  116
      FOR J := 1 STEP 1 UNTIL 3 DO                                          17054900  P  116
        BEGIN                                                               17055000  P  116
          K := (TIMINGS[I,J] + 30) DIV 60; % ROUND TO NEAREST SECON         17055010  P  116
          WRITE(LINE,STATS[J+3],K DIV 60,(K:=K MOD 60) DIV 10,              17055020  P  116
                                         K MOD 10);                         17055025  P  116
        END;                                                                17055030  P  116
    END;                                                                    17055100  P  116
END PRINTXREFSTATISTICS;                                                    17055200  P  116
    DEFINE REFCOUNT = TIMINGS[1,0]#; % NUMBER OF REFERENCES SORTED.         17069300  P  116
    BOOLEAN FIRSTTIME; % TRUE ON FIRST CALL OF OUTPUT PROCEDURE,            17069400  P  116
    REAL LASTADDRESS;                                                       17069600  P  116
      DEFINE I = LASTADDRESS#;                                              17073100  P  116
    A[0] := I & NEWID[].REFIDNOF] REFIDNOF;                                 17080000  P  116
    REFCOUNT := REFCOUNT + 1;                                               17080100  P  116
    XREFAY1[8] := XREFPT := LASTADDRESS := 0;                               17084000  P  116
    FILL IDTYPE[*] WITH                                                     17084010  P  116
      "UNKNOWN,                         ", % 0                            17084020  P  116
      "STREAM LABEL,                    ", % 1                            17084030  P  116
      "STREAM VARIABLE,                 ", % 2                            17084040  P  116
      "DEFINE,                          ", % 3                            17084050  P  116
      "LIST,                            ", % 4                            17084060  P  116
      "FORMAT,                          ", % 5                            17084070  P  116
      "SWITCH FORMAT,                   ", % 6                            17084080  P  116
      "FILE,                            ", % 7                            17084090  P  116
      "SWITCH FILE,                     ", % 8                            17084100  P  116
      "SWITCH LABEL,                    ", % 9                            17084110  P  116
      "PROCEDURE,                       ", % 10                           17084120  P  116
      "INTRINSIC,                       ", % 11                           17084130  P  116
      "STREAM PROCEDURE,                ", % 12                           17084140  P  116
      "BOOLEAN STREAM PROCEDURE,        ", % 13                           17084150  P  116
      "REAL STREAM PROCEDURE,           ", % 14                           17084160  P  116
      "ALPHA STREAM PROCEDURE,          ", % 15                           17084170  P  116
      "INTEGER STREAM PROCEDURE,        ", % 16                           17084180  P  116
      "BOOLEAN PROCEDURE,               ", % 17                           17084182  P  116
      "REAL PROCEDURE,                  ", % 18                           17084184  P  116
      "ALPHA PROCEDURE,                 ", % 19                           17084186  P  116
      "INTEGER PROCEDURE,               ", % 20                           17084188  P  116
      "BOOLEAN,                         ", % 21                           17084190  P  116
      "REAL,                            ", % 22                           17084200  P  116
      "ALPHA,                           ", % 23                           17084210  P  116
      "INTEGER,                         ", % 24                           17084220  P  116
      "BOOLEAN ARRAY,                   ", % 25                           17084230  P  116
      "REAL ARRAY,                      ", % 26                           17084240  P  116
      "ALPHA ARRAY,                     ", % 27                           17084250  P  116
      "INTEGER ARRAY,                   ", % 28                           17084260  P  116
      "LABEL,                           ", % 29                           17084270  P  116
      "FAULT,                           ", % 30 (CLASS = 126)             17084280  P  116
      "SWITCH LIST,                     "; % 31 (CLASS = 127)             17084290  P  116
```

```
          LABEL EOF2, SKIP;                                                  17091100  P  116
          OWN BOOLEAN B2, FWDTOG, LBLTOG, WAITINGFORFWDREF;                  17091110  P  116
          DEFINE MATCH(A,B) = REAL(BOOLEAN(A) EQV BOOLEAN(B)) =              17091115  P  116
                              REAL(NOT FALSE)#;                              17091116  P  116
          REAL I;                                                            17091120  P  116
          DEFINE LINECOUNT = TIMINGS[2,0]#; % NUMBER OF LINES PRINTED.       17091140  P  116
          OWN REAL FWDSEQNO;                                                 17091150  P  116
          IF FIRSTTIME THEN % PRINT HEADINGS AND SAVE TIMINGS.               17091155  P  116
            BEGIN                                                            17091160  P  116
              FIRSTTIME := FALSE;                                            17091162  P  116
              TIME1 := TIME(1);                                              17091165  P  116
              DATIME;                                                        17091170  P  116
              UPDATETIMES(1);                                                17091175  P  116
              SAVETIMES(2); % SAVE TIMES FOR START OF XREF PRINT.            17091180  P  116
            END;                                                             17091200  P  116
$VOIDT 17111001                                                              17091201  P  116
          IF NOT B2 THEN                                                     17091210  P  116
            IF B THEN % END OF SORT - LIST OUT REST OF SEQ. NO.               17091300  P  116
              IF XREFPT # 0 THEN % WE GOT SOME TO LIST OUT                   17091400  P  116
                BEGIN                                                        17091500  P  116
                  WRITE(LINE[DBL],15,PAY[*]);                                17091510  P  116
                  LINECOUNT := LINECOUNT + 1;                                17091520  P  116
                END                                                          17091530  P  116
              ELSE % NOTHING TO LIST OUT                                     17091600  P  116
            ELSE % NOT END OF SORT                                           17091700  P  116
              IF NOT MATCH(LASTADDRESS,A[0]) AND A[0].REFIDNOF # 0 AND       17091800  P  116
                 A[0].REFIDNOF ≥ XREFAY1[8].IDNOF THEN                       17091900  P  116
                IF A[0].TYPEREF = FORWARDREF THEN %                          17092000  P  116
                  WAITINGFORFWDREF := TRUE                                   17092100  P  116
ELSE                                                                         17092200  P  116
  IF A[0].TYPEREF = LBLREF THEN %                                            17092300  P  116
    BEGIN                                                                    17092400  P  116
      LBLTOG := TRUE;                                                        17092500  P  116
      FWDSEQNO := A[0].SEQNOF;                                               17092600  P  116
    END                                                                      17092700  P  116
  ELSE                                                                       17092800  P  116
    IF A[0].TYPEREF = DECLREF THEN                                           17092900  P  116
      IF WAITINGFORFWDREF THEN % THIS MUST BE IT                             17093000  P  116
        BEGIN                                                                17093100  P  116
          WAITINGFORFWDREF := FALSE;                                         17093200  P  116
          FWDTOG := TRUE;                                                    17093300  P  116
          FWDSEQNO := A[0].SEQNOF;                                           17093400  P  116
        END                                                                  17093500  P  116
      ELSE % ITS A NORMAL DECLARATION - NOT FORWARD                          17093600  P  116
        BEGIN                                                                17093700  P  116
          IF A[0].REFIDNOF > XREFAY1[8].IDNOF THEN                           17093850  P  116
            DO                                                               17093900  P  116
              READ(DSK1,10,XREFAY1[*]) [EOF2]                                17093950  P  116
            UNTIL                                                            17094000  P  116
              A[0].REFIDNOF ≤ XREFAY1[8].IDNOF;                              17094050  P  116
          IF A[0]. REFIDNOF < XREFAY1[8].IDNOF THEN                          17094100  P  116
            GO TO SKIP;                                                      17094150  P  116
          IF XREFPT > 0 THEN % THERE IS STUFF TO PRINT                       17094200  P  116
           BEGIN                                                             17094240  P  116
            IF SINGLTOG THEN                                                 17094250  P  116
              WRITE(LINE,15,PAY[*])                                          17094300  P  116
            ELSE                                                             17094350  P  116
```

```
          WRITE(LINE[DBL],15,PAY[*J);                          17094400  P  116
        LINECOUNT := LINECOUNT + 1;                            17094410  P  116
       END                                                     17094420  P  116
      ELSE                                                     17094450  P  116
        IF NOT SINGLTOG THEN                                   17094500  P  116
          WRITE(LINE);                                         17094550  P  116
      XREFPT := 0;                                             17094600  P  116
      BLANKET(PAY[*]);                                         17094650  P  116
       SETUPHEADING(XREFAY1[*],PAY[*],XREFAY1[8],              17094700  P  116
         SEGNOF,A[0],SEQNOF,FWDTOG,LBLTOG,                     17094800  P  116
        FWDSEQNO,IDTYPE[(IF (I :=                              17094900  P  116
        XREFAY1[9],CLASS) > FAULTID THEN                       17095000  P  116
        (IDMAX + I - FAULTID + 1) ELSE                         17095100  P  116
        IF I > IDMAX THEN 0 ELSE I) x 4],                      17095200  P  116
        REAL(I ≥ BOOID AND XREFAY1[9],[9:2] = 1),              17095300  P  116
        REAL((I ≥ BOOID OR I = LOCLID) AND BOOLEAN             17095310  P  116
        (XREFAY1[9],[9:1])), XREFAY1[9],[10:1]);               17095320  P  116
      FWDTOG := LBLTOG := FALSE;                               17095400  P  116
      WRITE(LINE,15,PAY[*]);                                   17095500  P  116
      LINECOUNT := LINECOUNT + 1;                              17095510  P  116
      BLANKET(PAY[*]);                                         17095550  P  116
    END                                                        17095600  P  116
  ELSE % IT MUST BE A NORMAL REFERENCE                         17095700  P  116
    IF A[0],SEQNOF ≠ LASTADDRESS,SEQNOF THEN                   17095750  P  116
      BEGIN                                                    17095800  P  116
        ADDASEQNO(A[0],SEQNOF,XREFPT,A[0],[5:1],               17095900  P  116
                  PAY[*]);                                     17096000  P  116
        IF (XREFPT := XREFPT + 1) = 11 THEN %FULL              17096100  P  116
          BEGIN                                                17096200  P  116
            WRITE(LINE,15,PAY[*]);                             17096300  P  116
            LINECOUNT := LINECOUNT + 1;                        17096350  P  116
            XREFPT := 0;                                       17096400  P  116
            BLANKET(PAY[*]);                                   17096450  P  116
          END                                                  17096500  P  116
        END                                                    17096550  P  116
      ELSE % REFERENCE TO SAME SEQ. NO. SKIP IT                17096575  P  116
  ELSE % THIS IS A REFERENCE TO THE SAME SEQ. NO. - SKIP        17096600  P  116
ELSE % HIT END OF IDENTIFIER FILE - JUST SKIP OVER REFERENCES   17096700  P  116
  EOF2: B2 := TRUE; %  SO SORT CAN GO TO NORMAL EOJ            17096800  P  116
IF NOT B THEN SKIP: LASTADDRESS := A[0];                       17096850  P  116
END OF OUTPUT2;                                                17096900  P  116
  A[0] := 3"777777777777777"; % BIGGEST FLOATING PT. NO.       17114000  P  116
    COMP2 := IF A[0],REFIDNOF < B[0],REFIDNOF THEN % DIF IDS   17117000  P  116
               TRUE                                            17117100  P  116
             ELSE                                              17117200  P  116
               IF A[0],REFIDNOF = B[0],REFIDNOF THEN           17117300  P  116
                 IF A[0],[1:4] LSS B[0],[1:4] THEN             17117400  P  116
                   TRUE                                        17117500  P  116
                 ELSE                                          17117600  P  116
                   IF A[0],[1:4] = B[0],[1:4] THEN             17117700  P  116
                     IF A[0],SEQNOF < B[0],SEQNOF THEN         17117702  P  116
                       TRUE                                    17117704  P  116
                     ELSE                                      17117706  P  116
                       IF A[0],SEQNOF = B[0],SEQNOF THEN       17117708  P  116
                         BOOLEAN(A[0],[5:1])                   17117710  P  116
                       ELSE                                    17117712  P  116
                         FALSE                                 17117714  P  116
                   ELSE                                        17117720  P  116
```

```
                        FALSE                                          17117730  P  116
                      ELSE                                             17117800  P  116
                        FALSE;                                         17117900  P  116
        SAVETIMES(1); % SAVE TIMES FOR START OF REFERENCES SORT        17117910  P  116
        FIRSTTIME := TRUE; % LET OUTPUT PROCEDURE KNOW ABOUT FIRST CAL 17117920  P  116
        SORT(OUTPUT2,INPUT2,0,HV2,COMP2,1,6000);                       17119000  P  116
       UPDATETIMES(2); % UPDATE TIMES FOR PRINTING CROSS REFERENCE     17119100  P  116
       PRINTXREFSTATISTICS;                                            17119200  P  116
$! DATE 2/1/76                                                         99990000  P  116
$! BY JTC - MSA CENTRAL.                                              99990100  P  116
$!                                                                     99990200  P  116
$! THIS PATCH IS NEARLY A COMPLETE REWRITE OF THE ALGOL               99990300  P  116
$! CROSS REFERENCE ROUTINES IN AN ATTEMPT TO REDUCE                   99990400  P  116
$! THE AMOUNT OF PAPER PRODUCED AND ALSO TO PROVICE MORE              99990500  P  116
$! INFORMATION ABOUT THE IDENTIFIERS.  IN PARTICULAR,                 99990600  P  116
$! THE FOLLOWING CHANGES HAVE BEEN MADE:                              99990700  P  116
$!                                                                     99990800  P  116
$! 1). ONE LINE IS LISTED IN THE CROSS REFERENCE FOR EACH             99990900  P  116
$!      IDENTIFIER WHICH INCLUDES THE FOLLOWING INFORMATION:          99991000  P  116
$!        A. THE NAME OF THE IDENTIFIER,                              99991100  P  116
$!        B. THE CLASS OF THE IDENTIFIER, E.G., REAL,                 99991200  P  116
$!           INTEGER, BOOLEAN, PROCEDURE, ETC.                        99991300  P  116
$!        C. IF THE IDENTIFIER IS A FORMAL PARAMETER IT IS            99991400  P  116
$!           MARKED AS A "NAME PARAMETER" OR "VALUE PARAMETER".       99991500  P  116
$!        D. THE SEGMENT AND SEQUENCE NUMBER AT WHICH THE             99991600  P  116
$!           IDENTIFIER IS DECLARED.                                  99991700  P  116
$!        E. IF THE IDENTIFIER IS A PROCEDURE IDENTIFIER, THEN        99991800  P  116
$!           IF THE PROCEDURE IS DECLARED FORWARD THE SEQUENCE        99991900  P  116
$!           NUMBER OF THE FORWARD DECLARATION IS LISTED.             99992000  P  116
$!        F. IF THE IDENTIFIER IS A LABEL IDENTIFIER, THE SEQUENCE    99992100  P  116
$!           NUMBER AT WHICH THE LABEL OCCURS IS LISTED.              99992200  P  116
$!                                                                    99992300  P  116
$! 2). ASSIGNMENTS TO IDENTIFIERS ARE NOW STARRED IN THE              99992400  P  116
$!      CROSS REFERENCE IN THE FOLLOWING ADDITIONAL CASES:            99992500  P  116
$!      A. WHEN A SUBSCRIPTED VARIABLE IS USED IN ASSIGNMENT.         99992600  P  116
$!      B. WHEN AN ARRAY ROW IS USED IN A FILL STATEMENT.             99992700  P  116
$!      C. WHEN AN ARRAY ROW IS USED IN A SEARCH STATEMENT.           99992800  P  116
$!      C. WHEN A SIMPLE VARIABLE IS USED IN A FOR STATEMENT.         99992900  P  116
$!      D. WHEN A LOCAL VARIABLE OR STREAM PARAMETER IS CHANGED       99993000  P  116
$!         BY BEING ASSIGNED THE VALUE OF SI,DI,CI OR TALLY.          99993100  P  116
$!                                                                    99993200  P  116
$! 3). DEFINE PARAMETERS ARE NO LONGER CROSS REFERENCED.              99993300  P  116
$!                                                                    99993400  P  116
$! 4). IDENTIFIERS REFERRED TO IN THE RIGHT HAND SIDE OF A            99993500  P  116
$!      DEFINE DECLARATION ARE NO LONGER CROSS                        99993600  P  116
$!      REFERENCED AGAINST THE DEFINE DECLARATION.                    99993700  P  116
$!                                                                    99993800  P  116
$! 5). IF NO IDENTIFIERS WERE CROSS REFERENCED, THE IDENTIFIER        99993900  P  116
$!      SORT WOULD BLOW UP WITH AN I/O ERROR 86, NOW, XREF            99994000  P  116
$!      CHECKS TO MAKE SURE THERE IS SOMETHING TO SORT BEFORE         99994100  P  116
$!      BEGINNING.                                                    99994200  P  116
$!                                                                    99994300  P  116
$! 6). IT SHOULD NOW BE POSSIBLE TO SET AND RESET XREF                99994400  P  116
$!      INDISCRIMINANTLY DURING A COMPILE WITHOUT CAUSING THE XREF    99994500  P  116
$!      TO GET CONFUSED. PREVIOUSLY, REFERENCES WOULD GET             99994600  P  116
$!      ASSOCIATED WITH THE WRONG IDENTIFIERS. THE RULE NOW IS THAT   99994700  P  116
$!      XREF MUST BE ON AT THE POINT OF DECLARATION FOR AN            99994800  P  116
```

```
$!       IDENTIFIER TO BE CROSS REFFRENCED AT ALL.                          99994900  P  116
$!                                                                          99995000  P  116
$! 7)* PREVIOUSLY, WHEN THE SAME IDENTIFIER OCCURRED IN                     99995100  P  116
$!       DIFFERENT BLOCKS, THE IDENTIFIERS WOULD BE LISTED                  99995200  P  116
$!       IN NO PARTICULAR ORDER. NOW, THE IDENTIFIER THAT                   99995300  P  116
$!       APPEARS FIRST WILL BE LISTED FIRST IN THE XREF.                    99995400  P  116
$!                                                                          99995500  P  116



$#PATCH NUMBER 117 FOR ALGOL CONTAINS 118 CARDS.                           00000000  P  117

 618     BLOCK; AUXMEM APPEARS IMMEDIATELY BEFORE IDENTIFIER (NO TYPE)     00418000  P  117
                 SBITF =[21:6]#, % STARTING BIT FOR FIELD ID.              01154200  P  117
                 NBITF =[27:6]#, % NUMBER OF BITS FOR FIELD ID.            01154300  P  117
         FIELDID        =125#,    COMMENT 175;                            01278090  P  117
         AUXMEMV        =20#,     COMMENT 24;                             01298500  P  117
         FIELDV         =21#;     COMMENT 25;                             01298600  P  117
         LASTSEQUENCE =166#,                                              01575000  P  117
         SORTA         =673#;                                             01580000  P  117
                 IF STEPI = FIELDID THEN % GET INFO FROM INFO              05273100  P  117
                   BEGIN                                                  05273200  P  117
                     FIRST := ELBAT[I].SBITF;                             05273300  P  117
                     SECOND := ELBAT[I].NBITF;                            05273400  P  117
                     GO TO EXIT;                                          05273500  P  117
                   END                                                    05273600  P  117
                 ELSE                                                     05273700  P  117
                   IF ELCLASS = LFTBRKET THEN                             05273800  P  117
                     IF STEPI = FIELDID THEN                              05273900  P  117
                       BEGIN                                              05274000  P  117
                         FIRST := ELBAT[I].SBITF;                         05274100  P  117
                         SECOND := ELBAT[I].NBITF;                        05274200  P  117
                         IF STEPI = RTBRKET THEN                          05274300  P  117
                           GO TO EXIT;                                    05274400  P  117
                       END                                                05274500  P  117
                     ELSE                                                 05274600  P  117
                       IF ELCLASS = LITNO THEN                            05275000  P  117
IF ELCLASS = FIELDID THEN                                                 06313550  P  117
  BEGIN                                                                   06313600  P  117
    FIRST := ELBAT[I].SBITF;                                              06313650  P  117
    SECOND := 48 - (THIRD := ELBAT[I].NBITF);                             06313700  P  117
    GO TO SKIP1;                                                          06313750  P  117
  END                                                                     06313800  P  117
ELSE                                                                      06313850  P  117
IF STEPI = FIELDID THEN                                                   06314050  P  117
  BEGIN                                                                   06314100  P  117
    FIRST := ELBAT[I].SBITF;                                              06314150  P  117
    SECOND := 48 - (THIRD := ELBAT[I].NBITF);                             06314200  P  117
    IF STEPI # RTBRKET THEN                                               06314250  P  117
      BEGIN                                                               06314300  P  117
        ERR(94);                                                          06314350  P  117
        GO TO EXIT;                                                       06314400  P  117
      END;                                                                06314450  P  117
    GO TO SKIP1;                                                          06314500  P  117
  END                                                                     06314550  P  117
ELSE                                                                      06314600  P  117
```

```
          IF ELCLASS ≠ LITNO THEN % PREPARE FOR DYNAMIC DIAL                    06314650  P  117
            GO TO L1;                                                           06314700  P  117
    OCTO430000250000000, "5FIELD",                                  %671 09214432  P  117
                      0, ">SORT ", "TEMPORAR", "Y0000000", % SORTA  %673 09214435  P  117
    "       " ;          COMMENT LASTSEQUENCE,LASTSEQROW ;          %674 09214440  P  117
639 STEP 3 UNTIL 660,663 STEP 2 UNTIL 667, 671                              09214516  P  117
      GOTSCHK,FIELDDEC,AUXMEMERR,                                           14016000  P  117
              STREAMERR,DEFINEDEC,AUXMEMERR,FIELDDEC;                       14021000  P  117
AUXMEMERR:FLAG(618);J:=J+1;GO TO REALDEC;                                   14136100  P  117
FIELDDEC:                                                                   14269020  P  117
  BEGIN                                                                     14269040  P  117
    REAL SAVEINFO, SB, NB;                                                  14269060  P  117
    BOOLEAN FOUNDLB; % TRUE IF LEFT-BRACKET WAS USED IN FIELD SPEC.         14269080  P  117
    LABEL EXIT, SAVEIT;                                                     14269100  P  117
    STOPENTRY := STOPGSP := TRUE;                                           14269120  P  117
    I := I - 1;                                                             14269140  P  117
    DO                                                                      14269160  P  117
      BEGIN                                                                 14269180  P  117
        STOPDEFINE := TRUE;                                                 14269200  P  117
        STEPIT;                                                             14269220  P  117
        ENTRY(FIELDID);                                                     14269240  P  117
        SAVEINFO := LASTINFO;                                               14269260  P  117
        IF ELCLASS = RELOP AND ACCUM[1] = "1=0000" THEN                     14269280  P  117
          BEGIN                                                             14269300  P  117
            IF STEPI = LFTBRKET THEN % REMEMBER THIS                        14269320  P  117
              BEGIN                                                         14269340  P  117
                FOUNDLB := TRUE;                                            14269360  P  117
                STEPIT;                                                     14269380  P  117
              END                                                           14269400  P  117
        ELSE                                                                14269420  P  117
          FOUNDLB := FALSE;                                                 14269440  P  117
        IF ELCLASS = FIELDID THEN                                           14269442  P  117
          BEGIN                                                             14269444  P  117
            SB := ELBAT[I].SBITF;                                           14269446  P  117
            NB := ELBAT[I].NBITF;                                           14269448  P  117
            GO TO SAVEIT;                                                   14269450  P  117
          END;                                                              14269452  P  117
        IF ELCLASS = LITNO THEN                                             14269460  P  117
          IF STEPI = COLON THEN                                             14269480  P  117
            IF STEPI = LITNO THEN                                           14269500  P  117
              IF (SB := ELBAT[I-2].ADDRESS) ×                               14269520  P  117
                 (NB := ELBAT[I].ADDRESS) ≠ 0 AND                           14269540  P  117
                 SB + NB ≤ 48 THEN                                          14269560  P  117
                BEGIN                                                       14269580  P  117
          SAVEIT:                                                           14269590  P  117
                  PUT(TAKE(SAVEINFO) & SB SBITF & NB NBITF,                 14269600  P  117
                       SAVEINFO);                                           14269620  P  117
                  STEPIT;                                                   14269640  P  117
                  IF FOUNDLB THEN % BETTER HAVE RIGHT BRACKET.              14269660  P  117
                    IF ELCLASS = RTBRKET THEN                               14269680  P  117
                      BEGIN                                                 14269700  P  117
                        STEPIT;                                             14269705  P  117
                        GO TO EXIT;                                         14269710  P  117
                      END                                                   14269715  P  117
                    ELSE                                                    14269720  P  117
                  ELSE                                                      14269740  P  117
                    GO TO EXIT;                                             14269760  P  117
```

```
                    END;                                                                14269780  P  117
                END;                                                                    14269800  P  117
              FLAG(114);                                                                14269820  P  117
              DO STEPIT UNTIL ELCLASS = COMMA OR ELCLASS = SEMICOLON;                    14269840  P  117
        EXIT;                                                                           14269860  P  117
          END                                                                           14269880  P  117
        UNTIL                                                                           14269900  P  117
          ELCLASS ≠ COMMA;                                                              14269920  P  117
        STOPENTRY := STOPGSP := FALSE;                                                  14269940  P  117
      END;                                                                              14269960  P  117
      GO TO START;                                                                      14269980  P  117
        ARRAY IDTYPE[0:(IDMAX+4)x4-1];                                                  17047100  P  117
              "FIELD.                                 ", % 30 (CLASS = 125)          17084275  P  117
              "FAULT.                                 ", % 32 (CLASS = 126)          17084280  P  117
              "SWITCH LIST.                           "; % 31 (CLASS = 127)          17084290  P  117
                              XREFAY1[9].CLASS) ≥ FIELDID THEN                          17095000  P  117
                              (IDMAX + I - FIELDID + 1) ELSE                            17095100  P  117
$: DATE 2/2/76                                                                          99990000  P  117
$: BY JTC - MSA CENTRAL                                                                 99990100  P  117
$:                                                                                      99990200  P  117
$: THIS PATCH IMPLEMENTS A NEW DECLARATION TYPE OF "FIELD" WHICH                        99990300  P  117
$: PROVIDES A CONVENIENT METHOD OF DESIGNATING A PARTIAL WORD FIELD                     99990400  P  117
$: A LA THE 6700 ESPOL AND ALGOL MECHANISM. THIS METHOD IS CONSIDERABLY                 99990500  P  117
$: MORE EFFICIENT THAN A DEFINE BECAUSE THE PARTIAL WORD FIELD INFOR-                   99990600  P  117
$: MATION IS STORED OFF WITH THE IDENTIFIER IN THE SYMBOL TABLE THUS                    99990700  P  117
$: MAKING REFERENCES TO THE PARTIAL WORD FIELD VERY INEXPENSIVE.                        99990800  P  117
$:                                                                                      99990900  P  117
$: THE SYNTAX FOR A PARTIAL WORD DECLARATION IS AS FOLLOWS:                             99991000  P  117
$:                                                                                      99991100  P  117
$:  <FIELD DECLARATION> ::= FIELD <FIELD LIST>                                          99991200  P  117
$:                                                                                      99991300  P  117
$:  <FIELD LIST> ::= <FIELD> / <FIELD LIST> , <FIELD>                                   99991400  P  117
$:                                                                                      99991500  P  117
$:  <FIELD> ::= <FIELD IDENTIFIER> = <FIELD SPECIFICATION>                              99991600  P  117
$:                                                                                      99991700  P  117
$:  <FIELD IDENTIFIER> ::= <IDENTIFIER>                                                 99991800  P  117
$:                                                                                      99991900  P  117
$:  <FIELD SPECIFICATION> ::= <INTEGER> : <INTEGER> /                                   99992000  P  117
$:                            [ <INTEGER> : <INTEGER> ] /                               99992100  P  117
$:                            <FIELD IDENTIFIER> /                                      99992200  P  117
$:                            [ <FIELD IDENTIFIER> ]                                    99992300  P  117
$:                                                                                      99992400  P  117
$: SOME EXAMPLES:                                                                       99992500  P  117
$:                                                                                      99992600  P  117
$:       FIELD CF = [33:15];                                                            99992700  P  117
$:                                                                                      99992800  P  117
$:       FIELD FF = 18:15,                                                              99992900  P  117
$:           SIZE = 08:10;                                                              99993000  P  117
$:                                                                                      99993100  P  117
$: FIELDS MAY BE USED ANYWHERE PARTIAL WORD DESIGNATORS MAY BE USED                     99993200  P  117
$: AND MAY BE ENCLOSED IN BRACKETS OR NOT AS DESIRED. ENCLOSING                         99993300  P  117
$: THE FIELD IDENTIFIER IN BRACKETS MAY HELP THE READABILITY OF                         99993400  P  117
$: THE CODE BUT OF COURSE REQUIRES EXTRA SCANNING  BY THE                               99993500  P  117
$: COMPILER TO PROCESS THE BRACKETS.                                                    99993600  P  117
$:                                                                                      99993700  P  117
$: SOME EXAMPLES:                                                                       99993800  P  117
$:                                                                                      99993900  P  117
$:        I := I,FF;                                                                    99994000  P  117
```

```
$!                                                                          99994100  P  117
$!       J := I.CF & 10 [SIZE] & K FF;                                      99994200  P  117
$!                                                                          99994300  P  117
$!       I.FF := J.[CF];                                                    99994400  P  117
$!                                                                          99994500  P  117
$!*************************************************************************99999999  P  117



$#PATCH NUMBER 118 FOR ALGOL CONTAINS 22 CARDS.                             00000000  P  118

      GLOBALNINFOO,COMMENT MAINTAINS VALUE OF NINFOO FROM BLOCK ON A        01620100  P  118
                           GLOBAL LEVEL SO THAT THE PROCEDURE "ENTRY"      01620200  P  118
                           CAN CHECK FOR DUPLICATE DECLARATIONS;           01620300  P  118
      OLDNINFOO,    COMMENT REMEMBERS OLD VALUE OF GLOBALNINFOO;          01620400  P  118
                 IF ELCLASS = DEFINEDID THEN % CHECK IF NEW DECLARATION    13339000  P  118
                   IF NOT (PTOG OR STREAMTOG) AND LINKF ≥ GLOBALNINFOO     13339100  P  118
                     THEN FLAG(1)                                          13339200  P  118
                   ELSE                                                    13339300  P  118
                 ELSE                                                      13339400  P  118
                   IF LEVELF = LEVEL THEN % DUPLICATE DECLARATION          13339500  P  118
                     FLAG(1);                                              13340000  P  118
     SAVEL0 := SAVEL;                                                      14055000  P  118
     AJUMP := FALSE;     % NO PENDING JUMPS IN THIS BLOCK YET.             14055100  P  118
     L := 0;             % START GENERATING CODE AT WORD 0, SYLLABLE 0.     14055200  P  118
     OLDNINFOO := GLOBALNINFOO;  % REMEMBER WHERE PREVIOUS BLOCKS           14055250  P  118
                                 % SYMBOLS BEGAN IN SYMBOL TABLE.           14055260  P  118
     GLOBALNINFOO := NINFOO := NEXTINFO;  % REMEMBER WHERE THE SYMBOLS       14055300  P  118
                                          % FROM THIS BLOCK WILL GO         14055400  P  118
                                          % IN THE SYMBOL TABLE.            14055450  P  118
          REAL J,K,DINFO,LINKA,LINKB;                                       14254100  P  118
$ VOIDT 14259005                                                            14259002  P  118
    GLOBALNINFOO := OLDNINFOO;                                              14606100  P  118
$! DATE 2/2/76                                                             99990000  P  118
$! BY JTC - MSA CENTRAL.                                                   99990100  P  118
$!                                                                         99990200  P  118
$! THIS PATCH PREVENTS AN IDENTIFIER FROM BEING USED AS BOTH A DEFINE      99990300  P  118
$! AND SOMETHING ELSE AT THE SAME BLOCK LEVEL.  WHEN THE IDENTIFIER        99990400  P  118
$! WAS DECLARED A SECOND TIME IN THE BLOCK, SINCE DEFINES ARE DISABLED,    99990500  P  118
$! THE DUPLICATE DECLARATION WAS NOT CORRECTLY RECOGNIZED.  NOW, NO        99990600  P  118
$! SYMBOL MAY BE DECLARED TWICE IN THE SAME BLOCK NO MATTER WHAT.          99990700  P  118
$!                                                                         99990800  P  118



$#PATCH NUMBER 119 FOR ALGOL CONTAINS 4 CARDS.                              00000000  P  119

          CLOSE(CARD,RELEASE);  % RELEASE PRIMARY INPUT FILE.               09407200  P  119
          CLOSE(TAPE,RELEASE);  % RELEASE SECONDARY INPUT FILE.             09407300  P  119
          LOCK(NEWTAPE,*);  % CLOSE WITH CRUNCH.                          09407400  P  119
$VOIDT 17002511                                                             17002490  P  119
$! DATE 2/2/76                                                             99990000  P  119
$! BY JTC - MSA CENTRAL.                                                   99990100  P  119
$!                                                                         99990200  P  119
$! THIS PATCH CAUSES THE COMPILER TO CLOSE THE NEWTAPE FILE WITH           99990300  P  119
```

```
$1 CRUNCH.                                                          99990400  P  119



$#PATCH NUMBER 120 FOR ALGOL CONTAINS 3 CARDS.                      00000000  P  120

    IF MERGETOG THEN % INDICATE NAME OF SOURCE FILE.                01835600  P  120
      WRITE(LINE,<X40,"SOURCE FILE: ",A1,A6,"/",A1,A6,//>,          01835700  P  120
        (N1:=TAPE.MFID).[6:6],N1,(N2:=TAPE.FID).[6:6],N2);          01835800  P  120
$1 DATE 2/6/76                                                      99990000  P  120
$1 BY JTC - MSA CENTRAL.                                            99990100  P  120
$1                                                                  99990200  P  120
$1 THIS PATCH PRINTS THE NAME OF THE SOURCE FILE BELOW              99990300  P  120
$1 THE NAME OF THE OBJECT FILE ON THE PRINTED LISTING IF TAPE       99990400  P  120
$1 IS SET AT THE TIME THE HEADING IS PRINTED.                       99990500  P  120



$#PATCH NUMBER 121 FOR ALGOL CONTAINS 2 CARDS.                      00000000  P  121

 $SET OMIT LISTA = LIST                                             00000999  P  121
 $POP OMIT LISTA                                                    00499999  P  121
$1 DATE 3/2/76                                                      99990000  P  121
$1 BY JTC - MSA CENTRAL                                             99990100  P  121
$1                                                                  99990200  P  121
$1 THIS PATCH SETS OMIT AROUND THE COMMENT AT THE BEGINNING OF THE  99990300  P  121
$1 ALGOL COMPILER TO SPEED UP SCANNING OF THE COMMENT.              99990400  P  121
$1*****************************************************************99999999  P  121
```

```
$#PATCH NUMBER 122 FOR ALGOL CONTAINS 14 CARDS.                     00000000  P  122

        LASTUSED:=(T:=DEFINEARRAY[DEFINEINDEX:=DEFINEINDEX-3]).[33:15];02721000  P  122
        IF (GT2 := T.[18:15]) # 0 THEN % THIS WAS A PARAMETRIC DEFINE  02721500  P  122
          BEGIN % PURGING PARAMETERS FROM DEFSTACKHEAD                 02722000  P  122
            GT2 := TAKE(GT2).LINK; % GET POINTER TO NEW DEFSTACKHEAD   02722500  P  122
            DO                                                         02723000  P  122
              PUT(TEXT[(NEXTTEXT:=(GT1:=TAKE(DEFSTACKHEAD)),DYNAM-1)   02723500  P  122
                    ,LINKR,NEXTTEXT.LINKC],DEFSTACKHEAD)               02724000  P  122
                % THIS RESTORES THE PREVIOUS ELBAT WORD FOR             02724500  P  122
                % THIS PARAMETER IN CASE OF NESTED DEFINE.              02725000  P  122
            UNTIL                                                      02725500  P  122
              GT2 = (DEFSTACKHEAD := GT1.LINK);                        02726000  P  122
          END;                                                         02727000  P  122
                TEXT[NEXTTEXT,LINKR,NEXTTEXT,LINKC] := TAKE(P);        12117000  P  122
                NEXTTEXT := NEXTTEXT + 1;                              12118000  P  122
$1 DATE 3/5/76                                                         99990000  P  122
$1 BY JTC - MSA CENTRAL                                                99990100  P  122
$1                                                                     99990200  P  122
$1 THIS PATCH ALLOWS RECURSIVE REFERENCES TO THE SAME DEFINE           99990300  P  122
$1 TO WORK CORRECTLY, FOR EXAMPLE, THE FOLLOWING DEFINE                99990400  P  122
$1                                                                     99990500  P  122
$1                DEFINE A(A1,A2) = A1 + A2#                           99990600  P  122
$1                                                                     99990700  P  122
```

```
$! WOULD YIELD THE WRONG RESULTS FOR THE FOLLOWING INVOCATION:                99990800  P  122
$!                                                                           99990900  P  122
$!                    A(A(1,2),3)                                            99991000  P  122
$!                                                                           99991100  P  122
$! THE RESULT WOULD HAVE BEEN 5 INSTEAD OF 6. RECURSIVE REFERENCES           99991200  P  122
$! ARE NOW HANDLED CORRECTLY.                                                99991300  P  122
$!*************************************************************************99999999  P  122



$#PATCH NUMBER 123 FOR ALGOL CONTAINS 3 CARDS.                               00000000  P  123

%                        MARK XV1.0.122                                      00001030  P  123
%                         MAY  9, 1977                                       00001040  P  123
"XV1.0.122"                                                                  01831000  P  123
$! BY DJZ - MSC DETROIT                                                      99990000  P  123
$! DATE 05/09/77                                                            99990100  P  123
$! THIS PATCH UPDATES THE MARK LEVELS IN THE SYMBOL FILE.                    99990200  P  123
$!*************************************************************************99990300  P  123



NUMBER OF ERRORS DETECTED =  0.
PROCESSOR TIME =    53 SECONDS.
I/O TIME =    33 SECONDS.
```

TOO MANY CHARACTERS OR SOMETHING, COMPILER MAY

```
LABEL  000000000LINE   00177280?EXECUTE PATCH/MERGE
```

```
? COMPILE ALGOL/NUDISK ALGOL LIBRARY

        PACKET   337
        INPUT   1044 CARDS FROM ZIP
        TIME    1600
        DATE   77280 FRIDAY, 10/07/77



 *** BURROUGHS B5700 DCMCP MARK XVI.0.178  AND INTRINSICS MARK XVI.0.132 ***

     #NO MESSAGES TODAY

16:00:56  ? COMPILE ALGOL/NUDISK ALGOL LIBRARY
16:00:56      ? ALGOL STACK= 1000
16:00:56      ? ALGOL FILE TAPE= SYMBOL/ALGOL DISK SERIAL
16:00:57      ? FILE LINE= LINE PRINT OR BACK UP
16:00:57  ? DATA CARD
16:00:57          4:ALGOL/ALGOL= 2 BOJ 1600 05/12/77
16:00:59          CDB IN CARD DB:ALGOL/ALGOL= 2
16:01:01          DKA IN SER SYMBOL ALGOL:ALGOL/ALGOL= 2
16:01:01          PBD0338 OUT 011 LINE:ALGOL/ALGOL= 2
16:01:12          DKA OUT SER DSK1 SITE:ALGOL/ALGOL= 2
16:01:13          DKA OUT SER DSK2 SITE:ALGOL/ALGOL= 2
16:01:55          DKA OUT RDM ALGOL NUDISK:ALGOL/ALGOL= 2
16:06:53          PBD0338 OUT 012:ALGOL/ALGOL= 2
16:09:49          CDB REL CARD DB:ALGOL/ALGOL= 2
16:09:50  ? END.
16:09:50          DKA REL SYMBOL ALGOL:ALGOL/ALGOL= 2
16:09:53          DKA OUT SER DSRT1 SITE:ALGOL/ALGOL= 2
16:10:34          DKA OUT SER DSRT2 SITE:ALGOL/ALGOL= 2
16:11:08          DKA REL DSRT1 SITE:ALGOL/ALGOL= 2
16:11:08          DKA REL DSRT2 SITE:ALGOL/ALGOL= 2
16:11:08          DKA OUT SER DSRT1 SITE:ALGOL/ALGOL= 2
16:13:23          DKA OUT SER DSRT2 SITE:ALGOL/ALGOL= 2
16:17:10          DKA REL DSRT1 SITE:ALGOL/ALGOL= 2
16:17:10          DKA REL DSRT2 SITE:ALGOL/ALGOL= 2
16:17:12          DKA REL DSK2 SITE:ALGOL/ALGOL= 2
16:17:13          DKA REL DSK1 SITE:ALGOL/ALGOL= 2
16:17:14          PBD0338 REL 012 LINE 19698:ALGOL/ALGOL= 2
16:17:15          DKA REL ALGOL NUDISK:ALGOL/ALGOL= 2
16:17:17          ALGOL/ALGOL= 2 SYNTX ERR 1617
16:17:20          FOR ALGOL/ALGOL= 2: PROCESS= 736 SECS, IO= 369 SECS, OLAY= 9
16:17:21          PKT#0337 REMOVED
```

ALGOL /NUDISK

SOURCE FILE: SYMBOL /ALGOL

```
                $SET OMIT LISTA = LIST                                          %121-       00000999 C  0001:000010
!OM!T!          %#######################################################################    00001000 T    !OMIT!
!OM!T!          %                                                                          00001010 T    !OMIT!
!OM!T!          %              B-5700 ALGOL/TSPOL SYMBOLIC                                 00001020 T    !OMIT!
!OM!T!          %                    MARK XVI.0.122                             %123-       00001030 P    !OMIT!
!OM!T!          %                     MAY  9, 1977                              %123-       00001040 P    !OMIT!
!OM!T!          %                                                                          00001050 T    !OMIT!
!OM!T!          %#######################################################################    00001060 T    !OMIT!
!OM!T!          %                                                                          00001070 T    !OMIT!
!OM!T!            COMMENT: * TITLE: B5500/B5700 MARK XVI SYSTEM RELEASE          *        00001072 T    !OMIT!
!OM!T!                     * FILE ID: SYMBOL/ALGOL TAPE ID: SYMBOL1/FILE000      *        00001073 T    !OMIT!
!OM!T!                     * THIS MATERIAL IS PROPRIETARY TO BURROUGHS CORPORATION *        00001074 T    !OMIT!
!OM!T!                     * AND IS NOT TO BE REPRODUCED, USED, OR DISCLOSED     *        00001075 T    !OMIT!
!OM!T!                     * EXCEPT IN ACCORDANCE WITH PROGRAM LICENSE OR UPON   *        00001076 T    !OMIT!
!OM!T!                     * WRITTEN AUTHORIZATION OF THE PATENT DIVISION OF     *        00001077 T    !OMIT!
!OM!T!                     * BURROUGHS CORPORATION, DETROIT, MICHIGAN 48232      *        00001078 T    !OMIT!
!OM!T!                     *                                                     *        00001079 T    !OMIT!
!OM!T!                     * COPYRIGHT (C)  1965, 1971, 1972, 1974             *        00001080 T    !OMIT!
!OM!T!                     *       BURROUGHS CORPORATION                       *        00001081 T    !OMIT!
!OM!T!                     *  AA759915   AA320206   AA393180   AA332366        *;       00001082 T    !OMIT!
!OM!T!          COMMENT####################################################################    00001110 T    !OMIT!
!OM!T!                     ERROR MESSAGES                                                  00001120 T    !OMIT!
!OM!T!          ###########################################################################    00001130 T    !OMIT!
!OM!T!          %                                                                          00001140 T    !OMIT!
!OM!T!          ERROR NUMBER  ROUTINE:ERROR MESSAGE                                        00002000 T    !OMIT!
!OM!T!           000          BLOCK:  DECLARATION NOT FOLLOWED BY SEMICOLON.             00003000 T    !OMIT!
!OM!T!           001          BLOCK:  IDENTIFIER DECLARED TWICE IN SAME BLOCK.           00004000 T    !OMIT!
!OM!T!           002          PROCEDUREDEC: SPECIFICATION PART CONTAINS                  00005000 T    !OMIT!
!OM!T!                                     IDENTIFIER NOT APPEARING IN                 00006000 T    !OMIT!
!OM!T!                                     FORMAL PARAMETER PART.                      00007000 T    !OMIT!
!OM!T!           003          BLOCK:  NON-IDENTIFIER APPEARS IN IDENTIFIER               00008000 T    !OMIT!
!OM!T!                                  LIST OF DECLARATION.                           00009000 T    !OMIT!
!OM!T!           004          PROCEDUREDEC: STREAM PROCEDURE DECLARATION                 00010000 T    !OMIT!
!OM!T!                                     PRECEDED BY ILLEGAL DECLARATOR.             00011000 T    !OMIT!
!OM!T!           005          PROCEDUREDEC: PROCEDURE DECLARATION PRECEDED               00012000 T    !OMIT!
!OM!T!                                     BY ILLEGAL DECLARATOR.                      00013000 T    !OMIT!
!OM!T!           006          PROCEDUREDEC: PROCEDURE IDENTIFIER USED BEFORE             00014000 T    !OMIT!
!OM!T!                                     IN SAME BLOCK(NOT FORWARD).                 00015000 T    !OMIT!
!OM!T!           007          PROCEDUREDEC: PROCEDURE IDENTIFIER NOT FOLLOWED            00016000 T    !OMIT!
!OM!T!                                     BY ( OR SEMICOLON IN PROCEDURE              00017000 T    !OMIT!
!OM!T!                                     DECLARATION.                                00018000 T    !OMIT!
!OM!T!           008          PROCEDUREDEC: FORMAL PARAMETER LIST NOT FOLLOWED           00019000 T    !OMIT!
!OM!T!                                     BY ).                                       00020000 T    !OMIT!
!OM!T!           009          PROCEDUREDEC: FORMAL PARAMETER PART NOT FOLLOWED           00021000 T    !OMIT!
```

```
!OMIT!                                  BY SEMICOLON.                                                    00022000 T    !OMIT!
!OMIT!          010     PROCEDUREDEC!   VALUE PART CONTAINS IDENTIFIER                                   00023000 T    !OMIT!
!OMIT!                                  WHICH DID NOT APPEAR IN FORMAL                                   00024000 T    !OMIT!
!OMIT!                                  PARAPART.                                                        00025000 T    !OMIT!
!OMIT!          011     PROCEDUREDEC!   VALUE PART NOT ENDED BY SEMICOLON.                               00026000 T    !OMIT!
!OMIT!          012     PROCEDUREDEC!   MISSING OR ILLEGAL SPECIFICATION                                  00027000 T    !OMIT!
!OMIT!                                  PART.                                                            00028000 T    !OMIT!
!OMIT!          013     PROCEDUREDEC!   OWN, SAVE, OR AUXMEM USED IN                                      00029000 T    !OMIT!
!OMIT!                                  ARRAY SPECIFICATION.                                             00029500 T    !OMIT!
!OMIT!          014     ARRAYDEC! AUXMEM AND SAVE ARE MUTUALLY EXCLUSIVE.                                00030000 T    !OMIT!
!OMIT!          015     ARRAYDEC!       ARRAY CALL-BY-VALUE NOT IMPLEMENTED.                             00030500 T    !OMIT!
!OMIT!                                                                                                   00031000 T    !OMIT!
!OMIT!          016     ARRAYDEC!       ARRAY ID IN DECLARATION NOT FOLLOWED                             00032000 T    !OMIT!
!OMIT!                                  BY [ .                                                           00033000 T    !OMIT!
!OMIT!          017     ARRAYDEC!       LOWER BOUND IN ARRAY DEC NOT                                     00034000 T    !OMIT!
!OMIT!                                  FOLLOWED BY : .                                                  00035000 T    !OMIT!
!OMIT!          018     ARRAYDEC!       BOUND PAIR LIST NOT FOLLOWED BY ].                               00036000 T    !OMIT!
!OMIT!          019     ARRAYSPEC!      ILLEGAL LOWER BOUND DESIGNATOR IN                                00037000 T    !OMIT!
!OMIT!                                  ARRAY SPECIFICATION.                                             00038000 T    !OMIT!
!OMIT!          020     BLOCK!          OWN APPEARS IMMEDIATELY BEFORE                                   00039000 T    !OMIT!
!OMIT!                                  IDENTIFIER(NO TYPE).                                             00040000 T    !OMIT!
!OMIT!          021     BLOCK!          SAVE APPEARS IMMEDIATELY BEFORE                                  00041000 T    !OMIT!
!OMIT!                                  IDENTIFIER(NO TYPE).                                             00042000 T    !OMIT!
!OMIT!          022     BLOCK!          STREAM APPEARS IMMEDIATELY BEFORE                                00043000 T    !OMIT!
!OMIT!                                  IDENTIFIER(THE WORD PROCEDURE LEFT                               00044000 T    !OMIT!
!OMIT!                                  OUT).                                                            00045000 T    !OMIT!
!OMIT!          023     BLOCK!          DECLARATOR PRECEDED ILLEGALLY BY                                 00046000 T    !OMIT!
!OMIT!                                  ANOTHER DECLARATOR.                                              00047000 T    !OMIT!
!OMIT!          024     PROCEDUREDEC!   LABEL CANNOT BE PASSED TO FUNCTION.                               00048000 T    !OMIT!
!OMIT!          025     BLOCK!          DECLARATOR OR SPECIFIER ILLEGALLY                                00049000 T    !OMIT!
!OMIT!                                  PRECEDED BY OWN OR SAVE OR SOME                                  00050000 T    !OMIT!
!OMIT!                                  OTHER DECLARATOR.                                                00051000 T    !OMIT!
!OMIT!          026     FILEDEC!        MISSING ( IN FILE DEC.                                           00052000 T    !OMIT!
!OMIT!          027     FILEDEC!        MISSING RECORD SIZE.                                             00053000 T    !OMIT!
!OMIT!                                                                                                   00054000 T    !OMIT!
!OMIT!          028     FILEDEC!        ILLEGAL BUFFER PART OR SAVE FACTOR                               00055000 T    !OMIT!
!OMIT!                                  IN FILE DEC.                                                     00056000 T    !OMIT!
!OMIT!          029     FILEDEC!        MISSING ) IN FILE DEC.                                           00057000 T    !OMIT!
!OMIT!          030     IODEC!          MISSING COLON IN DISK DESCRIPTION.                               00058000 T    !OMIT!
!OMIT!                                                                                                   00059000 T    !OMIT!
!OMIT!          031     LISTDEC!        MISSING ( IN LISTDEC.                                            00060000 T    !OMIT!
!OMIT!          032     FORMATDEC!      MISSING ( IN FORMAT DEC.                                         00061000 T    !OMIT!
!OMIT!          033     SWITCHDEC!      SWITCH DEC DOES NOT HAVE ← OR                                    00062000 T    !OMIT!
!OMIT!                                  FORWARD AFTER IDENTIFIER.                                        00063000 T    !OMIT!
!OMIT!          034     SWITCHFILEDEC!  MISSING ← AFTER FILED.                                           00064000 T    !OMIT!
!OMIT!          035     SWITCHFILEDEC!  NON FILE ID APPEARING IN DECLARATION                             00065000 T    !OMIT!
!OMIT!                                  OF SWITCHFILE.                                                   00066000 T    !OMIT!
!OMIT!          036     SUPERFORMATDEC! FORMAT ID NOT FOLLOWED BY ←   ,                                  00067000 T    !OMIT!
!OMIT!          037     SUPERFORMATDEC! MISSING ( AT START OF FORMATPHRASE ,                             00068000 T    !OMIT!
!OMIT!          038     SUPERFORMATDEC! FORMAT SEGMENT >1022 WORDS.                                      00069000 T    !OMIT!
!OMIT!          039     BLOCK!  NUMBER OF NESTED BLOCKS IS GREATER THAN 31                               00069100 T    !OMIT!
!OMIT!          040     IODEC!  PROGRAM PARAMETER BLOCK SIZE EXCEEDED                                    00069200 T    !OMIT!
!OMIT!          041 HANDLESWLIST!  MISSING ← AFTER SWITCH LIST ID.                                       00069300 T    !OMIT!
!OMIT!          042 HANDLESWLIST!  ILLEGAL LIST ID APPEARING IN SWITCH LIST.                             00069400 T    !OMIT!
!OMIT!          043 IODEC!  MISSING ] AFTER DISK IN FILEDEC.                                             00069500 T    !OMIT!
!OMIT!          044 IODEC!   MISSING [ AFTER DISK IN FILEDEC.                                           00069600 T    !OMIT!
!OMIT!         045      DEFINEDEC! MISSING "=" AFTER DEFINE ID.                                          00069700 T    !OMIT!
!OMIT!         046      ARRAE!  NON-LITERAL ARRAY BOUND NOT GLOBAL TO ARRAY DECL.                        00069800 T    !OMIT!
```

```
!OMIT!        047     TABLE: ITEM FOLLOWING @ NOT A NUMBER.                                    00069900 T    !OMIT!
!OMIT!        048:    PROCEDUREDEC: NUMBER OF PARAMETERS DIFFERS FROM FWD DECL.                00069910 T    !OMIT!
!OMIT!        049:    PROCEDUREDEC: CLASS OF PARAMETER DIFFERS FROM FWD DECL.                  00069920 T    !OMIT!
!OMIT!        050:    PROCEDUREDEC: VALUE PART DIFFERS FROM FWD DECL.                          00069930 T    !OMIT!
!OMIT!        051     SAVEPROC : FORWARD DECLARATION DOES NOT AGREE WITH                       00069931 T    !OMIT!
!OMIT!                    ACTUAL DECLARATION                                                   00069932 T    !OMIT!
!OMIT!        052     SAVEPROC :STATEMENT MAY NOT START WITH THIS KIND OF                      00069933 T    !OMIT!
!OMIT!                    IDENTIFIER.                                                          00069934 T    !OMIT!
!OMIT!        059     ARRAYDEC: IMPROPER ARRAY SIZE.                                           00069938 T    !OMIT!
!OMIT!        060     FAULTSTMT: MISSING + IN FAULT STATEMENT.                                 00069940 T    !OMIT!
!OMIT!        061     FAULTDEC: INVALID FAULT TYPE: MUST BE FLAG, EXPOVR, ZERO,                00069950 T    !OMIT!
!OMIT!                    INTOVR, OR INDEX.                                                    00069960 T    !OMIT!
!OMIT!        070     CASESTMT:  MISSING BEGIN.                                                00069970 T    !OMIT!
!OMIT!        071     CASESTMT:  MISSING END.                                                  00069980 T    !OMIT!
!OMIT!        080     PRIMARY : MISSING COMMA .                                                00069990 T    !OMIT!
!OMIT!        090     PARSE: MISSING LEFT BRACKET                                              00069991 T    !OMIT!
!OMIT!        091     PARSE: MISSING COLON                                                     00069992 T    !OMIT!
!OMIT!        092     PARSE: ILLEGAL BIT NUMBER                                                00069993 T    !OMIT!
!OMIT!        093     PARSE: FIELD SIZE MUST BE LITERAL                                        00069994 T    !OMIT!
!OMIT!        094     PARSE: MISSING RIGHT BRACKET                                             00069995 T    !OMIT!
!OMIT!        095     PARSE: ILLEGAL FIELD SIZE                                                00069996 T    !OMIT!
!OMIT!        100     ANYWHERE:  UNDECLARED IDENTIFIER.                                        00070000 T    !OMIT!
!OMIT!        101     CHECKER:  AN ATTEMPT HAS BEEN MADE TO ADDRESS AN                         00071000 T    !OMIT!
!OMIT!                    IDENTIFIER WHICH IS LOCAL TO ONE PROCEDURE AND GLOBAL                00072000 T    !OMIT!
!OMIT!                    TO ANOTHER.  IF THE QUANTITY IS A PROCEDURE NAME OR                  00073000 T    !OMIT!
!OMIT!                    AN OWN VARIABLE THIS RESTRICTION IS RELAXED.                         00074000 T    !OMIT!
!OMIT!        102     AEXP:  CONDITIONAL EXPRESSION IS NOT OF ARITHMETIC TYPEH                 00075000 T    !OMIT!
!OMIT!        103     PRIMARY:  PRIMARY MAY NOT BEGIN WITH A QUANTITY OF THIS                  00076000 T    !OMIT!
!OMIT!                    TYPE.                                                                00077000 T    !OMIT!
!OMIT!        104     ANYWHERE:  MISSING RIGHT PARENTHESIS.                                    00078000 T    !OMIT!
!OMIT!        105     ANYWHERE:  MISSING LEFT PARENTHESIS.                                     00079000 T    !OMIT!
!OMIT!        106     PRIMARY:  PRIMARY MAY NOT START WITH DECLARATOR.                         00080000 T    !OMIT!
!OMIT!        107     BEXP:  THE EXPRESSION IS NOT OF BOOLEAN TYPE.                            00081000 T    !OMIT!
!OMIT!        108     EXPRSS:  A RELATION MAY NOT HAVE CONDITIONAL EXPRESSIONS                 00082000 T    !OMIT!
!OMIT!                    AS THE ARITHMETIC EXPRESSIONS.                                       00083000 T    !OMIT!
!OMIT!        109     BOOSEC,SIMPBOO, AND BOOCOMP:  THE PRIMARY IS NOT BOOLEAN.                00084000 T    !OMIT!
!OMIT!        110     BOOCOMP: A NON-BOOLEAN OPERATOR OCCURS IN A BOOLEAN                      00085000 T    !OMIT!
!OMIT!                    EXPRESSION.                                                          00086000 T    !OMIT!
!OMIT!        111     BOOPRIM:  NO EXPRESSION (ARITHMETIC, BOOLEAN, OR DESIGNA-                00087000 T    !OMIT!
!OMIT!                    TIONAL) MAY BEGIN WITH A QUANTITY OF THIS TYPE.                      00088000 T    !OMIT!
!OMIT!        112     BOOPRIM:  NO EXPRESSION (ARITHMETIC, BOOLEAN, OR DESIGNA-                00089000 T    !OMIT!
!OMIT!                    TIONAL) MAY BEGIN WITH A DECLARATOR.                                 00090000 T    !OMIT!
!OMIT!        113     PARSE:  EITHER THE SYTAX OR THE RANGE OF THE LITERALS FOR                00091000 T    !OMIT!
!OMIT!                    A CONCATENATE OPERATOR IS INCORRECT.                                 00092000 T    !OMIT!
!OMIT!        114     DOTSYNTAX:  EITHER THE SYNTAX OR THE RANGE OF THE LITERALS               00093000 T    !OMIT!
!OMIT!                    FOR A PARTIAL WORD DESIGNATOR IS INCORRECT.                          00094000 T    !OMIT!
!OMIT!        115     DEXP:  THE EXPRESSION IS NOT OF DESIGNATIONAL TYPE.                      00095000 T    !OMIT!
!OMIT!        116     IFCLAUSE:  MISSING THEN.                                                 00096000 T    !OMIT!
!OMIT!        117     BANA:  MISSING LEFT BRAKET.                                              00097000 T    !OMIT!
!OMIT!        118     BANA:  MISSING RIGHT BRAKET.                                             00098000 T    !OMIT!
!OMIT!        119     COMPOUNDTAIL:  MISSING SEMICOLON OR END.                                 00099000 T    !OMIT!
!OMIT!        120     COMPOUNDTAIL:  MISSING END.                                              00100000 T    !OMIT!
!OMIT!        121     ACTUALPARAPART:  AN INDEXED FILE MAY BE PASSED BY NAME                   00101000 T    !OMIT!
!OMIT!                    ONLY AND ONLY TO A STREAM PROCEDURE - THE STREAM                     00102000 T    !OMIT!
!OMIT!                    PROCEDURE MAY NOT DO A RELEASE ON THIS TYPE PARA-                    00103000 T    !OMIT!
!OMIT!                    METER.                                                               00104000 T    !OMIT!
!OMIT!        122     ACTUALPARAPART:  STREAM PROCEDURE MAY NOT HAVE AN                        00105000 T    !OMIT!
```

```
!OMIT!                    EXPRESSION PASSED TO IT BY NAME.                                00106000 T    !OMIT!
!OMIT!          123       ACTUALPARAPART:  THE ACTUAL AND FORMAL PARAMETERS DO NOT        00107000 T    !OMIT!
!OMIT!                    AGREE AS TO TYPE.                                               00108000 T    !OMIT!
!OMIT!          124       ACTUALPARAPART:  ACTUAL AND FORMAL ARRAYS DO NOT HAVE SAME      00109000 T    !OMIT!
!OMIT!                    NUMBER OF DIMENSIONS.                                           00110000 T    !OMIT!
!OMIT!          125       ACTUALPARAPART:  STREAM PROCEDURES MAY NOT BE PASSED AS A       00111000 T    !OMIT!
!OMIT!                    PARAMETER TO A PROCEDURE.                                       00112000 T    !OMIT!
!OMIT!          126       ACTUALPARAPART:  NO ACTUAL PARAMETER MAY BEGIN WITH A           00113000 T    !OMIT!
!OMIT!                    QUANTITY OF THIS TYPE.                                          00114000 T    !OMIT!
!OMIT!          127       ACTUALPARAPART:  THIS TYPE QUANTITY MAY NOT BE PASSED TO A      00115000 T    !OMIT!
!OMIT!                    STREAM PROCEDURE.                                               00116000 T    !OMIT!
!OMIT!          128       ACTUALPARAPART:  EITHER ACTUAL AND FORMAL PARAMETERS DO         00117000 T    !OMIT!
!OMIT!                    NOT AGREE AS TO NUMBER, OR EXTRA RIGHT PARENTHESIS.             00118000 T    !OMIT!
!OMIT!          129       ACTUALPARAPART:  ILLEGAL PARAMETER DELIMITER.                   00119000 T    !OMIT!
!OMIT!          130       RELSESTMT:  NO FILE NAME.                                       00120000 T    !OMIT!
!OMIT!          131       DOSTMT:  MISSING UNTIL.                                         00121000 T    !OMIT!
!OMIT!          132       WHILESTMT:  MISSING DO.                                         00122000 T    !OMIT!
!OMIT!          133       LABELR:  MISSING C OLON.                                        00123000 T    !OMIT!
!OMIT!          134       LABELR:  THE LABEL WAS NOT DECLARED IN THIS BLOCK.              00124000 T    !OMIT!
!OMIT!          135       LABELR:  THE LABEL HAS ALREADY OCCURED.                         00125000 T    !OMIT!
!OMIT!          136       FORMATPHRASE:  IMPROPER FORMAT EDITING PHRASE.                  00126000 T    !OMIT!
!OMIT!          137       FORMATPHRASE:  A FORMAT EDITING PHRASE DOES NOT HAVE AN         00127000 T    !OMIT!
!OMIT!                    INTEGER WHERE AN INTEGER IS REQUIRED.                           00128000 T    !OMIT!
!OMIT!          138       FORMATPHRASE:  THE WIDTH IS TOO SMALL IN E OR F EDITING         00129000 T    !OMIT!
!OMIT!                    PHRASE.                                                         00130000 T    !OMIT!
!OMIT!          139       TABLE:  DEFINE IS NESTED MORE THAN EIGHT DEEP.                  00131000 T    !OMIT!
!OMIT!          140       NEXTENT:  AN INTEGER IN A FORMAT IS GREATER THAN 1023.          00132000 T    !OMIT!
!OMIT!          141       SCANNER:  INTEGER OR IDENTIFIER HAS MORE THAN 63                00133000 T    !OMIT!
!OMIT!                    CHARACTORS.                                                     00134000 T    !OMIT!
!OMIT!          142       DEFINEGEN:  A DEFINE CONTAINS MORE THAN 2047 CHARACTORS         00135000 T    !OMIT!
!OMIT!                    (BLANK SUPPRESSED).                                             00136000 T    !OMIT!
!OMIT!          143       COMPOUNDTAIL:  EXTRA END.                                       00137000 T    !OMIT!
!OMIT!          144       STMT:  NO STATEMENT MAY START WITH THIS TYPE IDENTIFIER.        00138000 T    !OMIT!
!OMIT!          145       STMT:  NO STATEMENT MAY START WITH THIS TYPE QUANTITY.          00139000 T    !OMIT!
!OMIT!          146       STMT:  NO STATEMENT MAY START WITH A DECLARATOR - MAY BE        00140000 T    !OMIT!
!OMIT!                    A MISSING END OF A PROCEDURE OR A MISPLACED                     00141000 T    !OMIT!
!OMIT!                    DECLARATION.                                                    00142000 T    !OMIT!
!OMIT!          147       SWITCHGEN:  MORE THAN 256 EXPRESSIONS IN A SWITCH               00143000 T    !OMIT!
!OMIT!                    DECLARATION.                                                    00144000 T    !OMIT!
!OMIT!          148       GETSPACE:  MORE THAN 1023 PROGRAM REFERENCE TABLE CELLS         00145000 T    !OMIT!
!OMIT!                    ARE REQUIRED FOR THIS PROGRAM.                                  00146000 T    !OMIT!
!OMIT!          149       GETSPACE:  MORE THAN 255 STACK CELLS ARE REQUIRED FOR THIS      00147000 T    !OMIT!
!OMIT!                    PROCEDURE.                                                      00148000 T    !OMIT!
!OMIT!          150       ACTUALPARAPART:  CONSTANTS MAY NOT BE PASSED BY NAME TO         00149000 T    !OMIT!
!OMIT!                    STREAM PROCEDURES.                                              00150000 T    !OMIT!
!OMIT!          151       FORSTMT:  INDEX VARIABLE MAY NOT BE BOOLEAN                     00151000 T    !OMIT!
!OMIT!          152       FORSTMT:  MISSING LEFT ARROW FOLLOWING INDEX VARIABLE.          00152000 T    !OMIT!
!OMIT!          153       FORSTMT:  MISSING UNTIL OR WHILE IN STEP ELEMENT.               00153000 T    !OMIT!
!OMIT!          154       FORSTMT:  MISSING DO IN FOR CLAUSE.                             00154000 T    !OMIT!
!OMIT!          155       IFEXP:  MISSING ELSE                                            00155000 T    !OMIT!
!OMIT!          156       LISTELEMENT:  A DESIGNATIONAL EXPRESSION MAY NOT BE A LIST      00156000 T    !OMIT!
!OMIT!                    ELEMENT.                                                        00157000 T    !OMIT!
!OMIT!          157       LISTELEMENT:  A ROW DESIGNATOR MAY NOT BE A LISTELEMENT         00158000 T    !OMIT!
!OMIT!          158       LISTELEMENT:  MISSING RIGHT BRAKET IN GROUP OF ELEMENTS         00159000 T    !OMIT!
!OMIT!          159       PROCSTMT:  ILLEGAL USE OF PROCEDURE OR FUNCTION IDENTIFIER      00160000 T    !OMIT!
!OMIT!          160       PURGE:  DECLARED LABEL DOES NOT OCCUR.                          00161000 T    !OMIT!
!OMIT!          161       PURGE:  DECLARED FORWARD PROCEDURE DOES NOT OCCUR.              00162000 T    !OMIT!
!OMIT!          162       PURGE:  DECLARED SWITCH FORWARD DOES NOT OCCUR.                 00162500 T    !OMIT!
```

```
!OMIT!          163     FORMATPHRASE: THE WIDTH OF A FIELD IS MORE THAN 63.                    00163000 T    !OMIT!
!OMIT!          164     UNKNOWNSTMT: MISSING COMMA IN ZIP OR WAIT STATEMENT.                   00164000 T    !OMIT!
!OMIT!          165     IMPFUN:      MISSING COMMA IN DELAY PARAMETER LIST                     00164100 T    !OMIT!
!OMIT!          172     DEFINEDEC: TOO MANY PARAMETERS IN PARAMETRIC DEFINE                    00164720 T    !OMIT!
!OMIT!                  DECLARATION.                                                           00164725 T    !OMIT!
!OMIT!          173     DEFINEDEC: RIGHT PARENTHESIS OR RIGHT BRACKET EXPECTED                 00164730 T    !OMIT!
!OMIT!                  AFTER PARAMETERS IN PARAMETRIC DEFINE DECLARATION.                     00164735 T    !OMIT!
!OMIT!          174     FIXDEFINEINFO: INCORRECT NUMBER OF PARAMETERS IN                       00164740 T    !OMIT!
!OMIT!                  PARAMETRIC DEFINE INVOCATION.                                          00164745 T    !OMIT!
!OMIT!          175     FIXDEFINEINFO: LEFT BRACKET OR LEFT PARENTHESIS EXPECTED.              00164750 T    !OMIT!
!OMIT!          185     IMPFUN:    LAST PARAMETER MUST BE A SIMPLE OR SUBSCRIPTED              00164850 T    !OMIT!
!OMIT!                             VARIABLE, OR A TYPED PROCEDURE IDENTIFIER.                  00164851 T    !OMIT!
!OMIT!          199     E:   INFO ARRAY HAS OVERFLOWED.                                        00164900 T    !OMIT!
!OMIT!          200     EMIT:   SEGMENT TOO LARGE ( > 4093SYLLABLES).                          00165000 T    !OMIT!
!OMIT!          201     SIMPLE VARIABLE: PARTIAL WORD DESIGNATOR NOT  LEFT-MOST                00166000 T    !OMIT!
!OMIT!                             IN A LEFT PART LIST.                                        00167000 T    !OMIT!
!OMIT!          202     SIMPLE VARIABLE: MISSING . OR * .                                      00168000 T    !OMIT!
!OMIT!          203     SUBSCRIPTED VARIABLE: WRONG NUMBER OF SUBSCRIPTS IN A ROW              00169000 T    !OMIT!
!OMIT!                             DESIGNATOR.                                                 00170000 T    !OMIT!
!OMIT!          204     SUBSCRIPTED VARIABLE: MISSING ] IN A ROW DESIGNATOR.                   00171000 T    !OMIT!
!OMIT!          205     SUBSCRIPTED VARIABLE: A ROW DESIGNATOR APPEARS OUTSIDE OF              00172000 T    !OMIT!
!OMIT!                             AN ACTUAL PARAMETER LIST OR FILL STATEMENT.                 00173000 T    !OMIT!
!OMIT!          206     SUBSCRIPTED VARIABLE: MISSING ].                                       00174000 T    !OMIT!
!OMIT!          207     SUBSCRIPTED VARIABLE: MISSING [.                                       00175000 T    !OMIT!
!OMIT!          208     SUBSCRIPTED VARIABLE: WRONG NUMBER OF SUBSCRIPTS.                      00176000 T    !OMIT!
!OMIT!          209     SUBSCRIPTED VARIABLE: PARTIAL WORD DESIGNATOR NOT LEFT-                00177000 T    !OMIT!
!OMIT!                             MOST IN A LEFT PART LIST.                                   00178000 T    !OMIT!
!OMIT!          210     SUBSCRIPTED VARIABLE: MISSING . OR * .                                 00179000 T    !OMIT!
!OMIT!          211     VARIABLE: PROCEDURE ID USED OUTSIDE OF SCOPE IN LEFT PART.             00180000 T    !OMIT!
!OMIT!          212     VARIABLE: SUB-ARRAY DESIGNATOR PERMITTED AS ACTUAL                     00180100 T    !OMIT!
!OMIT!                             PARAMETER ONLY.                                             00180200 T    !OMIT!
!OMIT!          250     STREAM STMT: ILLEGAL STREAM STATEMENT.                                 00181000 T    !OMIT!
!OMIT!          251     ANY STREAM STMT PROCEDURE: MISSING *.                                  00182000 T    !OMIT!
!OMIT!          252     INDEX: MISSING + OR - .                                                00183000 T    !OMIT!
!OMIT!          253     INDEX: MISSING NUMBER OR STREAM VARIABLE.                              00184000 T    !OMIT!
!OMIT!          254     SCANNER: STRING, OCTAL, OR HEX CONSTANT HAS FLAG BIT SET.              00185000 P    !OMIT!
!OMIT!          255     DSS: MISSING STRING IN DS* LIT STATEMENT.                              00186000 T    !OMIT!
!OMIT!          256     RELEASES: MISSING PARENTHESIS OR FILE IDENTIFIER IS NOT                00187000 T    !OMIT!
!OMIT!                             A FORMAL PARAMETER.                                         00188000 T    !OMIT!
!OMIT!          257     GOTOS,LABELS,OR JUMPS: LABEL SPECIFIED IS NOT ON THE SAME              00189000 T    !OMIT!
!OMIT!                             NEST LEVEL AS A PRECEDING APPEARANCE OF THE                 00190000 T    !OMIT!
!OMIT!                             LABEL.                                                      00191000 T    !OMIT!
!OMIT!          258     LABELS: MISSING ;.                                                     00192000 T    !OMIT!
!OMIT!          259     LABELS: LABEL APPEARS MORE THAN ONCE.                                  00193000 T    !OMIT!
!OMIT!          260     GOTOS: MISSING LABEL IN A GO TO OR JUMP OUT TO STATEMENT.              00194000 T    !OMIT!
!OMIT!          261     JUMPS: MISSING OUT IN JUMP OUT STATEMENT.                              00195000 T    !OMIT!
!OMIT!          262     NESTS: MISSING PARENTHESIS.                                            00196000 T    !OMIT!
!OMIT!          263     IFS:MISSING SC IN IF STATEMENT.                                        00197000 T    !OMIT!
!OMIT!          264     IFS: MISSING RELATIONAL IN IF STATEMENT.                               00198000 T    !OMIT!
!OMIT!          265     IFS: MISSING ALPHA,DC OR STRING IN IF STATEMENT.                       00199000 T    !OMIT!
!OMIT!          266     IFS: MISSING THEN IN IF STATEMENT.                                     00200000 T    !OMIT!
!OMIT!          267     PREDFIX: THERE ARE GO TO STATEMENTS IN WHICH THE LABEL IS              00201000 T    !OMIT!
!OMIT!                             UNDEFINED.                                                  00202000 T    !OMIT!
!OMIT!          268     EMITC:  A REPEAT INDEX ≥ 64 WAS SPECIFIED OR TOO MANY                  00203000 T    !OMIT!
!OMIT!                             FORMAL PARAMETERS,LOCALS AND LABELS.                        00204000 T    !OMIT!
!OMIT!          269     TABLE: A CONSTANT IS SPECIFIED WHICH IS TOO LARGE                      00205000 T    !OMIT!
!OMIT!                             OR TOO SMALL.                                               00206000 T    !OMIT!
```

```
!OMIT!          270         IFS! RELATIONAL IN SC<RELATIONAL>ALPHA MUST BE "EQUAL".                  00206100 T    !OMIT!
!OMIT!          271         IFS! IMPROPER CONSTRUCT FOR <SOURCE WITH LITERAL>.                       00206200 T    !OMIT!
!OMIT!          281         DBLSTMT!  MISSING (.                                                   00207000 T    !OMIT!
!OMIT!          282         DBLSTMT!  TOO MANY OPERATORS.                                          00208000 T    !OMIT!
!OMIT!          283         DBLSTMT!  TOO MANY OPERANDS.                                           00209000 T    !OMIT!
!OMIT!          284         DBLSTMT!  MISSING , .                                                  00210000 T    !OMIT!
!OMIT!          285         DBLSTMT!  TOO FEW OPERANDS.                                            00211000 T    !OMIT!
!OMIT!          286         DBLSTMT!  ILLEGAL PARAMETER .                                          00211100 T    !OMIT!
!OMIT!          290         FILEATTRIBUTEHANDLER!  MISSING . IN FILE ATTRIBUTE PART                00211510 T    !OMIT!
!OMIT!          291         FILEATTRIBUTEHANDLER!  MISSING OR UNDEFINED FILE ATTRIBUTE             00211520 T    !OMIT!
!OMIT!          292         FILEATTRIBUTEHANDLER!  MISSING ← IN FILE ATTR ASSIGN STMT              00211530 T    !OMIT!
!OMIT!          293         FILEATTRIBUTEHANDLER!  FILE ATTRIBUTE IS NON ASSIGNABLE                00211540 T    !OMIT!
!OMIT!          294         PRIMARY!  FILE ATTRIBUTE IS NOT TYPE REAL                              00211550 T    !OMIT!
!OMIT!          295         FILEATTRIBUTEHANDLER!  FILE ATTRIBUTE MUST BE LEFT MOST                00211551 T    !OMIT!
!OMIT!                                 IN A LEFT PART LIST.                                       00211552 T    !OMIT!
!OMIT!          300         FILLSTMT!  THE IDENTIFIER FOLLOWING "FILL" IS NOT                      00212000 T    !OMIT!
!OMIT!                      AN ARRAY IDENTIFIER.                                                   00213000 T    !OMIT!
!OMIT!          301         FILLSTMT!  MISSING "WITH" IN FILL STATEMENT.                           00214000 T    !OMIT!
!OMIT!          302         FILLSTMT!  IMPROPER FILL ELEMENT.                                      00215000 T    !OMIT!
!OMIT!          303         FILLSTMT!  NON-OCTAL CHARACTER IN OCTAL FILL.                          00216000 T    !OMIT!
!OMIT!          304         FILLSTMT!  IMPROPER ARRAY ROW DESIGNATOR IN FILL.                      00217000 T    !OMIT!
!OMIT!          305         FILLSTMT!  DATA IN FILL EXCEEDS 1023 WORDS.                            00218000 T    !OMIT!
!OMIT!          304     FILLSTMT! IMPROPER ROW DESIGNATOR.                                         00218100 T    !OMIT!
!OMIT!          306         FILLSTMT!  ODD NUMBER OF PARENTHESES IN FILL.                          00218110 T    !OMIT!
!OMIT!          307         WHIPOUT!  FORMAT > 1023 WORDS.                                         00218112 T    !OMIT!
!OMIT!          350         CHECKCOMMA!   MISSING OR ILLEGAL PARAMETER DELIMITER IN               00218200 T    !OMIT!
!OMIT!                                    SORT OR MERGE STATEMENT.                                00218210 T    !OMIT!
!OMIT!          351         OUTPROCHECK!  ILLEGAL TYPE FOR SORT OR MERGE OUTPUT PROC.              00218220 T    !OMIT!
!OMIT!          352         OUTPROCHECK!  OUTPUT PROCEDURE IN SORT OR MERGE STMT DOES              00218230 T    !OMIT!
!OMIT!                                    NOT HAVE EXACTLY TWO PARAMETERS.                        00218240 T    !OMIT!
!OMIT!          353         OUTPROCHECK!  FIRST PARAMETER OF OUTPUT PROCEDURE MUST                 00218250 T    !OMIT!
!OMIT!                                    BE BOOLEAN.                                             00218260 T    !OMIT!
!OMIT!          354         OUTPROCHECK!  SECOND PARAM OF OUTPUT PROCEDURE MUST BE                 00218270 T    !OMIT!
!OMIT!                                    ONE-DIM ARRAY.                                          00218280 T    !OMIT!
!OMIT!          355         SORTSTMT!     MISSING (.                                               00218290 T    !OMIT!
!OMIT!          356         HVCHECK!      ILLEGAL TYPE FOR SORT OR MERGE HIGHVALUE PRO             00218300 T    !OMIT!
!OMIT!          357         HVCHECK!      HIVALUE PROCEDURE DOES NOT HAVE EXACTLY ONE              00218310 T    !OMIT!
!OMIT!                                    PARAMETER.                                              00218320 T    !OMIT!
!OMIT!          358         HVCHECK!      HIVALUE PROCEDURE PARAM NOT ONE-DIM ARRAY.               00218330 T    !OMIT!
!OMIT!          359         EQLESCHECK!   SORT OR MERGE COMPARE PROCEDURE NOT BOOLEAN.            00218340 T    !OMIT!
!OMIT!          360         EQLESCHECK!   COMPARE PROCEDURE DOES NOT HAVE EXACTLY                  00218350 T    !OMIT!
!OMIT!                                    TWO PARAMETERS.                                         00218360 T    !OMIT!
!OMIT!          361         EQLESCHECK!   COMPARE PROCEDURE FIRST PARAM NOT 1-D ARRAY.            00218370 T    !OMIT!
!OMIT!          362         EQLESCHECK!   COMPARE PROCEDURE SECOND PARAM NOT 1-D ARRAY            00218380 T    !OMIT!
!OMIT!          363         INPROCHECK!   SORT STMT INPUT PROCEDURE NOT BOOLEAN.                  00218390 T    !OMIT!
!OMIT!          364         INPROCHECK!   INPUT PROCEDURE DOES NOT HAVE EXACTLY ONE               00218400 T    !OMIT!
!OMIT!                                    PARAMETER.                                              00218410 T    !OMIT!
!OMIT!          365         INPROCHECK!   INPUT PROCEDURE PARAMETER NOT ONE-D ARRAY.              00218420 T    !OMIT!
!OMIT!          366         SORTSTMT!     MISSING ).                                               00218430 T    !OMIT!
!OMIT!          367         MERGESTMT!    MISSING (.                                               00218440 T    !OMIT!
!OMIT!          368         MERGFSTMT!    MORE THAN 7 OR LESS THAN 2 FILES TO MERGE.               00218450 T    !OMIT!
!OMIT!          369         MERGFSTMT!    MISSING ).                                               00218460 T    !OMIT!
!OMIT!          381         CMPLXSTMT!     MISSING (.                                              00218500 T    !OMIT!
!OMIT!          382         CMPLXSTMT!     TOO MANY OPERATORS.                                     00218505 T    !OMIT!
!OMIT!          383         CMPLXSTMT!     TOO MANY OPERANDS.                                      00218510 T    !OMIT!
!OMIT!          384         CMPLXSTMT!     MISSING , .                                             00218515 T    !OMIT!
!OMIT!          385         CMPLXSTMT!     TOO FEW OPERANDS.                                       00218520 T    !OMIT!
!OMIT!          386         CMPLXSTMT!     ILLEGAL PARAMETER.                                      00218525 T    !OMIT!
```

```
!OMIT!      400     MERRIMAC:MISSING FILE ID IN MONITOR DEC.                                    00219000 T    !OMIT!
!OMIT!      401     MERRIMAC:MISSING LEFT PARENTHESIS IN MONITOR DEC.                           00220000 T    !OMIT!
!OMIT!      402     MERRIMAC:IMPROPER SUBSCRIPT FOR MONITOR LIST ELEMENT.                       00221000 T    !OMIT!
!OMIT!      403     MERRIMAC:IMPROPER SUBSCRIPT EXPRESSION DELIMITER IN                         00222000 T    !OMIT!
!OMIT!                  MONITOR LIST ELEMENT.                                                   00223000 T    !OMIT!
!OMIT!      404     MERRIMAC:IMPROPER NUMBER OF SUBSCRIPTS IN MONITOR LIST                      00224000 T    !OMIT!
!OMIT!                  ELEMENT.                                                                00225000 T    !OMIT!
!OMIT!      405     MERRIMAC:LABEL OR SWITCH MONITORED AT IMPROPER LAVEL.                       00226000 T    !OMIT!
!OMIT!      406     MERRIMAC:IMPROPER MONITOR LIST ELEMENT.                                     00227000 T    !OMIT!
!OMIT!      407     MERRIMAC:MISSING RIGHT PARENTHESIS IN MONITOR DECLARATION.                  00228000 T    !OMIT!
!OMIT!      408     MERRIMAC:IMPROPER MONITOR DECLARATION DELIMITER.                            00229000 T    !OMIT!
!OMIT!      409     DMUP:MISSING FILE IDENTIFIER IN DUMP DECLARATION.                           00230000 T    !OMIT!
!OMIT!      410     DMUP:MISSING LEFT PARENTHESIS IN DUMP DECLARATION.                          00231000 T    !OMIT!
!OMIT!      411     DMUP:SUBSCRIPTED VARIABLE IN DUMP LIST HAS WRONG NUMBER OF                  00232000 T    !OMIT!
!OMIT!                  SUBSCRIPTS.                                                             00233000 T    !OMIT!
!OMIT!      412     DMUP:SUBSCRIPTED VARIABLE IN DUMP LIST HAS WRONG NUMBER OF                  00234000 T    !OMIT!
!OMIT!                  SUBSCRIPTS.                                                             00235000 T    !OMIT!
!OMIT!      413     DMUP:IMPROPER ARRAY DUMP LIST ELEMENT.                                      00236000 T    !OMIT!
!OMIT!      414     DMUP:ILLEGAL DUMP LIST ELEMENT.                                             00237000 T    !OMIT!
!OMIT!      415     DMUP:MORE THAN 100 LABELS APPEAR AS DUMP LIST ELEMENTS                      00238000 T    !OMIT!
!OMIT!                  IN ONE DUMP DECLARATION.                                                00239000 T    !OMIT!
!OMIT!      416     DMUP:ILLEGAL DUMP LIST ELEMENT DELIMITER.                                   00240000 T    !OMIT!
!OMIT!      417     DMUP:MISSING OR NON-LOCAL LABEL IN DUMP DECLARATION.                        00241000 T    !OMIT!
!OMIT!      418     DMUP:MISSING COLON IN DUMP DECLARATION.                                     00242000 T    !OMIT!
!OMIT!      419     DMUP:IMPROPER DUMP DECLARATION DELIMITER.                                   00243000 T    !OMIT!
!OMIT!      420     READSTMT:MISSING LEFT PARENTHESIS IN READ STATEMENT.                        00244000 T    !OMIT!
!OMIT!      421     READSTMT:MISSING LEFT PARENTHESIS IN READ REVERSE                           00245000 T    !OMIT!
!OMIT!                  STATEMENT.                                                              00246000 T    !OMIT!
!OMIT!      422     READSTMT:MISSING FILE IN READ STATEMENT.                                    00247000 T    !OMIT!
!OMIT!                                                                                          00248000 T    !OMIT!
!OMIT!      424     READSTMT:IMPROPER FILE DELIMITER IN READ STATEMENT                          00249000 T    !OMIT!
!OMIT!      425     READSTMT:IMPROPER FORMAT DELIMITER IN READ STATEMENT.                       00250000 T    !OMIT!
!OMIT!      426     READSTMT:IMPROPER DELIMITER FOR SECOND PARAMETER IN READ                    00251000 T    !OMIT!
!OMIT!                  STATEMENT.                                                              00252000 T    !OMIT!
!OMIT!      427     READSTMT:IMPROPER ROW DESIGNATOR IN READ STATEMENT.                         00253000 T    !OMIT!
!OMIT!      428     READSTMT:IMPROPER ROW DESIGNATOR DELIMITER IN READ                          00254000 T    !OMIT!
!OMIT!                  STATEMENT.                                                              00255000 T    !OMIT!
!OMIT!      429     READSTMT:MISSING ROW DESIGNATOR IN READ STATEMENT.                          00256000 T    !OMIT!
!OMIT!      430     READSTMT:IMPROPER DELIMITER PRECEEDING THE LIST IN A READ                   00257000 T    !OMIT!
!OMIT!                  STATEMENT.                                                              00258000 T    !OMIT!
!OMIT!                                                                                          00259000 T    !OMIT!
!OMIT!                                                                                          00260000 T    !OMIT!
!OMIT!                                                                                          00261000 T    !OMIT!
!OMIT!                                                                                          00262000 T    !OMIT!
!OMIT!      433     HANDLETHETAILENDOFAREADORSPACESTATEMENT:MISSING RIGHT                        00263000 T    !OMIT!
!OMIT!                  BRACKET IN READ OR SPACE STATEMENT.                                     00264000 T    !OMIT!
!OMIT!      434     SPACESTMT:MISSING LEFT PARENTHESIS IN SPACE STATEMENT.                      00265000 T    !OMIT!
!OMIT!      435     SPACESTMT:IMPROPER FILE IDENTIFIER IN SPACE STATEMENT.                      00266000 T    !OMIT!
!OMIT!      436     SPACESTMT:MISSING COMMA IN SPACE STATEMENT.                                 00267000 T    !OMIT!
!OMIT!      437     SPACESTMT:MISSING RIGHT PARENTHESIS IN SPACE STATEMENT.                     00268000 T    !OMIT!
!OMIT!      438     WRITESTMT:MISSING LEFT PARENTHESIS IN A WRITE STATEMENT.                    00269000 T    !OMIT!
!OMIT!      439     WRITESTMT:IMPROPER FILE IDENTIFIER IN A WRITE STATEMENT.                    00270000 T    !OMIT!
!OMIT!      440     WRITESTMT:IMPROPER DELIMITER FOR FIRST PARAMETER IN A                       00271000 T    !OMIT!
!OMIT!                  WRITE STATEMENT.                                                        00272000 T    !OMIT!
!OMIT!      441     WRITESTMT:MISSING RIGHT BRACKET IN CARRIAGE CONTROL PART                    00273000 T    !OMIT!
!OMIT!                  OF A WRITE STATEMENT.                                                   00274000 T    !OMIT!
!OMIT!      442     WRITESTMT:ILLEGAL CARRIAGE CONTROL DELIMITER IN A WRITE                     00275000 T    !OMIT!
```

```
!OMIT!                    STATEMENT.                                                                  00276000 T    !OMIT!
!OMIT!          443       WRITESTMT:IMPROPER SECOND PARAMETER DELIMITER IN WRITE                      00277000 T    !OMIT!
!OMIT!                    STATEMENT.                                                                  00278000 T    !OMIT!
!OMIT!          444       WRITESTMT:IMPROPER ROW DESIGNATOR IN A WRITE STATEMENT.                     00279000 T    !OMIT!
!OMIT!          445       WRITESTMT:MISSING RIGHT PARENTHESIS AFTER A ROW DESIGNATOR                  00280000 T    !OMIT!
!OMIT!                    IN A WRITE STATEMENT.                                                       00281000 T    !OMIT!
!OMIT!          446       WRITESTMT:MISSING ROW DESIGNATOR IN A WRITE STATEMENT.                      00282000 T    !OMIT!
!OMIT!          447       WRITESTMT:IMPROPER DELIMITER PRECEEDING A LIST IN A WRITE                   00283000 T    !OMIT!
!OMIT!                    STATEMENT.                                                                  00284000 T    !OMIT!
!OMIT!          448       WRITESTMT:IMPROPER LIST DELIMITER IN A WRITE STATEMENT.                     00285000 T    !OMIT!
!OMIT!          449       READSTMT:IMPROPER LIST DELIMITER IN A READ STATEMENT.                       00286000 T    !OMIT!
!OMIT!          450       LOCKSTMT:MISSING LEFT PARENTHESIS IN A LOCK STATEMENT.                      00287000 T    !OMIT!
!OMIT!          451       LOCKSTMT:IMPROPER FILE PART IN A LOCK STATEMENT.                            00288000 T    !OMIT!
!OMIT!          452       LOCKSTMT:MISSING COMMA IN A LOCK STATEMENT.                                 00289000 T    !OMIT!
!OMIT!          453       LOCKSTMT:IMPROPER UNIT DISPOSITION PART IN A LOCK                           00290000 T    !OMIT!
!OMIT!                    STATEMENT.                                                                  00291000 T    !OMIT!
!OMIT!          454       LOCKSTMT:MISSING RIGHT PARENTHESIS IN A LOCK STATEMENT.                     00292000 T    !OMIT!
!OMIT!          455       CLOSESTMT:MISSING LEFT PARENTHESIS IN A CLOSE STATEMENT.                    00293000 T    !OMIT!
!OMIT!          456       CLOSESTMT:IMPROPER FILE PART IN A CLOSE STATEMENT.                          00294000 T    !OMIT!
!OMIT!          457       CLOSESTMT:MISSING COMMA IN A CLOSE STATEMENT.                               00295000 T    !OMIT!
!OMIT!          458       CLOSESTMT:IMPROPER UNIT DISPOSITION PART IN A CLOSE                         00296000 T    !OMIT!
!OMIT!                    STATEMENT.                                                                  00297000 T    !OMIT!
!OMIT!          459       CLOSESTMT:MISSING RIGHT PARENTHESIS IN A CLOSE STATEMENT.                   00298000 T    !OMIT!
!OMIT!          460       RWNDSTMT:MISSING LEFT PARENTHESIS IN A REWIND STATEMENT.                    00299000 T    !OMIT!
!OMIT!          461       RWNDSTMT:IMPROPER FILE PART IN A REWIND STATEMENT.                          00300000 T    !OMIT!
!OMIT!          462       RWNDSTMT:MISSING RIGHT PARENTHESIS IN A REWIND STATEMENT.                   00301000 T    !OMIT!
!OMIT!          463       BLOCK:A MONITOR DECLARATION APPEARS IN THE SPECIFICATION                    00302000 T    !OMIT!
!OMIT!                    PART OF A PROCEDURE.                                                        00303000 T    !OMIT!
!OMIT!          464       BLOCK:A DUMP DECLARATION APPEARS IN THE SPECIFICATION PART                  00304000 T    !OMIT!
!OMIT!                    OF A PROCEDURE.                                                             00305000 T    !OMIT!
!OMIT!          465       DMUP:DUMP INDICATOR MUST BE UNSIGNED INTEGER OR                             00305003 T    !OMIT!
!OMIT!                    SIMPLE VARIABLE                                                             00305004 T    !OMIT!
!OMIT!          500       SEARCHLIB: ILLEGAL LIBRARY IDENTIFIER.                                      00305010 T    !OMIT!
!OMIT!          501       SEARCHLIB: LIBRARY IDENTIFIER NOT CONTAINED IN DIRECTORY.                   00305020 T    !OMIT!
!OMIT!          502       SEARCHLIB: ILLEGAL LIBRARY START POINT.                                     00305030 T    !OMIT!
!OMIT!          503       SEARCHLIB: SEPARATOR REQUIRED BETWEEN START POINT AND LENGTH.               00305040 T    !OMIT!
!OMIT!          504       SEARCHLIB: ILLEGAL LIBRARY LENGTH.                                          00305050 T    !OMIT!
!OMIT!          505       SEARCHLIB: MISSING BRACKET.                                                 00305060 T    !OMIT!
!OMIT!                                                                                                00305070 T    !OMIT!
!OMIT!          507       SEARCHLIB: TAPE POSITIONING ERROR.                                          00305080 T    !OMIT!
!OMIT!          509       IODEC:    NON-LITERAL FILE VALUE NOT GLOBAL TO FILE DECL.                   00305100 T    !OMIT!
!OMIT!          520       TABLE: STRING LONGER THAN ONE WORD (48 BITS).                               00306200 T    !OMIT!
!OMIT!          521       TABLE: STRING CONTAINS A NON-PERMISSIBLE CHARACTER.                         00306300 T    !OMIT!
!OMIT!          600       DOLLARCARD: NUMBER EXPECTED.                                                00400000 T    !OMIT!
!OMIT!          601       DOLLARCARD: OPTION IDENTIFIER EXPECTED.                                     00401000 T    !OMIT!
!OMIT!          602       DOLLARCARD: TOO MANY USER-DEFINED OPTIONS.                                  00403000 T    !OMIT!
!OMIT!          603       DOLLARCARD: UNRECOGNIZED WORD OR CHARACTER.                                 00404000 T    !OMIT!
!OMIT!          604       DOLLARCARD: MISMATCHED PARENTHESES.                                         00405000 T    !OMIT!
!OMIT!          610       READACARD: SEQUENCE ERROR.                                                  00410000 T    !OMIT!
!OMIT!          611       READACARD: ERROR LIMIT HAS BEEN EXCEEDED.                                   00411000 T    !OMIT!
!OMIT!          612       INCLUDECARD:  TOOMANY NESTED INCLUDES.                  %107-               00412000 C    !OMIT!
!OMIT!          613       INCLUDECARD:  MISSING FILE NAME ON INCLUDE CARD.        %107-               00413000 C    !OMIT!
!OMIT!          614       INCLUDECARD:  ENDING SEQUENCE NUMBER MISSING.           %107-               00414000 C    !OMIT!
!OMIT!          615       INCLUDECARD:  COPY MISSING ON INCLUDE CARD.             %107-               00415000 C    !OMIT!
!OMIT!          616       INCLUDECARD:  MORE THAN ONE FILE NAME ON INCLUDE CARD   %107-               00416000 C    !OMIT!
!OMIT!          617       INCLUDECARD:  + COPY CAN NOT BE USED UNLESS $ IS IN COLUMN ONE              00417000 C    !OMIT!
!OMIT!          618       BLOCK: AUXMEM APPEARS IMMEDIATELY BEFORE IDENTIFIER (NO TYPE)               00418000 C    !OMIT!
!OMIT!          ;                                                                                     00490000 T    !OMIT!
```

```
  !OMIT!          $POP OMIT LISTA                                          %121-        00499999 C    !OMIT!
                 BEGIN COMMENT OUTERMOST BLOCK;                                        00500000 T  0001!000010
PRT(22) = *LIST, LABEL, OR SEGMENT DESCRIPTOR*
PRT(23) = *OUTER BLOCK DESCRIPTOR*
PRT(24) = *SEGMENT DESCRIPTOR*

                                                                  START OF SEGMENT **********    2
                  INTEGER ERRORCOUNT; COMMENT NUMBER OF ERROR MSGS. MCP WILL TYPE       00501000 T  0002!000010
PRT(25) = ERRORCOUNT
                                 SYNTX ERR AT EOJ IF THIS IS NON-ZERO. MUST BE @R+25;  00502000 T  0002!000010
                  INTEGER SAVETIME; COMMENT SAVE-FACTOR FOR CODE FILE, GIVEN BY MCP.    00503000 T  0002!000010
PRT(26) = SAVETIME
                                 IF COMPILE & GO =0, FOR SYNTAX, =-1, MUST BE AT R+26; 00504000 T  0002!000010
                INTEGER CARDNUMBER; % SEQ # OF CARD BEING PROCESSED.                   00504100 T  0002!000010
PRT(27) = CARDNUMBER
                INTEGER CARDCOUNT;   % NUMBER OF CARDS PROCESSED.                     00504150 T  0002!000010
PRT(30) = CARDCOUNT
                INTEGER LASTADDRESS;                                                   00504200 T  0002!000010
PRT(31) = LASTADDRESS
                ARRAY   ENIL[0:7,0:127];                                               00504300 T  0002!000010
PRT(32) = ENIL
                INTEGER ENILPTR;                                                       00504400 T  0002!000210
PRT(33) = ENILPTR
                DEFINE  ENILSPOT = ENIL[ENILPTR.[38:3], ENILPTR.[41:7]]#;              00504500 T  0002!000210
                ARRAY   LDICT[0:7,0:127];                                              00504600 T  0002!000210
PRT(34) = LDICT
                BOOLEAN BUILDLINE;                                                     00504700 T  0002!000410
PRT(35) = BUILDLINE
               BOOLEAN REL;                                                            00504801 T  0002!000410
PRT(36) = REL
                  COMMENT RR1-RR11 ARE USED BY SOME PROCEDURES IN LIEU OF LOCALS,      00505000 T  0002!000410
                      TO SAVE SOME STACK SPACE;                                        00506000 T  0002!000410
                REAL RR1,RR2,RR3,RR4,RR5,RR6,RR7,RR8,RR9,RR10,RR11;                    00507000 T  0002!000410
PRT(37) = RR1
PRT(40) = RR2
PRT(41) = RR3
PRT(42) = RR4
PRT(43) = RR5
PRT(44) = RR6
PRT(45) = RR7
PRT(46) = RR8
PRT(47) = RR9
PRT(50) = RR10
PRT(51) = RR11
                  COMMENT SOME OF THE RRI ARE USED TO PASS FILE INFORMATION TO         00508000 T  0002!000410
                      THE MAIN BLOCK;                                                  00509000 T  0002!000410
                  COMMENT EXAMIN RETURNS THE CHARACTER AT ABSOLUTE ADDRESS NCR;        00510000 T  0002!000410
                REAL STREAM PROCEDURE EXAMIN(NCR); VALUE NCR;                          00511000 T  0002!000410
PRT(52) = EXAMIN
                        BEGIN SI←NCR; DI←LOC EXAMIN; DI←DI+7; DS←CHR END;              00512000 T  0002!000410


                REAL STREAM PROCEDURE EXAMINELAST(AC, CT); VALUE CT;                   00512100 T  0002!000712
PRT(53) = EXAMINELAST
                  BEGIN                                                                00512200 T  0002!000712
                      SI ← AC; SI ← SI + CT;                                           00512300 T  0002!000712
```

```
                    DI ← LOC EXAMINELAST; DI ← DI+7;                                    00512400 T  0002:000?:3
                    DS ← 1 CHR;                                                         00512500 T  0002:0008:0
                END EXAMINELAST;                                                        00512600 T  0002:0008:1



                  COMMENT MOVECHARACTERS MOVES N CHARACTERS FROM THE SK-TH CHARACTER     00513000 T  0002:0009:3
                      IN SORCE TO THE DK-TH CHARACTER IN DEST. 0≤N≤63,0≤SK≤127;          00514000 T  0002:0009:3
                        DEFINE DK=DSK#;                                                 00514500 T  0002:0009:3
                STREAM PROCEDURE MOVECHARACTERS(N,SORCE,SK,DEST,DSK);                   00515000 T  0002:0009:3
PRT(54) = MOVECHARACTERS
                                      VALUE N,      SK,     DSK ;                     00516000 T  0002:0009:3
                    BEGIN SI←LOC SK; SI←SI+6;                                           00517000 T  0002:0009:3
                        IF SC≠"0" THEN BEGIN SI←SORCE; 2(SI←SI+32); SORCE←SI END;       00518000 T  0002:0010:1
                        SI←LOC DK; SI←SI+6; DI←DEST;                                    00519000 T  0002:0012:1
                        IF SC≠"0" THEN 2(DI←DI+32);                                     00520000 T  0002:0013:3
                        SI←SORCE; SI←SI+SK; DI←DI+DK; DS←N CHR;                         00521000 T  0002:0015:2
                    END MOVECHARACTERS;                                                 00522000 T  0002:0017:2



                INTEGER STREAM PROCEDURE GETF(Q); VALUE Q;                              00523000 T  0002:0017:2
PRT(55) = GETF
                  BEGIN SI←LOC GETF; SI←SI-7; DI←LOC Q; DI←DI+5;                        00524000 T  0002:0017:2
                      SKIP 3 DB; 9(IF SB THEN DS←SET ELSE DS←RESET; SKIP SB);           00525000 T  0002:0019:2
                      DI←LOC Q; SI←Q; DS←WDS; SI←Q; GETF←SI                             00526000 T  0002:0021:2
                  END GETF;                                                             00527000 T  0002:0022:1



                COMMENT START SETTING UP FILE PARAMETERS;                               00528000 T  0002:0023:3
                        IF EXAMIN(RR11←GETF(3)+"Y08" )≠12 THEN RR1←5 ELSE               00529000 T  0002:0023:3
                            BEGIN RR1←2; RR2←150END;                                    00530000 T  0002:0029:2
                        IF EXAMIN(RR11+5)≠12 THEN RR3←4 ELSE                            00531000 T  0002:0032:1
                            BEGIN RR3←2; RR4←150END;                                    00532000 T  0002:0036:0
                        IF EXAMIN(RR11+10)=12 THEN                                      00533000 T  0002:0038:0
                            BEGIN RR5←2; RR6←10; RR7←150END ELSE                        00534000 T  0002:0040:0
                            BEGIN RR5←1; RR6←56; RR7←10 END;                            00535000 T  0002:0043:2
                        IF EXAMIN(RR11+15)=12 THEN                                      00536000 T  0002:0045:3
                            BEGIN RR8←10; RR9←150END ELSE                               00537000 T  0002:0048:0
                            BEGIN RR8←56; RR9←10 END;                                   00538000 T  0002:0050:0
                        IF EXAMIN(RR11+20)=12 THEN RR10←150;                            00539000 T  0002:0052:0
                BEGIN                                                                   01000000 T  0002:0055:3
                                                                                        01000100 T  0002:0055:3
                                                                                        01000200 T  0002:0055:3
                                                                                        01000300 T  0002:0055:3
                                                                                        01000400 T  0002:0055:3
                                                                                        01000500 T  0002:0055:3
                                                                                        01000600 T  0002:0055:3
                                                                                        01000700 T  0002:0055:3
                INTEGER NUMSEQUENCEERRORS;
PRT(56) = *SEGMENT DESCRIPTOR*                                                  START OF SEGMENT **********     3

PRT(57) = NUMSEQUENCEERRORS
                INTEGER OPINX; % USED FOR INDEXING INTO OPTIONS ARRAY.                  01000800 T  0003:0000:0
PRT(60) = OPINX
```

```
                BOOLEAN SETTING; % USED BY DOLLARCARD FOR AN OPTION'S SETTING.                    01000802 T  0003:0000:0
PRT(61) = SETTING
                BOOLEAN GOGOGO;% TRUE FOR SPECIAL WRITES AND READS                                01000810 T  0003:0000:0
PRT(62) = GOGOGO
            PROCEDURE CHECKBOUNDLVL;FORWARD;                                                      01000830 T  0003:0000:0
PRT(63) = CHECKBOUNDLVL
                BOOLEAN ARRAYFLAG;% USED TO INFORM PRIMARY AND BOOPRIM THAT WE ARE                01000840 T  0003:0000:0
PRT(64) = ARRAYFLAG
                                    % EVALUATING AN ARRAY BOUND                                   01000850 T  0003:0000:0
             INTEGER NEWINX, ADDVALUE, BASENUM, TOTALNO;                                          01000860 T  0003:0000:0
PRT(65) = NEWINX
PRT(66) = ADDVALUE
PRT(67) = BASENUM
PRT(70) = TOTALNO
                COMMENT  ADDVALUE IS INCREMENT VALUE FOR RESEQUENCING                             01000870 T  0003:0000:0
                          BASENUM IS STARTING VALUE                                               01000880 T  0003:0000:0
                          TOTALNO IS BASENUM + ADDVALUE CALCULATED FOR EACH                       01000890 T  0003:0000:0
                          CARD AS TOTALNO = TOTALNO + ADDVALUE;                                   01000900 T  0003:0000:0
            DEFINE OPARSIZE = 200 #;                                                              01000902 T  0003:0000:0
            ARRAY OPTIONS[0:OPARSIZE];                                                            01000904 T  0003:0000:0
PRT(71) = OPTIONS
            BOOLEAN OPTIONWORD;                                                                   01000910 T  0003:0002:1
PRT(72) = OPTIONWORD
            DEFINE   CHECKBIT       =  1#,                                                        01000920 T  0003:0002:1
                     DEBUGBIT       =  2#,                                                        01000930 T  0003:0002:1
                     DECKBIT        =  3#,                                                        01000940 T  0003:0002:1
                     FORMATBIT      =  4#,                                                        01000950 T  0003:0002:1
                     INTBIT         =  5#,                                                        01000960 T  0003:0002:1
                     LISTABIT       =  6#,                                                        01000970 T  0003:0002:1
                     LISTBIT        =  7#,                                                        01000980 T  0003:0002:1
                     LISTPBIT       =  8#,                                                        01000990 T  0003:0002:1
                     MCPBIT         =  9#,                                                        01001000 T  0003:0002:1
                     MERGEBIT       = 10#,                                                        01001010 T  0003:0002:1
                     NESTBIT        = 11#,                                                        01001020 T  0003:0002:1
                     NEWBIT         = 12#,                                                        01001030 T  0003:0002:1
                     NEWINCLBIT     = 13#,                                                        01001040 T  0003:0002:1
                     OMITBIT        = 14#,                                                        01001050 T  0003:0002:1
                     PRINTDOLLARBIT = 15#,                                                        01001060 T  0003:0002:1
                     PRTBIT         = 16#,                                                        01001070 T  0003:0002:1
                     PUNCHBIT       = 17#,                                                        01001080 T  0003:0002:1
                     PURGEBIT       = 18#,                                                        01001090 T  0003:0002:1
                     SEGSBIT        = 19#,                                                        01001100 T  0003:0002:1
                     SEQBIT         = 20#,                                                        01001110 T  0003:0002:1
                     SEQERRBIT      = 21#,                                                        01001120 T  0003:0002:1
                     SINGLBIT       = 22#,                                                        01001130 T  0003:0002:1
                     STUFFBIT       = 23#,                                                        01001140 T  0003:0002:1
                     VOIDBIT        = 24#,                                                        01001150 T  0003:0002:1
                     VOIDTBIT       = 25#,                                                        01001160 T  0003:0002:1
                     XREFBIT        = 26#,                                                        01001170 T  0003:0002:1
                     BENDBIT        = 27#,                                                        01001171 T  0003:0002:1
                     CODEFILEBIT = 29#,                                  %106-                     01001172 P  0003:0002:1
                     USEROPINX   = 30#;                                  %106-                     01001173 C  0003:0002:1
COMMENT     IF A NEW COMPILER-DEFINED OPTION IS ADDED, CHANGE USEROPINX                           01001180 T  0003:0002:1
            AND ADD OPTION IN DEFINES BELOW, IN DOLLARCARD, AND IN                                01001190 T  0003:0002:1
            FILL STATEMENT IN INITIALIZATION OF COMPILER;                                         01001200 T  0003:0002:1
DEFINE      CHECKTOG        = OPTIONWORD.[CHECKBIT:1] #,                                          01001210 T  0003:0002:1
            DEBUGTOG        = OPTIONWORD.[DEBUGBIT:1] #,                                          01001220 T  0003:0002:1
```

```
                    DECKTOG          = OPTIONWORD.[DECKBIT:1] #,                                    01001230 T  0003:0002:1
                    FORMATOG         = OPTIONWORD.[FORMATBIT:1] #,                                  01001240 T  0003:0002:1
                    INTOG            = OPTIONWORD.[INTBIT:1] #,                                     01001250 T  0003:0002:1
                    LISTATOG         = OPTIONWORD.[LISTABIT:1] #,                                   01001260 T  0003:0002:1
                    LISTOG           = OPTIONWORD.[LISTBIT:1] #,                                    01001270 T  0003:0002:1
                    LISTPTOG         = OPTIONWORD.[LISTPBIT:1] #,                                   01001280 T  0003:0002:1
                    MCPTOG           = OPTIONWORD.[MCPBIT:1] #,                                     01001290 T  0003:0002:1
                    MERGETOG         = OPTIONWORD.[MERGEBIT:1] #,                                   01001300 T  0003:0002:1
                    NESTOG           = OPTIONWORD.[NESTBIT:1] #,                                    01001310 T  0003:0002:1
                    NEWTOG           = OPTIONWORD.[NEWBIT:1] #,                                     01001320 T  0003:0002:1
                    NEWINCL          = OPTIONWORD.[NEWINCLBIT:1] #,                                 01001330 T  0003:0002:1
                    OMITTING         = OPTIONWORD.[OMITBIT:1] #,                                    01001340 T  0003:0002:1
                    PRINTDOLLARTOG   = OPTIONWORD.[PRINTDOLLARBIT:1] #,                             01001350 T  0003:0002:1
                    PRTOG            = OPTIONWORD.[PRTBIT:1] #,                                     01001360 T  0003:0002:1
                    PUNCHTOG         = OPTIONWORD.[PUNCHBIT:1] #,                                   01001370 T  0003:0002:1
                    PURGETOG         = OPTIONWORD.[PURGEBIT:1] #,                                   01001380 T  0003:0002:1
                    SEGSTOG          = OPTIONWORD.[SEGSBIT:1] #,                                    01001390 T  0003:0002:1
                    SEQTOG           = OPTIONWORD.[SEQBIT:1] #,                                     01001400 T  0003:0002:1
          COMMENT   SEQTOG INDICATES RESEQUENCING IS TO BE DONE;                                    01001410 T  0003:0002:1
                    SEQERRTOG        = OPTIONWORD.[SEQERRBIT:1] #,                                  01001420 T  0003:0002:1
                    SINGLTOG         = OPTIONWORD.[SINGLBIT:1] #,                                   01001430 T  0003:0002:1
                    STUFFTOG         = OPTIONWORD.[STUFFBIT:1] #,                                   01001440 T  0003:0002:1
                    VOIDING          = OPTIONWORD.[VOIDBIT:1] #,                                    01001450 T  0003:0002:1
                    VOIDTAPE         = OPTIONWORD.[VOIDTBIT:1] #,                                   01001460 T  0003:0002:1
                    XREF             = OPTIONWORD.[XREFBIT:1] #,                                    01001461 T  0003:0002:1
                    BEND             = OPTIONWORD.[BENDBIT:1]#,                                     01001462 T  0003:0002:1
                    CODEFILE     = OPTIONWORD.[CODEFILEBIT:1]#,                    %106-           01001463 C  0003:0002:1
                    DUMMY            = #;                                                           01001470 T  0003:0002:1
          BOOLEAN NOHEADING;     % TRUE IF DATIME HAS NOT BEEN CALLED.                             01001480 T  0003:0002:1
PRT(73) = NOHEADING
          BOOLEAN NEWBASE;       % NEW BASENUM FOUND ON A NEW $-CARD.                               01001490 T  0003:0002:1
PRT(74) = NEWBASE
          BOOLEAN LASTCRDPATCH;  % NORMALLY FALSE, SET TO TRUE WHEN THE                             01001500 T  0003:0002:1
PRT(75) = LASTCRDPATCH
                                 % LAST CARD FROM SYMBOLIC LIBRARY READ                            01001510 T  0003:0002:1
                                 % IS PATCHED FROM THE CARD READER.                                01001520 T  0003:0002:1
          INTEGER XMODE;         % TELLS DOLLARCARD HOW TO SET OPTIONS.                             01001530 T  0003:0002:1
PRT(76) = XMODE
          BOOLEAN DOLLARTOG;     % TRUE IF SCANNING A DOLLAR CARD.                                  01001540 T  0003:0002:1
PRT(77) = DOLLARTOG
          INTEGER ERRMAX;        % COMPILATION STOPS IF EXCEEDED.                                   01001550 T  0003:0002:1
PRT(100) = ERRMAX
          BOOLEAN SEQXEQTOG;     % GIVE SEQ. NO. WHEN DS-ING OBJ.                                   01001560 T  0003:0002:1
PRT(101) = SEQXEQTOG
          BOOLEAN LISTER;        % LISTOG OR LISTATOG OR DEBUGTOG.                                  01001570 T  0003:0002:1
PRT(102) = LISTER
          ALPHA MEDIUM;          % INPUT IS: T,C,P,CA,CB,CC.                                        01001580 T  0003:0002:1
PRT(103) = MEDIUM
          INTEGER MYCLASS;       % USED IN DOLLARCARD EVALUATION.                                   01001590 T  0003:0002:1
PRT(104) = MYCLASS
          REAL BATMAN;           % USED IN DOLLARCARD EVALUATION.                                   01001600 T  0003:0002:1
PRT(105) = BATMAN
           ARRAY SPECIAL[0:31];                                                                     01003000 T  0003:0002:1
PRT(106) = SPECIAL
              COMMENT THIS ARRAY HOLDS THE INTERNAL CODE FOR THE SPECIAL                            01004000 T  0003:0005:2
                      CHARACTORS; IT IS FILLED DURING INITIALIZATION;                               01005000 T  0003:0005:2
                      SAVE ALPHA ARRAY IDARRAY[0:127];                                              01006000 T  0003:0005:2
PRT(107) = IDARRAY
```

```
                ARRAY INFO[0:31,0:255];
PRT(110) = INFO
                %*************************************************************************
                %              X R E F   S T U F F                                        %116-
                %*************************************************************************
                %                                                                        %116-
                ARRAY                                                                   %116-
                  XREFAY2[0:29],        % ARRAY OF ONE WORD REFERENCE RECORDS.         %116-
PRT(111) = XREFAY2
                                        % THE LAYOUT OF EACH WORD IS                    %116-
                                        %                                              %116-
                                        %         .[1:5]  TYPE OF REFERENCE            %116-
                                        %                   = 0 FOR FORWARD DECL       %116-
                                        %                   = 1 FOR LABEL OCCURENCE    %116-
                                        %                   = 2 FOR NORMAL DECL        %116-
                                        %                   = 4 FOR NORMAL REFERENCE   %116-
                                        %                   = 5 FOR ASSIGNMENT         %116-
                                        %                                              %116-
                                        %                    NOTE: THE LOWER ORDER BIT
                                        %                          OF THIS FIELD IS ON
                                        %                          IF YOU WANT STARS%116-
                                        %                          AROUND THIS REFERENCE
                                        %                          IN THE XREF       %116-
                                        %                                              %116-
                                        %         .[6:15] IDENTIFIER ID. NO.           %116-
                                        %                   THIS IS A UNIQUE NUMBER THAT
                                        %                   IS ASSIGNED WHEN THE       %116-
                                        %                   IDENTIFIER IS ENCOUNTERE   %116-
                                        %                   FOR THE FIRST TIME.        %116-
                                        %                                              %116-
                                        %         .[21:27] SEQUENCE NUMBER             %116-
                                        %                                              %116-
                  XREFAY1[0:9],         % RECORD BUFFER AREA FOR WRITING OUT THE      %116-
PRT(112) = XREFAY1
                                        % NAME INFORMATION RECORDS, ONE RECORD        %116-
                                        % IS WRITTEN FOR EACH IDENTIFIER IN THE SYMBOL
                                        % TABLE WHEN THE IDENTIFIER IS PURGED FROM THE
                                        % SYMBOL TABLE, I.E., WHEN LEAVING THE BLOCK%116-
                                        % IN WHICH THE IDENTIFIER IS DECLARED.        %116-
                                        %                                              %116-
                                        % THE LAYOUT OF EACH RECORD IS:               %116-
                                        %                                              %116-
                                        %      WORDS 0-7    THE IDENTIFIER WITH BLANK%116-
                                        %                     FILL ON THE RIGHT        %116-
                                        %                                              %116-
                                        %      WORD 8                                  %116-
                                        %        .[21:12]   SEGMENT NUMBER IN WHICH  %116-
                                        %                     THIS IDENTIFIER WAS DECLARED
                                        %                                              %116-
                                        %        .[33:15]   IDENTIFIER ID. NO.         %116-
                                        %                                              %116-
                                        %      WORD 9       ELBAT WORD                 %116-
                                        %                                              %116-
                  XINFO[0:31,0:127];    % THIS ARRAY CONTAINS ONE ENTRY FOR EACH ENTRY
PRT(113) = XINFO
                                        % IN THE INFO TABLE, IF YOU HAVE THE INDEX  %116-
                                        % OF THE ELBAT WORD FOR AN IDENTIFIER IN    %116-
```

```
% THE INFO TABLE YOU CAN FIND THE XINFO WORD%116-        01007260 C  0003:001810
% FOR THE IDENTIFIER  BY REFERRING TO:      %116-        01007265 C  0003:001810
%                                           %116-        01007270 C  0003:001810
%      XINFO[INDEX.LINKR,INDEX.LINKC DIV 2] %116-        01007275 C  0003:001810
%                                           %116-        01007280 C  0003:001810
% EACH ENTRY CONTAINS:                      %116-        01007285 C  0003:001810
%                                           %116-        01007290 C  0003:001810
%            .[21:12]  SEGMENT NUMBER IN WHICH          01007295 C  0003:001810
%                        THIS IDENTIFIER WAS DECL       01007300 C  0003:001810
%                                           %116-        01007305 C  0003:001810
%            .[33:15]  IDENTIFIER ID. NO.   %116-        01007310 C  0003:001810
%                        IF THIS ID. NO. IS ZERO        01007315 C  0003:001810
%                        THEN XREF WAS NOT ON           01007320 C  0003:001810
%                        AT THE TIME THE IDENT          01007325 C  0003:001810
%                        WAS DECLARED AND ALL           01007330 C  0003:001810
%                        FUTURE REFERENCES WILL         01007335 C  0003:001810
%                        BE DISCARDED.      %116-       01007340 C  0003:001810
%                                           %116-        01007345 C  0003:001810
%                                           %116-        01007350 C  0003:001810
% CONTAINS INDEX OF NEXT AVAILABLE SLOT IN  %116-        01007355 C  0003:001810

% XREFAY2. WHEN THIS BECOMES GREATER        %116-        01007360 C  0003:001810
% THAN 30 THE CURRENT ARRAY IS DUMPED TO DISK            01007365 C  0003:001810
% AND XREFPT IS RESET TO ZERO.              %116-        01007370 C  0003:001810
%                                           %116-        01007375 C  0003:001810
% THIS VARIABLE CONTROLS THE ASSIGNING OF   %116-        01007380 C  0003:001810

% ID. NO. TO IDENTIFIERS. IT IS INCREMENTED %116-        01007385 C  0003:001810
% EACH TIME A NEW IDENTIFIER IS ENCOUNTERED.%116-        01007390 C  0003:001810
%                                           %116-        01007395 C  0003:001810
DEFINE                                      %116-        01007400 C  0003:001810
  SEGNOF = [21:12]#,    % FIELDS IN XINFO ENTRIES AND WORD 8 OF %116-  01007405 C  0003:001810
  IDNOF  = [33:15]#,    % IDENTIFIER RECORDS.               %116-      01007410 C  0003:001810
                        %                                  %116-      01007415 C  0003:001810
  TYPEREF = [1:5]#,     % FIELDS OF REFERENCE WORDS         %116-      01007420 C  0003:001810
  REFIDNOF =[6:15]#,    %                                  %116-      01007425 C  0003:001810
  SEQNOF = [21:27]#,    %                                  %116-      01007430 C  0003:001810
                        %                                  %116-      01007435 C  0003:001810
  XREFIT(INDEX,SEQNO,REFTYPE) = % DEFINE TO ADD INFO TO REF TABLE %116- 01007440 C  0003:001810
    BEGIN IF XREF THEN CROSSREFIT(INDEX,SEQNO,REFTYPE); END#,  %116-    01007445 C  0003:001810
                        %                                  %116-      01007450 C  0003:001810
  XMARK(REFTYPE) =      % DEFINE TO CHANGE LAST ENTRY IN REF TABLE TO A 01007455 C  0003:001810
    BEGIN IF XREF THEN XREFAY2[XREFPT-1].TYPEREF := REFTYPE END#,  %116- 01007460 C  0003:001810
                        %                                  %116-      01007465 C  0003:001810
  XREFDUMP(INDEX) =     % DEFINE TO DUMP SYMBOL TABLE INFO FOR IDENTIFIER 01007470 C  0003:001810
    BEGIN IF DEFINING.[1:1] THEN CROSSREFDUMP(INDEX); END#,    %116-    01007475 C  0003:001810
                        %                                  %116-      01007480 C  0003:001810
  XREFINFO[INDEX] =     % DEFINE TO TRANSLATE INFO ROW AND COLUMN TO %116- 01007481 C  0003:001810
    XINFO[(INDEX).LINKR,(INDEX).LINKC DIV 2]#, % XINFO ROW AND COL %116- 01007482 C  0003:001810
                        %                                  %116-      01007483 C  0003:001810
  FORWARDREF = 0#,      % DEFINES FOR DIFFERENT REFERENCE TYPES %116-  01007485 C  0003:001810
  LBLREF     = 1#,      %                                  %116-      01007486 C  0003:001810
  DECLREF    = 2#,      %                                  %116-      01007490 C  0003:001810
  NORMALREF  = 4#,      %                                  %116-      01007495 C  0003:001810
  ASSIGNREF  = 5#;      %                                  %116-      01007500 C  0003:001810
    ARRAY BEGINSTACK[0:255]; INTEGER BSPOINT;                         01007600 T  0003:001810
```

```
                INTEGER
                  XREFPT,
PRT(114) = XREFPT


                  XLUN;
PRT(115) = XLUN
```

```
PRT(116) = BEGINSTACK
PRT(117) = BSPOINT
```

```
                    BOOLEAN DEFINING;                                                    01007650 T  0003:0020:1
PRT(120) = DEFINING
                    COMMENT INFO CONTAINS ALL THE INFORMATION ABOUT A GIVEN IDENTIFIER    01008000 T  0003:0020:1
                            OR RESERVED WORD.  THE FIRST WORD OF A GIVEN ENTRY IS         01009000 T  0003:0020:1
                            THE INTERNAL CODE ( OR ELBAT WORD AS IT IS USUALLY            01010000 T  0003:0020:1
                            CALLED).  THE SECOND WORD CONTAINS THE FORWARD BIT (IN        01011000 T  0003:0020:1
                            [1:1]) FOR PROCEDURES, THE LINK TO PREVIOUS ENTRY (IN         01012000 T  0003:0020:1
                            [4:8]), THE NUMBER OF CHARACTORS IN THE ALPHA REPRESENTA-      01013000 T  0003:0020:1
                            TION (IN [12:6]), AND THE FIRST 5 CHARACTORS OF ALPHA.        01014000 T  0003:0020:1
                            SUCCEDING WORDS CONTAIN THE REMAINING CHARACTORS OF ALPHA,     01015000 T  0003:0020:1
                            FOLLOWED BY ANY ADDITIONAL INFORMATION. THE ELBAT WORD        01016000 T  0003:0020:1
                            AND THE ALPHA FOR ANY QUANTITY ARE NOT SPLIT ACROSS A ROW     01017000 T  0003:0020:1
                            OF INFO.  FOR PURPOSES OF FINDING AN IDENTIFIER OR            01018000 T  0003:0020:1
                            RESERVED WORD THE QUANTITIES ARE SCATTERED INTO 125           01019000 T  0003:0020:1
                            DIFERENT LISTS OR STACKES. WHICH STACK CONTAINS A QUANTITY    01020000 T  0003:0020:1
                            IS GIVEN BY TAKING NAAAAA MOD 125 WHERE N IS THE NUMBER       01021000 T  0003:0020:1
                            OF CHARACTORS AND AAAAA IS THE FIRST 5 CHARACTORS OF          01022000 T  0003:0020:1
                            ALPHA, FILLED IN WITH ZEROS FROM THE RIGHT IF NEEDED.         01023000 T  0003:0020:1
                            THIS NUMBER IS CALLED THE SCRAMBLE NUMBER OR INDEX.           01024000 T  0003:0020:1
                            THE FIRST ROW OF INFO IS USED FOR OTHER PURPOSES.  THE        01025000 T  0003:0020:1
                            RESERVED WORDS OCCUPY THE SECOND ROW. IT IS FILLED DURING     01026000 T  0003:0020:1
                            INITIALIZATION;                                               01027000 T  0003:0020:1
                    COMMENT                 INFO FORMAT                                   01028000 T  0003:0020:1
                      FOLLOWING IS A DESCRIPTION OF THE FORMAT OF ALL TYPES OF ENTRIES     01029000 T  0003:0020:1
                      ENTERED IN INFO:                                                    01030000 T  0003:0020:1
                         THE FIRST WORD OF ALL ENTRIES IS THE ELBAT WORD.                 01031000 T  0003:0020:1
                         THE INCR FIELD ([27:8]) CONTAINS AN INCREMENT WHICH WHEN         01032000 T  0003:0020:1
                         ADDED TO THE CURRENT INDEX INTO INFO YELDSAN INDEX TO ANY        01033000 T  0003:0020:1
                         ADDITIONAL INFO (IF ANY) FOR THIS ENTRY.                         01034000 T  0003:0020:1
                       E.G. IF THE INDEX IS  IX  THEN  INFO[(IX+INCR).LINKR.(IX+INCR).    01035000 T  0003:0020:1
                           LINKC] WILL CONTAIN THE FIRST WORD OF ADDITIONAL INFO.          01036000 T  0003:0020:1
                      THE LINK FIELD OF THE ELBAT WORD IN INFO IS DIFFERENT FROM        01037000 T  0003:0020:1
                      THAT OF THE ENTRY IN ELBAT PUT IN BY TABLE.THE ENTRY IN ELBAT     01038000 T  0003:0020:1
                      POINTS TO ITS OWN LOCATION (RELATIVE) IN INFO.                    01039000 T  0003:0020:1
                      THE LINK IN INFO POINTS TO THE PREVIOUS ENTRY E.G.,THE            01040000 T  0003:0020:1
                      LINK FROM STACKHEAD WHICH THE CURRENT ENTRY REPLACED.             01041000 T  0003:0020:1
                      FOR SIMPLICITY,I WILL CONSIDER INFO TO BE A ONE DIMENSIONAL       01042000 T  0003:0020:1
                      ARRAY,SO THAT THE BREAKING UP OF THE LINKS INTO ROW AND COLUMN     01043000 T  0003:0020:1
                      WILL NOT DETRACT FROM THE DISCUSSION.                             01044000 T  0003:0020:1
                         ASSUME THAT THREE IDENTIFIERS A,B,AND C "SCRAMBLE" INTO        01045000 T  0003:0020:1
                      THE SAME STACKHEAD LOCATION IN THE ORDER OF APPEARANCE.           01046000 T  0003:0020:1
                        FURTHER ASSUME THERE ARE NO OTHER ENTRIES CONNECTED TO          01047000 T  0003:0020:1
                      THIS STACKHEAD INDEX. LET THIS STACKHEAD LOCATION BE              01048000 T  0003:0020:1
                                    S[L]                                                01049000 T  0003:0020:1
                      NOW THE DECLARATION                                               01050000 T  0003:0020:1
                           BEGIN REAL A,B,C  IS ENCOUNTERED                             01051000 T  0003:0020:1
                          IF THE NEXT AVAILABLE INFO SPACE IS CALLED NEXTINFO           01052000 T  0003:0020:1
                       THEN  A IS ENTERED AS FOLLOWS:(ASSUME AN ELBAT WORD T HAS BEEN   01053000 T  0003:0020:1
                                                      CONSTRUCTED FOR A)                01054000 T  0003:0020:1
                                    T.LINK← S[L]. (WHICH IS ZERO AT FIRST).             01055000 T  0003:0020:1
INFO[NEXTINFO]←T.                   S[L]←NEXTINFO.                                      01056000 T  0003:0020:1
                                    NEXTINFO←NEXTINFO+NUMBER OF WORDS IN THIS           01057000 T  0003:0020:1
                                                     ENTRY.                             01058000 T  0003:0020:1
                        NOW  S[L] POINTS TO THE ENTRY FOR A IN INFO AND THE ENTRY       01059000 T  0003:0020:1
                             ITSELF CONTAINS THE STOP FLAG ZERO.                        01060000 T  0003:0020:1
                      B IS ENTERED SIMILARLY TO A.                                      01061000 T  0003:0020:1
                       NOW  S[L] POINTS TO THE ENTRY FOR B AND IT POINTS TO THE          01062000 T  0003:0020:1
```

```
                                ENTRY FOR A.
     SIMILARLY,AFTER C IS ENTERED
               S[L] POINTS TO C,WHOSE ENTRY POINTS TO B WHOSE ENTRY
                    POINTS TO A.
  THE SECOND WORD OF EACH ENTRY IN INFO IS MADE UP AS FOLLOWS:
     FWDPT =[1:1],THIS TELLS WHETHER A PROCEDURE WAS DECLARED
                  FORWARD.IT IS RESET AT THE TIME OF ITS ACTUAL
                  FULL DECLARATION.
     PURPT =[4:8] THIS GIVES A DECREMENT WHICH GIVES THE RELATIVE
                  INDEX TO THE PREVIOUS INFO ENTRY WHEN SUBTRACTED
                  FROM THE CURRENT ENTRY INDEX.
    [12:6] TELLS THE NUMBER OF CHARACTERS IN THE ENTRY.(<64)
    [18:30] CONTAINS THE FIRST FIVE ALPHA CHARACTERS OF THE ENTRY
           AND SUCCEEDING WORDS CONTAIN ALL OVERFLOW IF NEEDED.
           THESE WORDS CONTAIN 8 CHARACTERS EACH,LEFT JUSTIFIED.
          THUS,AN ENTRY FOR  SYMBOL FOLLOWED BY AN ENTRY
                         FOR  X      WOULD APPEAR AS FOLLOWS:
    INFO[I] =      ELBATWRD      (MADE FOR SYMBOL)
        I+1 =      OP6SYMBO     (P DEPENDS ON PREVIOUS ENTRY)
        I+2 =      L
        I+3 =      ELBATWRD      (MADE FOR X)
        I+4 =      031X
      THIS SHOWS THAT INFO[I-P] WOULD POINT TO THE BEGINNING OF
      THE ENTRY BEFORE SYMBOL, AND
      INFO[I+3-3] POINTS TO THE ENTRY FOR SYMBOL.
ALL ENTRIES OF IDNETIFIERS HAVE THE INFORMATION DESCRIBED ABOVE
THAT IS,THE ELBAT WORD FOLLOWED BY THE WORD CONTAING THE FIRST
FIVE CHARACTERS OF ALPHA,AND ANY ADDITIONAL WORDS OF ALPHA IF
 NECESSARY.
THIS IS SUFFICIENT FOR ENTRIES OF THE FOLLOWING TYPES.
       REAL
       BOOLEAN
       INTEGER
       ALPHA
       FILE
       FORMAT
       LIST
   OTHER ENTRIES REQUIRE ADDITIONAL INFORMATION.
ARRAYS:
     THE FIRST WORD OF ADDITIONAL INFO CONTAINS THE NUMBER OF
     DIMENSIONS(IN THE LOW ORDER PART).[40:8]
    EACH SUCCEEDING WORD CONTAINS INFORMATION ABOUT EACH LOWER
    BOUND IN ORDER OF APPEARANCE,ONE WORD FOR EACH LOWER BOUND.
    THESE WORDS ARE MADE UP AS FOLLOWS:
          [23:12] =ADD OPERATOR SYLLABLE (0101) OR
                   SUB OPERATOR SYLLABLE (0301) CORRESPONDING
                   RESPECTIVELY TO WHETHER THE LOWER BOUND IS
                   TO BE ADDED TO THE SUBSCRIPT IN INDEXING OR
                   SUBTRACTED.
          [35:11] =11 BIT ADDRESS OF LOWER BOUND,IF THE LOWER BOUND
                   REQUIRES A PRT OR STACK CELL,OTHERWISE THE BIT
                   35 IS IGNORED AND THE NEXT TEN BITS([36:10])
                   REPRESENT THE ACTUAL VALUE OF THE LOWER BOUND
           [46:2] =00 OR 10  DEPENDING ON WHETHER THE [35:11] VALUE
                   IS A LITERAL OR OPERAND,RESPECTIVELY.
PROCEDURES:
      THE FIRST WORD OF ADDITIONAL INFO CONTAINS THE NUMBER OF
      PARAMETERS [40:8]
```

```
01063000 T  0003:0020:1
01064000 T  0003:0020:1
01065000 T  0003:0020:1
01066000 T  0003:0020:1
01067000 T  0003:0020:1
01068000 T  0003:0020:1
01069000 T  0003:0020:1
01070000 T  0003:0020:1
01071000 T  0003:0020:1
01072000 T  0003:0020:1
01073000 T  0003:0020:1
01074000 T  0003:0020:1
01075000 T  0003:0020:1
01076000 T  0003:0020:1
01077000 T  0003:0020:1
01078000 T  0003:0020:1
01079000 T  0003:0020:1
01080000 T  0003:0020:1
01081000 T  0003:0020:1
01082000 T  0003:0020:1
01083000 T  0003:0020:1
01084000 T  0003:0020:1
01085000 T  0003:0020:1
01086000 T  0003:0020:1
01087000 T  0003:0020:1
01088000 T  0003:0020:1
01089000 T  0003:0020:1
01090000 T  0003:0020:1
01091000 T  0003:0020:1
01092000 T  0003:0020:1
01093000 T  0003:0020:1
01094000 T  0003:0020:1
01095000 T  0003:0020:1
01096000 T  0003:0020:1
01097000 T  0003:0020:1
01098000 T  0003:0020:1
01099000 T  0003:0020:1
01100000 T  0003:0020:1
01101000 T  0003:0020:1
01102000 T  0003:0020:1
01103000 T  0003:0020:1
01104000 T  0003:0020:1
01105000 T  0003:0020:1
01106000 T  0003:0020:1
01107000 T  0003:0020:1
01108000 T  0003:0020:1
01109000 T  0003:0020:1
01110000 T  0003:0020:1
01111000 T  0003:0020:1
01112000 T  0003:0020:1
01113000 T  0003:0020:1
01114000 T  0003:0020:1
01115000 T  0003:0020:1
01116000 T  0003:0020:1
01117000 T  0003:0020:1
01118000 T  0003:0020:1
01119000 T  0003:0020:1
01120000 T  0003:0020:1
```

```
      IF A STREAM PROCEDURE THEN THIS WORD CONTAINS ALSO IN
      [13:11] ENDING  PRT ADDRESS FOR LABELS.
      [ 7:6]  NO OF LABELS REQUIRING PRT ADDRESSES, AND [1:6] NUMBER
      OF LOCALS.
      SUCCEEDING WORDS (ONE FOR EACH FORMAL PARAMETER,IN ORDER
                        OF APPEARANCE IN FORMAL PARAPART) ARE
      ELBAT WORDS SPECIFYING TYPE OF EACH PARAMETER AND WHETHER
      VALUE OR NOT([10:1]).
      THE ADDRESS([16:11]) IS THE F- ADDRESS FOR EACH,
      IF THE PARAMETER IS AN ARRAY THEN THE INCR FIELD([27:8])
      CONTAINS THE NUMBER OF DIMENSIONS.OTHERWISE INCR IS MEANINGLESS.
      LINK([35:13]) IS MEANINGLESS.
      IF A STREAM PROCEDURE THEN THE CLASS OF EACH PARAMATER IS
      THAT OF LOCAL ID OR FILE ID, DEPENDING ON WHETHER OR NOT A RELEASE
      IS DONE IN THE STREAM PROCEDURE.
LABELS:
      AT DECLARATION TIME THE ADDITIONAL INFO CONTAINS 0.  THE SIGN
      BIT TELLS WHETHER OR NOT THE DEFINITION POINT HAS BEEN REACHED.
      IF SIGN = 0, THEN [36:12] CONTAINS AN ADDRESS IN CODEARRAY OF A
      LIST OF FORWARD REFERENCES TO THIS LABEL.  THE END OF LIST FLAG IS
      0.  IF SIGN =0, THEN [36:12] CONTAINS L FOR THIS LABEL.
SWITCHES:
      THE FIELD [36:12] CONTAINS L FOR THE BEGINNING OF SWITCH DECLAR-
      ATION.  [24:12] CONTAINS L FOR FIRST SIMPLE REFERENCE TO SWITCH.
      IF SWITCH IS NOT SIMPLE, IT IS MARKED FORMAL.  HERE SIMPLE MEANS
      NO POSSIBILITY OF JUMPING OUT OF A BLOCK.                        ;
            DEFINE MON    =[ 1: 1]#,
                   CLASS  =[ 2: 7]#,
                   FORMAL =[ 9: 1]#,
                   VO     =[10: 1]#,
                   LVL    =[11: 5]#,
                   ADDRESS=[16:11]#,
                   INCR   =[27: 8]#,
                   LINK   =[35:13]#,
                   DYNAM  =[11:16]#,
                   SBITF  =[21:6]#, % STARTING BIT FOR FIELD ID,  %117-
                   NBITF  =[27:6]#, % NUMBER OF BITS FOR FIELD ID,%117-
                   LINKR  =[35: 5]#,
                   LINKC  =[40: 8]#;
COMMENT THESE DEFINES ARE USED TO PICK APART THE ELBAT WORD,
        MON IS THE BIT WHICH IS TURNED ON IF:
            1. THE QUANTITY IS TO BE MONITORED, OR
            2. THE QUANTITY IS A PARAMETRIC DEFINE AND NOT
               A DEFINE WITHOUT PARAMETERS.
        CLASS IS THE PRINCIPAL IDENTIFICATION OF A GIVEN
            QUANTITY.
        FORMAL IS THE BIT WHICH IS ON IF THE QUANTITY IS A FORMAL
            PARAMETER.
        VO IS THE VALUE-OWN BIT, IF FORMAL = 1 THEN THE BIT
            DISTINGUISHES VALUE PARAMETERS FROM OTHERS. IF
            FORMAL = 0 THEN THE BIT DISTINGUISHES OWN VARIABLES
            FROM OTHERS,
        LVL GIVES THE LEVEL AT WHICH A QUANTITY WAS DECLARED,
        ADDRESS GIVES THE STACK OR PRT ADDRESS,
        DYNAM IS USED INSTEAD OF LVL AND ADDRESS FOR DEFINE AND
            DEFINE PARAMETER ENTRIES, ONLY, IT IS AN INDEX
            INTO THE ARRAY CONTAINING THE DEFINE TEXT,
```

```
01121000 T  0003:0020:1
01122000 T  0003:0020:1
01123000 T  0003:0020:1
01124000 T  0003:0020:1
01125000 T  0003:0020:1
01126000 T  0003:0020:1
01127000 T  0003:0020:1
01128000 T  0003:0020:1
01129000 T  0003:0020:1
01130000 T  0003:0020:1
01131000 T  0003:0020:1
01132000 T  0003:0020:1
01133000 T  0003:0020:1
01134000 T  0003:0020:1
01135000 T  0003:0020:1
01136000 T  0003:0020:1
01137000 T  0003:0020:1
01138000 T  0003:0020:1
01139000 T  0003:0020:1
01140000 T  0003:0020:1
01141000 T  0003:0020:1
01142000 T  0003:0020:1
01143000 T  0003:0020:1
01144000 T  0003:0020:1
01145000 T  0003:0020:1
01146000 T  0003:0020:1
01147000 T  0003:0020:1
01148000 T  0003:0020:1
01149000 T  0003:0020:1
01150000 T  0003:0020:1
01151000 T  0003:0020:1
01152000 T  0003:0020:1
01153000 T  0003:0020:1
01154000 T  0003:0020:1
01154100 T  0003:0020:1
01154200 C  0003:0020:1
01154300 C  0003:0020:1
01155000 T  0003:0020:1
01156000 T  0003:0020:1
01157000 T  0003:0020:1
01158000 T  0003:0020:1
01158100 T  0003:0020:1
01158200 T  0003:0020:1
01158300 T  0003:0020:1
01159000 T  0003:0020:1
01160000 T  0003:0020:1
01161000 T  0003:0020:1
01162000 T  0003:0020:1
01163000 T  0003:0020:1
01164000 T  0003:0020:1
01165000 T  0003:0020:1
01166000 T  0003:0020:1
01167000 T  0003:0020:1
01168000 T  0003:0020:1
01168100 T  0003:0020:1
01168200 T  0003:0020:1
01168300 T  0003:0020:1
```

```
                    THEREFORE, WHEN THE COMPILER CHECKS TO SEE IF A                  01168400 T  0003!0020!1
                    DEFINE WAS DECLARED B4 IN THE SAME BLOCK, IT DOES                01168500 T  0003!0020!1
                    NOT USE THE LVL FIELD, BUT MAKES USE OF NINFOO,                  01168600 T  0003!0020!1
                INCR GIVES A RELATIVE LINK TO ANY ADDITIONAL INFORMATION             01169000 T  0003!0020!1
                    NEEDED, RELATIVE TO THE LOCATION IN INFO,                        01170000 T  0003!0020!1
                LINK CONTAINS A LINK TO THE LOCATION IN INFO IF THE                  01171000 T  0003!0020!1
                    QUANTITY LIES IN ELBAT, OTHERWISE IT LINKS TO THE                01172000 T  0003!0020!1
                    NEXT ITEM IN THE STACK, ZERO IS AN END FLAG,                     01173000 T  0003!0020!1
                LINKR AND LINKC ARE SUBDIVISIONS OF LINK.;                           01174000 T  0003!0020!1
COMMENT  CLASSES FOR ALL QUANTITIES  -  OCTAL CLASS IS IN COMMENT;                   01175000 T  0003!0020!1
   COMMENT   CLASSES FOR IDENTIFIERS;                                                01176000 T  0003!0020!1
DEFINE  UNKNOWNID       =00#,    COMMENT 000;                                        01177000 T  0003!0020!1
        STLABID         =01#,    COMMENT 001;                                        01178000 T  0003!0020!1
        LOCLID          =02#,    COMMENT 002;                                        01179000 T  0003!0020!1
        DEFINEDID       =03#,    COMMENT 003;                                        01180000 T  0003!0020!1
        LISTID          =04#,    COMMENT 004;                                        01181000 T  0003!0020!1
        FRMTID          =05#,    COMMENT 005;                                        01182000 T  0003!0020!1
        SUPERFRMTID     =06#,    COMMENT 006;                                        01183000 T  0003!0020!1
        FILEID          =07#,    COMMENT 007;                                        01184000 T  0003!0020!1
        SUPERFILEID     =08#,    COMMENT 010;                                        01185000 T  0003!0020!1
        SWITCHID        =09#,    COMMENT 011;                                        01186000 T  0003!0020!1
        PROCID          =10#,    COMMENT 012;                                        01187000 T  0003!0020!1
        INTRNSICPROCID  =11#,    COMMENT 013;                                        01188000 T  0003!0020!1
        STRPROCID       =12#,    COMMENT 014;                                        01189000 T  0003!0020!1
        BOOSTRPROCID    =13#,    COMMENT 015;                                        01190000 T  0003!0020!1
        REALSTRPROCID   =14#,    COMMENT 016;                                        01191000 T  0003!0020!1
        ALFASTRPROCID   =15#,    COMMENT 017;                                        01192000 T  0003!0020!1
        INTSTRPROCID    =16#,    COMMENT 020;                                        01193000 T  0003!0020!1
        BOOPROCID       =17#,    COMMENT 021;                                        01194000 T  0003!0020!1
        REALPROCID      =18#,    COMMENT 022;                                        01195000 T  0003!0020!1
        ALFAPROCID      =19#,    COMMENT 023;                                        01196000 T  0003!0020!1
        INTPROCID       =20#,    COMMENT 024;                                        01197000 T  0003!0020!1
        BOOID           =21#,    COMMENT 025;                                        01198000 T  0003!0020!1
        REALID          =22#,    COMMENT 026;                                        01199000 T  0003!0020!1
        ALFAID          =23#,    COMMENT 027;                                        01200000 T  0003!0020!1
        INTID           =24#,    COMMENT 030;                                        01201000 T  0003!0020!1
        BOOARRAYID      =25#,    COMMENT 031;                                        01202000 T  0003!0020!1
        REALARRAYID     =26#,    COMMENT 032;                                        01203000 T  0003!0020!1
        ALFAARRAYID     =27#,    COMMENT 033;                                        01204000 T  0003!0020!1
        INTARRAYID      =28#,    COMMENT 034;                                        01205000 T  0003!0020!1
        LABELID         =29#,    COMMENT 035;                                        01206000 T  0003!0020!1
COMMENT   CLASSES FOR PRIMARY BEGINNERS;                                             01207000 T  0003!0020!1
        TRUTHV          =30#,    COMMENT 036;                                        01208000 T  0003!0020!1
        NONLITNO        =31#,    COMMENT 037;                                        01209000 T  0003!0020!1
        LITNO           =32#,    COMMENT 040;                                        01210000 T  0003!0020!1
        STRNGCON        =33#,    COMMENT 041;                                        01211000 T  0003!0020!1
        LEFTPAREN       =34#,    COMMENT 042;                                        01212000 T  0003!0020!1
COMMENT   CLASS FOR ALL DECLARATORS;                                                 01213000 T  0003!0020!1
        DECLARATORS     =35#,    COMMENT 043;                                        01214000 T  0003!0020!1
COMMENT   CLASSES FOR STATEMENT BEGINNERS                                            01215000 T  0003!0020!1
        READV           =36#,    COMMENT 044;                                        01216000 T  0003!0020!1
        WRITEV          =37#,    COMMENT 045;                                        01217000 T  0003!0020!1
        SPACEV          =38#,    COMMENT 046;                                        01218000 T  0003!0020!1
        CLOSEV          =39#,    COMMENT 047;                                        01219000 T  0003!0020!1
        LOCKV           =40#,    COMMENT 050;                                        01220000 T  0003!0020!1
        REWINDV         =41#,    COMMENT 051;                                        01221000 T  0003!0020!1
        DOUBLEV         =42#,    COMMENT 052;                                        01222000 T  0003!0020!1
```

```
        FORV            =43#,   COMMENT 053;                                    01223000 T  0003:0020:1
        WHILEV          =44#,   COMMENT 054;                                    01224000 T  0003:0020:1
        DOV             =45#,   COMMENT 055;                                    01225000 T  0003:0020:1
        UNTILV          =46#,   COMMENT 056;                                    01226000 T  0003:0020:1
        ELSEV           =47#,   COMMENT 057;                                    01227000 T  0003:0020:1
        ENDV            =48#,   COMMENT 060;                                    01228000 T  0003:0020:1
        FILLV           =49#,   COMMENT 061;                                    01229000 T  0003:0020:1
        SEMICOLON       =50#,   COMMENT 062;                                    01230000 T  0003:0020:1
        IFV             =51#,   COMMENT 063;                                    01231000 T  0003:0020:1
        GOV             =52#,   COMMENT 064;                                    01232000 T  0003:0020:1
        RELEASEV        =53#,   COMMENT 065;                                    01233000 T  0003:0020:1
        BEGINV          =54#,   COMMENT 066;                                    01234000 T  0003:0020:1
    COMMENT   CLASSES FOR STREAM RESERVED WORDS;                                01235000 T  0003:0020:1
        SIV             =55#,   COMMENT 067;                                    01236000 T  0003:0020:1
        DIQ             =56#,   COMMENT 070;                                    01237000 T  0003:0020:1
        CIV             =57#,   COMMENT 071;                                    01238000 T  0003:0020:1
        TALLYV          =58#,   COMMENT 072;                                    01239000 T  0003:0020:1
        DSV             =59#,   COMMENT 073;                                    01240000 T  0003:0020:1
        SKIPV           =60#,   COMMENT 074;                                    01241000 T  0003:0020:1
        JUMPV           =61#,   COMMENT 075;                                    01242000 T  0003:0020:1
        DBV             =62#,   COMMENT 076;                                    01243000 T  0003:0020:1
        SBV             =63#,   COMMENT 077;                                    01244000 T  0003:0020:1
        TOGGLEV         =64#,   COMMENT 100;                                    01245000 T  0003:0020:1
        SCV             =65#,   COMMENT 101;                                    01246000 T  0003:0020:1
        LOCV            =66#,   COMMENT 102;                                    01247000 T  0003:0020:1
        DCV             =67#,   COMMENT 103;                                    01248000 T  0003:0020:1
        LOCALV          =68#,   COMMENT 104;                                    01249000 T  0003:0020:1
        LITV            =69#,   COMMENT 105;                                    01250000 T  0003:0020:1
        TRNSFER         =70#,   COMMENT 106;                                    01251000 T  0003:0020:1
    COMMENT   CLASSES FOR VARIOUS MISCELLANEOUS QUANTITIES;                     01252000 T  0003:0020:1
        COMMENTV        =71#,   COMMENT 107;                                    01253000 T  0003:0020:1
        FORWARDV        =72#,   COMMENT 110;                                    01254000 T  0003:0020:1
        STEPV           =73#,   COMMENT 111;                                    01255000 T  0003:0020:1
        THENV           =74#,   COMMENT 112;                                    01256000 T  0003:0020:1
        TOV             =75#,   COMMENT 113;                                    01257000 T  0003:0020:1
        VALUEV          =76#,   COMMENT 114;                                    01258000 T  0003:0020:1
        WITHV           =77#,   COMMENT 115;                                    01259000 T  0003:0020:1
        COLON           =78#,   COMMENT 116;                                    01260000 T  0003:0020:1
        COMMA           =79#,   COMMENT 117;                                    01261000 T  0003:0020:1
        CROSSHATCH      =80#,   COMMENT 120;                                    01262000 T  0003:0020:1
        LFTBRKET        =81#,   COMMENT 121;                                    01263000 T  0003:0020:1
        PERIOD          =82#,   COMMENT 122;                                    01264000 T  0003:0020:1
        RTBRKET         =83#,   COMMENT 123;                                    01265000 T  0003:0020:1
        RTPAREN         =84#,   COMMENT 124;                                    01266000 T  0003:0020:1
    COMMENT   CLASSES FOR OPERATORS;                                            01267000 T  0003:0020:1
        NOTOP           =85#,   COMMENT 125;                                    01268000 T  0003:0020:1
        ASSIGNOP        =86#,   COMMENT 126;                                    01269000 T  0003:0020:1
        AMPERSAND       =87#,   COMMENT 127;                                    01270000 T  0003:0020:1
        EQVOP           =88#,   COMMENT 130;                                    01271000 T  0003:0020:1
        IMPOP           =89#,   COMMENT 131;                                    01272000 T  0003:0020:1
        OROP            =90#,   COMMENT 132;                                    01273000 T  0003:0020:1
        ANDOP           =91#,   COMMENT 133;                                    01274000 T  0003:0020:1
        RELOP           =92#,   COMMENT 134;                                    01275000 T  0003:0020:1
        ADOP            =93#,   COMMENT 135;                                    01276000 T  0003:0020:1
        MULOP           =94#,   COMMENT 136;                                    01277000 T  0003:0020:1
        FACTOP          =95#,   COMMENT 137;                                    01278000 T  0003:0020:1
        STRING          =99#,   COMMENT 143;                                    01278050 T  0003:0020:1
```

```
                FIELDID          =125#,   COMMENT 175;                    %117-   01278090 C  0003:0020:1
                FAULTID          =126#,   COMMENT 176;                            01278100 T  0003:0020:1
                SUPERLISTID      =127#,   COMMENT 177;                            01278500 T  0003:0020:1
            COMMENT   SUBCLASSES FOR DECLARATORS (KEPT IN ADDRESS);               01279000 T  0003:0020:1
                OWNV             =01#,    COMMENT  01;                            01280000 T  0003:0020:1
                SAVEV            =02#,    COMMENT  02;                            01281000 T  0003:0020:1
                BOOV             =03#,    COMMENT  03;                            01282000 T  0003:0020:1
                REALV            =04#,    COMMENT  04;                            01283000 T  0003:0020:1
                ALFAV            =05#,    COMMENT  05;                            01284000 T  0003:0020:1
                INTV             =06#,    COMMENT  06;                            01285000 T  0003:0020:1
                LABELV           =07#,    COMMENT  07;                            01286000 T  0003:0020:1
                DUMPV            =08#,    COMMENT  10;                            01287000 T  0003:0020:1
                LISTV            =09#,    COMMENT  11;                            01288000 T  0003:0020:1
                OUTV             =10#,    COMMENT  12;                            01289000 T  0003:0020:1
                INV              =11#,    COMMENT  13;                            01290000 T  0003:0020:1
                MONITORV         =12#,    COMMENT  14;                            01291000 T  0003:0020:1
                SWITCHV          =13#,    COMMENT  15;                            01292000 T  0003:0020:1
                PROCV            =14#,    COMMENT  16;                            01293000 T  0003:0020:1
                ARRAYV           =15#,    COMMENT  17;                            01294000 T  0003:0020:1
                FORMATV          =16#,    COMMENT  20;                            01295000 T  0003:0020:1
                FILEV            =17#,    COMMENT  21;                            01296000 T  0003:0020:1
                STREAMV          =18#,    COMMENT  22;                            01297000 T  0003:0020:1
                DEFINEV          =19#,    COMMENT  23;                            01298000 T  0003:0020:1
                AUXMEMV          =20#,    COMMENT 24;                      %117-   01298500 P  0003:0020:1
                FIELDV           =21#;    COMMENT 25;                      %117-   01298600 C  0003:0020:1
            DEFINE ADES=0#,LDES=2#,PDES=1#,CHAR=3#;                               01299000 T  0003:0020:1
             REAL TIME1;                                                          01300000 T  0003:0020:1
PRT(121) = TIME1
             INTEGER SCRAM;                                                       01301000 T  0003:0020:1
PRT(122) = SCRAM
            COMMENT SCRAM CONTAINS THE SCRAMBLE INDEX FOR THE LAST IDENTIFIER     01302000 T  0003:0020:1
                    OR RESERVED WORD SCANNED;                                     01303000 T  0003:0020:1
          ARRAY FILEATTRIBUTES[0:30] ;                                            01303500 T  0003:0020:1
PRT(123) = FILEATTRIBUTES
           ALPHA ARRAY ACCUM[0:10];                                               01304000 T  0003:0023:2
PRT(124) = ACCUM
            COMMENT ACCUM HOLDS THE ALPHA AND CHARACTER COUNT OF THE LAST         01305000 T  0003:0025:3
                    SCANNED ITEM IN A FORM COMPATIBLE WITH ITS APPEARANCE         01306000 T  0003:0025:3
                    IN INFO. THAT IS ACCUM[1] = 00NAAAAA. ACCUM[I] , I> 1,        01307000 T  0003:0025:3
                    HAS ANY ADDITIONAL CHARACTERS. ACCUM[0] IS USED FOR           01308000 T  0003:0025:3
                    THE ELBAT WORD BY THE ENTER ROUTINES;                         01309000 T  0003:0025:3
          ARRAY STACKHEAD,SUPERSTACK[0:124];                          %WF         01310000 T  0003:0025:3
PRT(125) = STACKHEAD
PRT(126) = SUPERSTACK
            COMMENT STACKHEAD[N] CONTAINS AN INDEX INTO INFO.  THIS INDEX %WF     01311000 T  0003:0029:2
                    POINTS TO THE TOP ITEM IN THE N-TH STACK (ACTUALLY A  %WF     01311100 T  0003:0029:2
                    LINKED-LIST).  SUPERSTACK IS NOT A TELEVISION STAR,   %WF     01311200 T  0003:0029:2
                    BUT RATHER A SPECIAL STACKHEAD WHICH ALWAYS POINTS    %WF     01311300 T  0003:0029:2
                    AT CERTAIN COMMONLY USED RESERVED WORDS.  THOSE       %WF     01311400 T  0003:0029:2
                    WORDS POINTED TO (IN THREE GROUPS) ARE:               %WF     01311500 T  0003:0029:2
                    1)   ALPHA, LABEL, OWN, REAL, SAVE                    %WF     01311600 T  0003:0029:2
                    2)   AND, DIV, EQV, IMP, MOD, NOT, OR, TRUE           %WF     01311700 T  0003:0029:2
                    3)   BEGIN, DO, ELSE, END, FOR, GO, IF,               %WF     01311800 T  0003:0029:2
                           STEP, THEN, TO, UNTIL, WHILE, WRITE;           %WF     01311900 T  0003:0029:2
                    FOR MORE INFORMATION ON THE USE OF SUPERSTACK, SEE    %WF     01312000 T  0003:0029:2
                    COMMENTS IN THE TABLE PROCEDURE.                  ;   %WF     01312100 T  0003:0029:2
          INTEGER COUNT;                                                          01313000 T  0003:0029:2
PRT(127) = COUNT
```

```
                    COMMENT COUNT CONTAINS THE NUMBER OF CHARACTORS OF THE LAST ITEM          01314000 T  0003:002912
                            SCANNED;                                                          01315000 T  0003:002912
                ALPHA Q;                                                                      01316000 T  0003:002912
PRT(130) = Q
                    COMMENT Q CONTAINS ACCUM[1] FOR THE LAST IDENTIFIER OR RESERVED           01317000 T  0003:002912
                            WORD SCANNED;                                                     01318000 T  0003:002912
                ARRAY ELBAT[0:76]; INTEGER I,NXTELBT;                                         01319000 T  0003:002912
PRT(131) = ELBAT
PRT(132) = I
PRT(133) = NXTELBT
                    COMMENT ELBAT IS AN ARRAY HOLDING ELBAT WORDS FOR RECENTLY SCANNED        01320000 T  0003:003113
                            QUANTITIES.  THE TABLE ROUTINE MAINTAINS THIS ARRAY.              01321000 T  0003:003113
                            (ELBAT IS TABLE SPELLED BACKWARDS.)  THE TABLE ROUTINE            01322000 T  0003:003113
                            GUARANTIES THAT ELBAT ALWAYS CONTAINS THE ELBAT WORDS             01323000 T  0003:003113
                            FOR THE LAST 10 QUANTITIES SCANNED.  NXTELBT IS AN INDEX          01324000 T  0003:003113
                            POINTING TO THE NEXT AVAILABLE WORD IN ELBAT.  I IS AN            01325000 T  0003:003113
                            INDEX USED BY THE REST OF THE COMPILER TO FETCH THINGS            01326000 T  0003:003113
                            FROM ELBAT.  I IS ALSO MAINTAINED BY THE TABLE ROUTINE;           01327000 T  0003:003113
                INTEGER ELCLASS;                                                              01328000 T  0003:003113
PRT(134) = ELCLASS
                    COMMENT ELCLASS USUALLY CONTAINS ELBAT[I],CLASS;                          01329000 T  0003:003113
                INTEGER LASTELCLASS;                                                          01329100 T  0003:003113
PRT(135) = LASTELCLASS
                    COMMENT LASTELCLASS IS SET TO PREV ELCLASS BY NEXTENT;                    01329200 T  0003:003113
                INTEGER FCR, NCR, LCR,TLCR,CLCR;                                              01330000 T  0003:003113
PRT(136) = FCR
PRT(137) = NCR
PRT(140) = LCR
PRT(141) = TLCR
PRT(142) = CLCR
                INTEGER MAXTLCR;                                                              01331000 T  0003:003113
PRT(143) = MAXTLCR
                    COMMENT FCR CONTAINS ABSOLUTE ADDRESS OF THE FIRST CHARACTOR OF           01332000 T  0003:003113
                            THE CARD IMAGE CURRENTLY BEING SCANNED, NCR THE ADDRESS           01333000 T  0003:003113
                            OF THE NEXT CHARACTOR TO BE SCANNED, AND LCR THE LAST             01334000 T  0003:003113
                            CHARACTOR (COLUMN 73). TLCR AND CLCR CONTAIN ADDRESS OF           01335000 T  0003:003113
                            THE LAST CHARACTER IN THE TAPE AND CARD BUFFERS.  MAXTLCR          01336000 T  0003:003113
                            IS THE MAXIMUM OF TLCR WHEN THE INPUT IS BLOCKED;                 01337000 T  0003:003113
                DEFINE BUFFSIZE =  56#;                                                       01338000 T  0003:003113
                                                                                              01339000 T  0003:003113
                        INTEGER GTIX;                                                         01339050 T  0003:003113
PRT(144) = GTIX
                ARRAY TEN[0:69];                                                              01340000 T  0003:003113
PRT(145) = TEN
                    INTEGER  NOOFARRAYS; COMMENT NOOFARRAYS IS THE SUM OF ARRAYS              01340050 T  0003:003410
PRT(146) = NOOFARRAYS
                                DECLARED IN THE OBJECT PROGRAM;                               01340060 T  0003:003410
                    INTEGER IOBUFFSIZE; COMMENT IOBUFFSIZE IS FILE SPACE NEEDED,              01340070 T  0003:003410
PRT(147) = IOBUFFSIZE
                                        GTI1 EQUALS TOTAL CORE STORAGE REQD;                  01340080 T  0003:003410
                REAL FSAVE; COMMENT SAVES FRACTIONAL PART EXPONENT WHEN CONV NUM;              01340500 T  0003:003410
PRT(150) = FSAVE
                    INTEGER IDLOC,IDLOCTEMP;                                                  01341000 T  0003:003410
PRT(151) = IDLOC
PRT(152) = IDLOCTEMP
                ARRAY PDPRT[0:31,0:63];                                                       01342000 T  0003:003410
PRT(153) = PDPRT
```

```
COMMENT PDPRT CONTAINS INFORMATION FOR USE AT THE END OF COMPILATION
        IT IS BUILT BY PROGDESCBLDR.THIS INFORMATION IS USED TO
        BUILD THE SEGMENT DICTIONARY AND PRT. THERE ARE TWO TYPES
        OF ENTRIES IN THIS TABLE AS DESCRIBED BELOW.
                TYPE 1 ENTRY
           BIT POSITION          KIND OF ENTRY
             0-3                   ZERO
               4                   MODE BIT(1=CHAR  0=WORD)
               5                   ARGUMENT BIT
             6-7                   ZERO
            8-17                   RELATIVE ADDRESS IN PRT
           18-27                   RELATIVE ADDRESS IN SEGMENT
           28-37                   SEGMENT NUMBER
           38-47                   ZERO
                TYPE 2 ENTRY
           BIT POSITION          KIND OF ENTRY
               0                   EMPTY
               1                   ON IFF TYPE 2 (DATA) SEGMENT
               2                   ON IFF INTRINSIC PROCEDURE
               3                   ON IFF "PSEUDO-SAVE" SEGMENT
             4-12                  EMPTY
           13-27                   DISK ADDRESS OR INTRINSIC NUMBER
           28-37                   SEGMENT NUMBER
           38-47                   NUMBER OF WORDS IN SEGMENT
        THERE IS ONLY ONE TYPE 2 ENTRY PER SEGMENT.THE TYPE 2 ENTRY
        IS DISTINGUISHED BY THE NON ZERO FIELD IN BITS 38-47. THIS
        ENTRY IS USED TO BUILD THE DRUM DESCRIPTOR IN THE SEGMENT
        DICTIONARY.TYPE 2 ENTRIES ARE PUT INTO PDPRT WHEN ANY SEGMENT
        IS READY FOR OUTPUT;                                          01343000 T 0003:0036:1
                                                                      01344000 T 0003:0036:1
                                                                      01345000 T 0003:0036:1
                                                                      01346000 T 0003:0036:1
                                                                      01347000 T 0003:0036:1
                                                                      01348000 T 0003:0036:1
                                                                      01349000 T 0003:0036:1
                                                                      01350000 T 0003:0036:1
                                                                      01351000 T 0003:0036:1
                                                                      01352000 T 0003:0036:1
                                                                      01353000 T 0003:0036:1
                                                                      01354000 T 0003:0036:1
                                                                      01355000 T 0003:0036:1
                                                                      01356000 T 0003:0036:1
                                                                      01357000 T 0003:0036:1
                                                                      01358000 T 0003:0036:1
                                                                      01359000 T 0003:0036:1
                                                                      01360000 T 0003:0036:1
                                                                      01361000 T 0003:0036:1
                                                                      01361050 T 0003:0036:1
                                                                      01361100 T 0003:0036:1
                                                                      01361200 T 0003:0036:1
                                                                      01361300 T 0003:0036:1
                                                                      01362000 T 0003:0036:1
                                                                      01363000 T 0003:0036:1
                                                                      01364000 T 0003:0036:1
                                                                      01365000 T 0003:0036:1
                                                                      01366000 T 0003:0036:1
                                                                      01367000 T 0003:0036:1
COMMENT THE FORMAT OF SEGMENT DICTIONARY AND PRT ENTRIES AT THE END OF
        COMPILATION IS AS FOLLOWS;                                    01367010 T 0003:0036:1
                                                                      01367020 T 0003:0036:1
             SEGMENT DICTIONARY ENTRY (IE., SD[I] FOR SEGMENT NUM. I) 01367030 T 0003:0036:1
        BIT POSITIONS  CONTENTS OF FIELD                              01367040 T 0003:0036:1
            [0:1]        EMPTY                                        01367050 T 0003:0036:1
            [1:1]        ON IFF TYPE 2 (DATA) SEGMENT                 01367060 T 0003:0036:1
            [2:1]        ON IFF INTRINSIC PROCEDURE                   01367070 T 0003:0036:1
            [3:1]        EMPTY (USED BY MCP PRESENCE-BIT ROUTINE)     01367075 T 0003:0036:1
            [4:1]        ON IFF "PSEUDO-SAVE" SEGMENT                 01367080 T 0003:0036:1
            [5:1]        EMPTY (USED BY MCP OVERLAY ROUTINE)          01367085 T 0003:0036:1
            [8:10]       R-RELATIVE LINK TO A PRT ENTRY FOR THIS SEGMENT 01367090 T 0003:0036:1
           [18:15]       SIZE (NOT USED FOR INTRINSICS)               01367100 T 0003:0036:1
           [33:15]       DISK ADDRESS OR INTRINSIC NUMBER             01367110 T 0003:0036:1
              PRT ENTRY (IE., PROGRAM DESCRIPTOR FOR SEGMENT NUMBER I) 01367120 T 0003:0036:1
        BIT POSITIONS  CONTENTS OF FIELD                              01367130 T 0003:0036:1
            [0:4]        1101(BINARY) NON-PRESENT PROG. DESC. ID BITS 01367140 T 0003:0036:1
            [4:2]        MODE AND ARGUMENT BITS                       01367150 T 0003:0036:1
            [6:1]        STOPPER (ON IFF THIS ENTRY LINKS TO SEG. DICT.) 01367160 T 0003:0036:1
            [7:11]       IF [6:1] THEN I ELSE R-RELATIVE LINK TO ANOTHER 01367170 T 0003:0036:1
                                          PRT ENTRY FOR SEGMENT I     01367180 T 0003:0036:1
           [18:15]       I                                            01367190 T 0003:0036:1
           [33:15]     RELATIVE ADDRESS WITHIN THE SEGMENT OF THIS DESC; 01367200 T 0003:0036:1
COMMENT THE CONTENTS OF RELATIVE DISK SEGMENT ZERO OF THE CODE FILE ARE; 01367210 T 0003:0036:1
        WORD        CONTENTS                                          01367220 T 0003:0036:1
          0         RELATIVE LOCATION OF SEGMENT DICTIONARY           01367230 T 0003:0036:1
          1         SIZE OF SEGMENT DICTIONARY                        01367240 T 0003:0036:1
          2         RELATIVE LOCATION OF PRT                          01367250 T 0003:0036:1
          3         SIZE OF PRT                                       01367260 T 0003:0036:1
```

```
                        4       RELATIVE LOCATION OF FILE PARAMETER BLOCK                                01367270 T  0003:003611
                        5       SIZE OF FILE PARAMETER BLOCK                                             01367280 T  0003:003611
                        6       SEGMENT NUMBER OF FIRST SEGMENT TO EXECUTE (IE., 1)                      01367290 T  0003:003611
                        7       N                                                                        01367300 T  0003:003611
                        .        O    U                                                                  01367310 T  0003:003611
                        .         T     S                                                               01367320 T  0003:003611
                        .                E                                                            01367330 T  0003:003611
                       29                  D;                                                           01367340 T  0003:003611
            INTEGER PDINX;COMMENT THIS IS THE INDEX FOR PDPRT;                                          01368000 T  0003:003611
PRT(154) = PDINX
            INTEGER SGAVL;COMMENT NEXT AVAILABLE SEGMENT NUMBER;                                        01369000 T  0003:003611
PRT(155) = SGAVL
            INTEGER SGNO;COMMENT THIS IS THE CURRENT SEGMENT NUMBER;                                    01370000 T  0003:003611
PRT(156) = SGNO
             ARRAY EDOC[0:7,0:127],COP[0:63],WOP[0:127],POP[0:10];                                      01371000 T  0003:003611
PRT(157) = EDOC
PRT(160) = COP
PRT(161) = WOP
PRT(162) = POP
                COMMENT  THE EMIT ROUTINES PLACE EACH SYLLABLE INTO THE EDOC ARRAY                      01372000 T  0003:004712
                         AS SPECIFIED BY "L",                                                           01373000 T  0003:004712
                         IF DEBUGTOG IS TRUE, COP, WOP, AND POP ARE FILLED                              01374000 T  0003:004712
                         THE BCD FOR THE OPERATORS,OTHERWISE THEY ARE NOT USED;                         01375000 T  0003:004712
             REAL  LASTENTRY ;                                                                          01376000 T  0003:004712
PRT(163) = LASTENTRY
                COMMENT LASTENTRY IS USED BY EMITNUM AND CONSTANTCLEAN, IT POINTS                       01377000 T  0003:004712
                         INTO INFO[0,*] AT THE NEXT AVAILABLE CELL FOR CONSTANTS;                       01378000 T  0003:004712
             BOOLEAN MRCLEAN ;                                                                          01379000 T  0003:004712
PRT(164) = MRCLEAN
                COMMENT  NO  CONSTANTCLEAN ACTION TAKES PLACE WHILE MRCLEAN IS                          01380000 T  0003:004712
                         FALSE. THIS FEATURE IS USED BY BLOCK BECAUSE OF THE                            01381000 T  0003:004712
                         POSSIBILITY THAT CONSTANTCLEAN WILL USE INFO[NEXTINFO]                         01382000 T  0003:004712
                         DURING AN ARRAY DECLARATION ;                                                  01383000 T  0003:004712
            REAL GT1,GT2,GT3,GT4,GT5;                                                                   01384000 T  0003:004712
PRT(165) = GT1
PRT(166) = GT2
PRT(167) = GT3
PRT(170) = GT4
PRT(171) = GT5
            INTEGER GTI1;                                                                               01384500 T  0003:004712
PRT(172) = GTI1
                COMMENT THESE VARIABLES ARE USED FOR TEMPORARY STORAGE;                                 01385000 T  0003:004712
            INTEGER RESULT;                                                                             01386000 T  0003:004712
PRT(173) = RESULT
                COMMENT THIS VARIABLE IS USED FOR A DUAL PURPOSE BY THE TABLE                           01387000 T  0003:004712
                         ROUTINE AND THE SCANNER,  THE TABLE ROUTINE USES THIS                          01388000 T  0003:004712
                         VARIABLE TO SPECIFY SCANNER OPERATIONS AND THE SCANNER                         01389000 T  0003:004712
                         USES IT TO INFORM THE TABLE ROUTINE OF THE ACTION TAKEN;                       01390000 T  0003:004712
            INTEGER LASTUSED;                                                                           01391000 T  0003:004712
PRT(174) = LASTUSED
                COMMENT LASTUSED IS A VARIABLE THAT CONTROLS THE ACTION OF                              01392000 T  0003:004712
                         READACARD, THE ROUTINE WHICH READS CARDS AND INITIALIZES                       01393000 T  0003:004712
                         OR PREPARES THE CARD FOR THE SCANNER.                                          01394000 T  0003:004712
                     LASTUSED  LAST CARD READ FROM                                                      01394500 T  0003:004712
                     --------  ------------------                                                       01394600 T  0003:004712
                         1     CARD READER ONLY, NO TAPE.                                             01395000 T  0003:004712
                         2     CARD READER, TAPE AND CARD MERGE.                                      01396000 T  0003:004712
```

```
                                3     TAPE, TAPE AND CARD MERGE.                                    01397000 T  0003:004712
                                4     INITIALIZATION ONLY, CARD ONLY.                               01398000 T  0003:004712
                                5     CARD READER - MAKCAST, MERGING.                               01398100 T  0003:004712
                                6     TAPE - MAKCAST, MERGING.                                      01398200 T  0003:004712
                                ;                                                                   01398300 T  0003:004712
                        BOOLEAN LINKTOG;                                                            01399000 T  0003:004712
PRT(175) = LINKTOG
                    COMMENT LINKTOG IS FALSE IF THE LAST THING EMITTED IS A LINK,                   01400000 T  0003:004712
                            OTHERWISE IT IS TRUE;                                                   01401000 T  0003:004712
                        INTEGER LEVEL,FRSTLEVEL,SUBLEVEL,MODE;                                      01402000 T  0003:004712
PRT(176) = LEVEL
PRT(177) = FRSTLEVEL
PRT(200) = SUBLEVEL
PRT(201) = MODE
                    COMMENT THESE VARIABLES ARE MAINTAINED BY THE BLOCK ROUTINE TO KEEP             01403000 T  0003:004712
                            TRACK OF LEVELS OF DEFINITION.  LEVEL GIVES THE DEPTH OF                01404000 T  0003:004712
                            NESTING IN DEFINITION, WHERE EACH BLOCK AND EACH PROCEDURE              01405000 T  0003:004712
                            GIVES RISE TO A NEW LEVEL.  SUBLEVEL GIVES THE LEVEL OF                 01406000 T  0003:004712
                            THE PARAMETERS OF THE PROCEDURE CURRENTLY BEING COMPILED.               01407000 T  0003:004712
                            FRSTLEVEL IS THE LEVEL OF THE PARAMETERS OF THE MOST                    01408000 T  0003:004712
                            GLOBAL OF THE PROCEDURES CURRENTLY BEING COMPILED.  MODE                01409000 T  0003:004712
                            IS THE CURRENT DEPTH OF THE PROCEDURE IN WHICH WE ARE                   01410000 T  0003:004712
                            NESTED (AT COMPILE TIME);                                               01411000 T  0003:004712
                        INTEGER AUXMEMREQ;                                                          01411010 T  0003:004712
PRT(202) = AUXMEMREQ
                        BOOLEAN SAVEPRTOG;                                                          01411020 T  0003:004712
PRT(203) = SAVEPRTOG
                    COMMENT VARIABLES USED TO CONTROL SEGMENT DICTIONARY                            01411030 T  0003:004712
                            ENTRIES FOR "PSEUDO-SAVE" PROCEDURES,                                   01411040 T  0003:004712
                            AUXMEMREQ IS THE AMOUNT OF AUXILIARY MEMORY                             01411050 T  0003:004712
                            WHICH WOULD BE REQUIRED IF ALL OF THESE                                 01411060 T  0003:004712
                            "PSEUDO-SAVE" ROUTINES ARE TO BE OVERLAID                               01411070 T  0003:004712
                            TO AUXILIARY MEMORY.  SAVEPRTOG IS USED                                 01411080 T  0003:004712
                            TO COMMUNICATE TO THE OUTSIDE WORLD THAT A                              01411090 T  0003:004712
                            ROUTINE IS "PSEUDO-SAVE".                                               01411100 T  0003:004712
                            ;                                                                       01411110 T  0003:004712
                    BOOLEAN ERRORTOG;                                                               01412000 T  0003:004712
PRT(204) = ERRORTOG
                COMMENT ERRORTOG IS TRUE IF MESSAGES ARE CURRENTLY ACCEPTABLE TO THE                01413000 T  0003:004712
                        ERROR ROUTINES.  ERRORCOUNT IS THE COUNT OF ERROR MSSGS;                    01414000 T  0003:004712
                BOOLEAN ENDTOG;     COMMENT ENDTOG TELLS THE TABLE TO ALLOW                         01415000 T  0003:004712
PRT(205) = ENDTOG
                                    COMMENT TO BE PASSED BACK TO COMPOUNDTAIL;                      01416000 T  0003:004712
                BOOLEAN STREAMTOG; % STREAMTOG IS TRUE IF WE ARE COMPILING A                        01416500 T  0003:004712
PRT(206) = STREAMTOG
                                    % STREAM STATEMENT IN ALGOL, TSPOL, OR ESPOL;                   01417000 T  0003:004712
                                    % IT IS USED TO CONTROL COUMPOUNDTAIL.                         01417500 T  0003:004712
                                    % IT IS ALSO USED WHEN WE ARE COMPILING A                      01418000 T  0003:004712
                                    % "FILL" STATEMENT (SEE "FILLSTMT" PROCEDURE) OR               01418500 T  0003:004712
                                    % AN ALPHA (BCL) STRING (SEE "TABLE" PROCEDURE).               01419000 T  0003:004712
                    DEFINE FS = 1#, FP = 2#, FL = 3#, FR = 4#, FA = 5#,                             01420000 T  0003:004712
                           FI = 6#, FIO = 7#;                                                       01420500 T  0003:004712
                        COMMENT THESE DEFINES ARE USED WHEN CALLING THE VARIABLE ROUTINE.           01421000 T  0003:004712
                                THEIR PURPOSES IS TO TELL VARIABLE WHO IS CALLING.                  01422000 T  0003:004712
                                THEIR MEANING IS;                                                   01423000 T  0003:004712
                                FS MEANS FROM STATEMENT,                                            01424000 T  0003:004712
                                FP MEANS FROM PRIMARY,                                              01425000 T  0003:004712
```

```
                        FL MEANS FROM LIST,                                                    01426000 T  0003:004712
                        FR MEANS FROM FOR,                                                     01427000 T  0003:004712
                        FIO MEANS FROM IODEC,                                                  01427250 T  0003:004712
                        FA MEANS FROM ACTUALPARAPART,                                          01427500 T  0003:004712
                        FI MEANS FUNNY CALL FROM STATUS (IMPFUN);                              01427600 T  0003:004712
                  INTEGER L;                                                                   01428000 T  0003:004712
PRT(207) = L
                     COMMENT   L IS THE LOCATION OF THE NEXT SYLLABLE TO BE EMITTED;           01429000 T  0003:004712
DEFINE BLOCKCTR = 16#, JUNK = 17 #, XITR = 18 #, LSTRTN = 19#;                                 01430000 T  0003:004712
                     COMMENT   THESE DEFINES NAME THE FIXED PRT CELLS USED BY ALL OBJECT       01431000 T  0003:004712
                        PROGRAMS.                                                              01432000 T  0003:004712
                             BLOCKCTR IS A TALLY WHICH IS INCREMENT EACH TIME A                01433000 T  0003:004712
                        BLOCK IS ENTERED WHICH OBTAINS STORAGE, OR CONTAINS WITH               01434000 T  0003:004712
                        IN IT A NON-LOCAL GO TO.  EACH TIME SUCH A BLOCK IS LEFT               01435000 T  0003:004712
                        BLOCKCTR IS DECREMENTED.  THE PRIMARY PURPOSE SERVED IS TO             01436000 T  0003:004712
                        INFORM THE MCP OF THE STORAGE WHICH NEEDS TO BE RETURNED.              01437000 T  0003:004712
                             JUNK IS AN ALL-PURPOSE CELL FOR STORING VALUES USED               01438000 T  0003:004712
                        IN LINKAGE BETWEEN VARIOUS ROUTINES AND FOR INTEGERIZING               01439000 T  0003:004712
                        THINGS ON THE TOP OF THE STACK.                                        01440000 T  0003:004712
                             XITR CONTAINS A CHARACTOR MODE PROGRAM DESCRIPTOR                 01441000 T  0003:004712
                        WHICH POINTS AT AN EXIT CHARACTOR MODE OPERATOR.  IT IS                01442000 T  0003:004712
                        USED TO CLEAN UP THE STACK AFTER A MKS HAS BEEN GIVEN.                 01443000 T  0003:004712
                        THIS A USFULL WAY TO ELIMINATE MANY REDUNDENT ITEMS IN THE             01444000 T  0003:004712
                        STACK.  SEE FOR EXAMPLE THE ARRAY DECLARATIONS.                        01445000 T  0003:004712
                             LSTRTN IS A CELL USED AS LINKAGE BETWEEN A LIST AND               01446000 T  0003:004712
                        THE I-O FORMATING ROUTINES.  THE FIRST SYLLABLES EXECUTED              01447000 T  0003:004712
                        BY A LIST ARE: 1) OPDC LSTRTN, 2) BFW.  THIS CARRIES YOU               01448000 T  0003:004712
                        TO THE PROPER ITEM IN THE LIST.  THE FORMATING ROUTINES                01449000 T  0003:004712
                        SET LSTRTN INITIALLY TO ZERO.  THE LIST ITSELF UPDATES                 01450000 T  0003:004712
                        LSTRTN.  THE LIST EXHAUSTED FLAG IS =1;                                01451000 T  0003:004712
                  DEFINE BTYPE =1#, DTYPE =2#, ATYPE =3#;                                      01452000 T  0003:004712
                     COMMENT   THESE DEFINES NAME THE VALUES USED BY THE EXPRESSION            01453000 T  0003:004712
                        ROUTINES IF REPORT THE TYPE OF EXPRESSION COMPILED.                    01454000 T  0003:004712
                        BTYPE IS FOR BOOLEAN, DTYPE FOR DESIGNATIONAL, AND ATYPE               01455000 T  0003:004712
                        FOR ARITHMETIC EXPRESSIONS;                                            01456000 T  0003:004712
                  BOOLEAN TB1;                                                                 01457000 T  0003:004712
PRT(210) = TB1
                     COMMENT TB1 IS A TEMPORARY BOOLEAN VARIABLE;                              01458000 T  0003:004712
                  INTEGER JUMPCTR;                                                             01459000 T  0003:004712
PRT(211) = JUMPCTR
                     COMMENT   JUMPCTR IS A VARIABLE USED FOR COMMUNICATION BETWEEN BLOCK      01460000 T  0003:004712
                        AND GENGO.  IT GIVES HIGHEST LEVEL TO WHICH A JUMP HAS                 01461000 T  0003:004712
                        BEEN MADE FROM WITHIN A THE PRESENTLY BEING COMPILED                   01462000 T  0003:004712
                        SEGMENT.  THE BLOCK COMPILES CODE TO INCREMENT AND DECRE-              01463000 T  0003:004712
                        MENT THE BLOCKCTR ON THE BASIS OF JUMPCTR AT COMPLETION                01464000 T  0003:004712
                        OF COMPILATION OF A SEGMENT - I.E. THE BLOCKCTR IS TALLIED             01465000 T  0003:004712
                        IF LEVEL = JUMPCTR;                                                    01466000 T  0003:004712
                  BOOLEAN GOTOG;                                                               01467000 T  0003:004712
PRT(212) = GOTOG
                     COMMENT GOTOG IS SET FALSE  BY GOSTMT.   DEXP SETS GOTOG TRUE IF ANY      01468000 T  0003:004712
                             LABEL OR SWITCH IS NON LOCAL.  GOSTMT FINDS OUT BY THIS           01469000 T  0003:004712
                             MEANS WHETHER OR NOT A CALL ON MCP IS NECESSARY;                  01470000 T  0003:004712
                  REAL STLB;                                                                   01471000 T  0003:004712
PRT(213) = STLB
                     COMMENT STLB IS USED BY VARIABLE AND ACTUALPARAPART TO COMMUNICATE        01472000 T  0003:004712
                             THE LOWER BOUND INFORMATION FOR THE LAST DIMENSION OF THE         01473000 T  0003:004712
                             ARRAY INVOLVED IN A ROW DESIGNATOR.  THE FORMAT OF THE            01474000 T  0003:004712
```

```
                        INFORMATION IS THAT OF INFO,  STLB IS ALSO SOMETIMES USED          01475000 T  0003:004?:2
                        FOR TEMPORARY STORAGE;                                             01476000 T  0003:004?:2
                DEFINE BUMPL = L+L+2#;                                                     01477000 T  0003:004?:2
                   COMMENT BUMPL IS USED MOSTLY TO PREPARE A FORWARD JUMP;                 01478000 T  0003:004?:2
                DEFINE IDMAX =   LABELID#;                                                 01479000 T  0003:004?:2
                   COMMENT IDMAX IS THE MAXIMUM CLASS NUMBER FOR IDENTIFIERS;              01480000 T  0003:004?:2
                INTEGER DEFINECTR,DEFINEINDEX;                                             01481000 T  0003:004?:2
PRT(214) = DEFINECTR
PRT(215) = DEFINEINDEX
                ALPHA ARRAY DEFINFO[0:89];                                                 01481100 T  0003:004?:2
PRT(216) = DEFINFO
                ALPHA ARRAY TEXT[0:31,0:255];                                              01481200 T  0003:004?:3
PRT(217) = TEXT
                INTEGER DEFSTACKHEAD;  % STACKHEAD FOR DEFINE PARAMETERS                  01481300 T  0003:005?:0
PRT(220) = DEFSTACKHEAD
                INTEGER NEXTTEXT;      % NEDEX OF NEXT DEFINE TEXT                       01481400 T  0003:005?:0
PRT(221) = NEXTTEXT
                   REAL  JOINFO,       COMMENT POINTS TO PSEUDO LABEL FOR JUMP OUTS;    01482000 T  0003:005?:0
PRT(222) = JOINFO
                        LPRT, COMMENT SHOWS LOCATION OF THE LAST LABEL IN THE PRT ;        01483000 T  0003:005?:0
PRT(223) = LPRT
                     NESTLEVEL,        COMMENT  COUNTS NESTING FOR GO TO AND JUMP OUTS;  01484000 T  0003:005?:0
PRT(224) = NESTLEVEL
                      JUMPLEVEL;       COMMENT NUMBER OF LEVELS TO BE JUMPED OUT;        01485000 T  0003:005?:0
PRT(225) = JUMPLEVEL
                   COMMENT THE REALS ABOVE ARE FOR STREAM STATEMENT;                       01486000 T  0003:005?:0
                   ARRAY  MACRO[0:35];                                                     01487000 T  0003:005?:0
PRT(226) = MACRO
                   COMMENT MACRO IS FILLED WITH SYLLABLES FOR STREAM STATEMENT;            01488000 T  0003:005?:1
                REAL  P,     COMMENT CONTAINS NUMBER OF FORMALS FOR STREAM PROCS;        01489000 T  0003:005?:1
PRT(227) = P
                      Z;      COMMENT CONTAINS 1ST WORD OF INFO FOR STREAM FUNCTIONS;   01490000 T  0003:005?:1
PRT(230) = Z
             SAVE ALPHA ARRAY DEFINEARRAY[0:34];                                       01491000 T  0003:005?:1
PRT(231) = DEFINEARRAY
                   COMMENT   THESE VARIABLES ARE USED TO CONTROL ACTION OF THE DEFINE,  01492000 T  0003:005?:2
                             DEFINECTR COUNTS DEPTH OF NESTING OF DEFINE-# PAIRS,      01493000 T  0003:005?:2
                             THE CROSSHATCH PART OF THE TABLE ROUTINE USES DEFINECTR    01494000 T  0003:005?:2
                             TO DETERMINE THE MEANING OF A CROSSHATCH,  DEFINEINDEX IS 01495000 T  0003:005?:2
                             THE NEXT AVAILABLE CELL IN THE DEFINEARRAY,  THE DEFINE-  01496000 T  0003:005?:2
                             ARRAY HOLDS THE ALPHA OF THE DEFINE BEING RECREATED AND   01497000 T  0003:005?:2
                             THE PREVIOUS VALUES OF LASTUSED, LCR, AND NCR;            01498000 T  0003:005?:2
                INTEGER BEGINCTR;                                                      01499000 T  0003:005?:2
PRT(232) = BEGINCTR
                   COMMENT   BEGINCTR GIVES THE NUMBER OF UNMATCHED BEGINS,  IT IS USED 01500000 T  0003:005?:2
                             FOR ERROR CONTROL ONLY;                                   01501000 T  0003:005?:2
                INTEGER DIALA,DIALB;                                                   01502000 T  0003:005?:2
PRT(233) = DIALA
PRT(234) = DIALB
                   COMMENT THESE VARIABLES GIVE THE LAST VALUE TO WHICH A AND B WERE   01503000 T  0003:005?:2
                           DIALED,  THIS GIVES SOME LOCAL OPTIMIZATION,  EMITD         01504000 T  0003:005?:2
                           WORRIES ABOUT THIS,  OTHER ROUTINES CAUSE A LOSS OF MEMORY  01505000 T  0003:005?:2
                           BY SETTING DIALA AND DIALB TO ZERO;                         01506000 T  0003:005?:2
                                                                                       01507000 T  0003:005?:2
                                                                                       01508000 T  0003:005?:2
                                                                                       01509000 T  0003:005?:2
                                                                                       01510000 T  0003:005?:2
```

```
                BOOLEAN RRB1;  COMMENT RRB1---RRBN ARE BOOLEAN VARIABLES THAT SERVE THE                01522000 T  0003:0057:2
PRT(235) = RRB1
                        SAME FUNCTION AS RR1---RRN FOR REAL VARIABLES.  SEE                            01523000 T  0003:0057:2
                        COMMENT AT RR1;                                                                01524000 T  0003:0057:2
                    BOOLEAN RRB2;  COMMENT SEE COMMENT AT RRB1 DECLARATION;                            01525000 T  0003:0057:2
PRT(236) = RRB2
                DEFINE ARRAYMONFILE = [27:11]#;  COMMENT ARRAYMONFILE IS THE DEFINE FOR                01526000 T  0003:0057:2
                                        THE ADDRESS OF THE FILE DESCRIPTOR IN                          01527000 T  0003:0057:2
                                        THE FIRST WORD OF ADDITIONAL INFO;                             01528000 T  0003:0057:2
                DEFINE SVARMONFILE = [37:11]#;  COMMENT MONITORFILE IS THE DEFINE FOR                  01529000 T  0003:0057:2
                                       THE ADDRESS OF THE FILE DESCRIPTOR IN                           01530000 T  0003:0057:2
                                       INFO FOR MONITORED SIMPLE VARIABLES;                            01531000 T  0003:0057:2
                DEFINE NODIMPART = [40:8]#;  COMMENT THE FIRST ADDITIONAL WORD OF INFO                  01532000 T  0003:0057:2
                                      FOR ARRAYS CONTAINS THE NUMBER OF DIMENSIONS                     01533000 T  0003:0057:2
                                      IN NODIMPART;                                                    01534000 T  0003:0057:2
                DEFINE LABLMONFILE = [13:11]#;  COMMENT LABLMONFILE DESIGNATES THE BIT                 01535000 T  0003:0057:2
                                       POSITION IN THE FIRST WORD OF ADDITIONAL                        01536000 T  0003:0057:2
                                       INFO THAT CONTAINS THE MONITOR FILE                             01537000 T  0003:0057:2
                                       ADDRESS FOR LABELS;                                             01538000 T  0003:0057:2
                DEFINE SWITMONFILE = [13:11]#;  COMMENT SWITMONFILE DESIGNATES THE BIT                 01539000 T  0003:0057:2
                                       POSITION IN THE FIRST WORD OF ADDITIONAL                        01540000 T  0003:0057:2
                                       INFO THAT CONTAINS THE MONITOR FILE                             01541000 T  0003:0057:2
                                       ADDRESS FOR LABELS;                                             01542000 T  0003:0057:2
                DEFINE FUNCMONFILE = [27:11]#;  COMMENT FUNCMONFILE DESIGNATES THE BIT                 01543000 T  0003:0057:2
                                       POSITION IN THE FIRST WORD OF ADDITIONAL                        01544000 T  0003:0057:2
                                       INFO THAT CONTAINS THE MONITOR FILE                             01545000 T  0003:0057:2
                                       ADDRESS FOR LABELS;                                             01546000 T  0003:0057:2
                DEFINE DUMPEE = [2:11]#;  COMMENT THE DUMPEE FIELD IN THE FIRST                        01547000 T  0003:0057:2
                                     ADDITIONAL WORD OF INFO FOR LABELS CONTAINS                       01548000 T  0003:0057:2
                                     THE ADDRESS OF THE COUNTER THAT IS INCREMENTED                    01549000 T  0003:0057:2
                                     EACH TIME THE LABEL IS PASSED IF THAT LABEL                       01550000 T  0003:0057:2
                                     APPEARS IN A DUMP DECLARATION;                                    01551000 T  0003:0057:2
                DEFINE DUMPOR = [24:11]#;  COMMENT THE DUMPOR FIELD IN THE FIRST                       01552000 T  0003:0057:2
                                     ADDITIONAL WORD OF INFO FOR LABELS CONTAINS                       01553000 T  0003:0057:2
                                     THE ADDRESS OF THE ROUTINE THAT IS GENERATED                      01554000 T  0003:0057:2
                                     FROM THE DUMP DECLARATION THAT IN TURN CALLS                      01555000 T  0003:0057:2
                                     THE PRINTI ROUTINE;                                               01556000 T  0003:0057:2
                 DEFINE CHUNK = 180#;                                                                  01556100 T  0003:0057:2
                 FILE OUT CODF DISK[20:CHUNK](4,30,SAVE ABS(SAVETIME));                                01556200 T  0003:0057:2
PRT(237) = CODE
PRT(240) =
PRT(241) = FILE ATTRBUTS
                 FILE IN CARD (RR1,10,RR2);                                                            01557000 T  0003:0064:1
PRT(242) = CARD
                 SAVE                                                                                  01558000 T  0003:0068:1
                 FILE OUT LINE DISK SERIAL [20:2400] (RR3,15,RR4,SAVE 10);                             01559000 T  0003:0068:1
```

```
PRT(243) = LINE
                ARRAY LIN[0:20]; COMMENT PRINT OUTPUT BUILT IN LIN;                    01559010 T  0003:0075:3
PRT(244) = LIN
              INTEGER DA;                                                              01559020 T  0003:0078:0
PRT(245) = DA
               SAVE FILE OUT NEWTAPE DISK SERIAL [20:2400] (RR5,RR6,RR7,SAVE 1);       01560000 T  0003:0078:0
PRT(246) = NEWTAPE
               FILE IN TAPE "OCRDIMG" (2,RR8,RR9);                                    01561000 T  0003:0085:2
PRT(247) = TAPE
               SAVE FILE OUT PNCH DISK SERIAL [20:2400](2,10,RR10,SAVE 1);            01561005 T  0003:0089:3
PRT(250) = PNCH
                COMMENT THE FOLLOWING ARE DECLARATIONS FOR THE SYMBOLIC LIBRARIES;     01561010 T  0003:0096:1
                  FILE IN CASTA DISK SERIAL "CASTA" "LIBRARY"(1,BUFFSIZE);             01561020 T  0003:0096:1
PRT(251) = CASTA
                  FILE IN CASTB(1,BUFFSIZE);                                           01561030 T  0003:0100:1
PRT(252) = CASTB
                  FILE IN CASTC(1,BUFFSIZE);                                           01561040 T  0003:0105:2
PRT(253) = CASTC
                       SWITCH FILE LIBRARY=CASTA,CASTB,CASTC;                          01561050 T  0003:0109:2
PRT(254) = LIBRARY
                  FILE OUT REMOTE 19 (2,10);                                           01561055 T  0003:0115:3
PRT(255) = REMOTE
               SAVE ARRAY CBUFF,TBUFF[0:9]; % INPUT BUFFERS.                          01561056 T  0003:0119:3
PRT(256) = CBUFF
PRT(257) = TBUFF
               BOOLEAN REMOTOG;                                                       01561060 T  0003:0123:2
PRT(260) = REMOTOG
               ARRAY LIBARRAY[0:24]; % LIBARRAY IS USED TO KEEP INFORMATION AS        01561065 T  0003:0123:2
PRT(261) = LIBARRAY
                                     % TO LAST COMPILED LIBRARY SEQUENCE NUMBERS,     01561070 T  0003:0125:3
                                     % EACH ENTRY CONSISTS OF THREE WORDS CONTAINING; 01561080 T  0003:0125:3
                  FILE DSK1 DISK SERIAL [20:816](2,10,30);                  %116-    01561085 P  0003:0125:3
PRT(262) = DSK1
                  FILE DSK2 DISK SERIAL [20:450](2,30,30);                  %116-    01561087 P  0003:0130:0
PRT(263) = DSK2
             DEFINE LSTUSD=[9:3]#, FILEINDEX=[12:4]#, STOPPOINT=[16:16]#,             01561090 T  0003:0134:0
                    NEXTENTRY =[32:16]#; COMMENT SECOND WORD IS THE $$ SEQ NO;        01561100 T  0003:0134:0
             DEFINE  NCRLINK = [18:15]#,  LCRLINK= [33:15]#;                          01561110 T  0003:0134:0
             INTEGER LIBINDEX,LTLCR,MAXLTLCR,FILEINX,SEQSUM;                          01561120 T  0003:0134:0
PRT(264) = LIBINDEX
PRT(265) = LTLCR
PRT(266) = MAXLTLCR
PRT(267) = FILEINX
PRT(270) = SEQSUM
                                COMMENT LIBINDEX IS A INDEX INTO LIBARRAY         01561130 T  0003:0134:0
                                INDICATING LAST ENTRY MADE IN THE ARRAY.          01561140 T  0003:0134:0
                                LTLCR AND MAXLTLCR CORRESPOND TO TLCR AND         01561150 T  0003:0134:0
                                MAXTLCR USED IN READACARD. FILEINX IS THE         01561160 T  0003:0134:0
                                LIBRARY SWITCH FILE INDEX,  SEQSUM IS THE         01561170 T  0003:0134:0
                                SUM OF BASE SEQUENCE NUMBERS AT THIS POINT.       01561180 T  0003:0134:0
                                FINISHPT IS THE LAST RECORD NUMBER TO COMPILE;    01561190 T  0003:0134:0
             REAL RECOUNT,FINISHPT;                                               01561200 T  0003:0134:0
PRT(271) = RECOUNT
PRT(272) = FINISHPT
               BOOLEAN FIRSTIMEX; COMMENT USED TO INDICATE WHEN                   01561202 T  0003:0134:0
PRT(273) = FIRSTIMEX
                PROCESSING FIRST CARDIMAGE OF A NESTED CALL;                      01561204 T  0003:0134:0
               BOOLEAN CARDCALL; COMMENT TRUE IF NESTED CALL CAME FROM THE        01561206 T  0003:0134:0
```

```
PRT(274) = CARDCALL
                                    CARD READER ELSE FALSE;                                    01561208 T  0003:013:0
                               COMMENT RECOUNT IS THE LIBRARY RECORD COUNT;                     01561210 T  0003:013:0
                             BOOLEAN NORELEASE; COMMENT  NORELEASE ALLOWS PRINTING              01561215 T  0003:013:0
PRT(275) = NORELEASE
                               OF CURRENT BUFFER WHEN COMMING OUT OF LIBRARIES;                 01561217 T  0003:013:0
                          DEFINE NOROWS = 3#; COMMENT THIS IS THE MAXIMUM NUMBER OF DIRECTORY   01561220 T  0003:013:0
                                    BLOCKS PER LIBRARY TAPE;                                    01561230 T  0003:013:0
                         ARRAY DIRECTORY[0:3*NOROWS-1, 0:55]; COMMENT THIS IS THE ACTUAL        01561240 T  0003:013:0
PRT(276) = DIRECTORY
                                      DIRECTORY AND IS MADE UP AS FOLLOWS:                      01561250 T  0003:013:2
                                 A: 1 CAR= NUMBER OF DIRECTORY BLOCKS.                          01561260 T  0003:013:2
                                 B: 1 CHR = NUMBER OF CHARACTERS IN THE LIBRARY                 01561270 T  0003:013:2
                                        IDENTIFIER NAME.                                        01561280 T  0003:013:2
                                 C N CHR = ACTUAL ALPHA OF THE LIBRARY IDENTIFIER.              01561290 T  0003:013:2
                                 D: 3 CHR = STARTING RECORD NUMBER FOR THE ACTUAL               01561300 T  0003:013:2
                                        ENTRIES.                                                01561310 T  0003:013:2
                                    ITEMS B,C,D ARE THE REPEATED FOR EACH IDENTIFIER,           01561320 T  0003:013:2
                                    LIBRARY DIRECTORY ENTRIES ARE NOT SPLIT ACROSS              01561330 T  0003:013:2
                                      DIRECTORY BLOCKS.                                         01561340 T  0003:013:2
                                    ITEM B WHEN 0 INDICATES THE END OF THE DIRECTORY            01561350 T  0003:013:2
                                      AND THEN ITEM D WILL FOLLOW INDICATING THE                01561360 T  0003:013:2
                                      LAST SEQUENCE NUMBER + 1 PUT ON THE LIBRARY.              01561370 T  0003:013:2
                                    ITEM B WHEN  INDICATS LAST DIRECTORY ITEM IN THIS           01561380 T  0003:013:2
                                      BLOCK.                                                    01561390 T  0003:013:2
                                IN ORDER TO CHANGE:                                             01561400 T  0003:013:2
                                     NUMBER OF LIBRARY TAPES = ADD FILE DECLARATIONS AT         01561410 T  0003:013:2
                                                               01561020 = 01561050.             01561420 T  0003:013:2
                                                           = CHANGE "3" AT                      01561430 T  0003:013:2
                                                 NUMBER OF LIBRARY ENTRIES PER TAPE = CHANGE NOROWS  01561440 T  0003:013:2
                                                                                    AT      ;   01561450 T  0003:013:2
                                DEFINE                                                          %107-01561500 C  0003:013:2
                                    INSERTMAX = 20#, % CHANGE THIS IF YOU NEED MORE LEVELS OF INCLUDES  01561510 C  0003:013:2
                                    INSERTCOP = INSERTINFO[INSERTDEPTH,4]#, % = 1 IF COPY TO NEWTAPE  01561520 C  0003:013:2
                                    INSERTMID = INSERTINFO[INSERTDEPTH,0]#, % MFID OF THE LIBRARY FILE  01561530 C  0003:013:2
                                    INSERTFID = INSERTINFO[INSERTDEPTH,1]#, % FID  OF THE LIBRARY FILE  01561540 C  0003:013:2
                                    INSERTINX = INSERTINFO[INSERTDEPTH,2]#, % POINTER TO THE RECORD%107-01561550 C  0003:013:2
                                    INSERTSEQ = INSERTINFO[INSERTDEPTH,3]#; % LAST SEQUENCE TO BE INCLUD  01561560 C  0003:013:2
                                INTEGER SAVECARD, INSERTDEPTH;                                  %107-01561570 C  0003:013:2
PRT(277) = SAVECARD
PRT(300) = INSERTDEPTH
                                ARRAY INSERTINFO[0:INSERTMAX,0:4];                              %107-01561580 C  0003:013:2
PRT(301) = INSERTINFO
                                FILE LIBRARYFIL DISK RANDOM(2,10,30);                           %107-01561590 C  0003:014:3
PRT(302) = LIBRARYFIL
                                DEFINE LF = LIBRARYFIL#;                                        %107-01561600 C  0003:014:0
                                SAVE ARRAY LBUFF[0:9]; % INPUT BUFFER                           %107-01561610 C  0003:014:0
PRT(303) = LBUFF
                                REAL STREAM PROCEDURE CMPD(A,B);                                %107-01561620 C  0003:014:1
PRT(304) = CMPD
                                    BEGIN                                                       %107-01561630 C  0003:015:2
                                    SI:=A; DI:=B;                                               %107-01561640 C  0003:015:2
                                    IF 8 SC ≥ DC THEN                                           %107-01561650 C  0003:015:3
                                        BEGIN                                                   %107-01561660 C  0003:015:0
                                        SI:=SI"8; DI:=DI"8; TALLY:=2;                           %107-01561670 C  0003:015:0
                                        IF 8 SC = DC THEN TALLY:=1;                             %107-01561680 C  0003:015:2
                                        END;                                                    %107-01561690 C  0003:015:0
```

```
                  CMPD!=TALLY;                                                    %107-    01561700 C  0003!015410
                  END CMPD;                                                       %107-    01561710 C  0003!015410



               REAL C;                                                                     01562000 T  0003!015512
PRT(305) = C
                  COMMENT  C CONTAINS ACTUAL VALUE OF LAST CONSTANT SCANNED;               01563000 T  0003!015512
               REAL T;                                                                     01564000 T  0003!015512
PRT(306) = T
                  COMMENT  T IS A TEMPORARY CELL;                                          01565000 T  0003!015512
               INTEGER TCOUNT;                                                             01566000 T  0003!015512
PRT(307) = TCOUNT
                  COMMENT  TCOUNT IS A VARIABLE WHICH HOLDS A PREVIOUS VALUE OF COUNT      01567000 T  0003!015512
                           FOR THE USE OF CONVERT;                                         01568000 T  0003!015512
                  REAL STACKCT;                                                  %A       01568500 T  0003!015512
PRT(310) = STACKCT
               DEFINE  LOGI          =443#,                                             01569000 T  0003!015512
                       EXPI          =440#,                                             01570000 T  0003!015512
                       XTOTHEI       =480#,                                             01571000 T  0003!015512
                       GOTOSOLVER    =484#,                                             01572000 T  0003!015512
                       PRINTI        =477#,                                             01573000 T  0003!015512
                       MERGEI        =500#,                                             01573100 T  0003!015512
                       POWERSOFTEN   =670#,                                             01574000 T  0003!015512
                       LASTSEQUENCE  =166#,                                  %117-      01575000 P  0003!015512
                       LASTSEQROW    =  2#,                                             01576000 T  0003!015512
                       INTERPTO      =461#,                                             01577000 T  0003!015512
                       SUPERMOVER    =555#,                                             01577500 T  0003!015512
                       CHARI         =465#,                                             01578000 T  0003!015512
                       INTERPTI      =469#,                                             01579000 T  0003!015512
                       SORTI         =473#,                                             01579100 T  0003!015512
                       DIALER        =559#,                                             01579200 T  0003!015512
                       FILATTINT     =563#,                                             01579300 T  0003!015512
                       POWERALL      =567#,                                             01579350 T  0003!015512
                       SPECIALMATH   =570#,                                             01579355 T  0003!015512
                       SORTA         =673#;                                  %117-      01580000 P  0003!015512
                  COMMENT THESE DEFINES ARE USED TO TALK TO GNAT,  THEY GIVE THE INDEX    01581000 T  0003!015512
                            IN INFO OF THE CORRESPONDING ROUTINE;                          01582000 T  0003!015512
               INTEGER KOUNT,BUFFACCUM;                                                    01583000 T  0003!015512
PRT(311) = KOUNT
PRT(312) = BUFFACCUM
               INTEGER FILENO;                                                             01584000 T  0003!015512
PRT(313) = FILENO
                BOOLEAN                                                                    01585000 T  0003!015512
                     FUNCTOG,     COMMENT TELLS WHETHER PROCEDURE BEING DECLARED IS A     01586000 T  0003!015512
PRT(314) = FUNCTOG
                                  FUNCTION;                                                01587000 T  0003!015512
                     P2,          COMMENT GENERALY TELLS WHETHER OWN WAS SEEN;            01588000 T  0003!015512
PRT(315) = P2
                     P3,          COMMENT TELLS WHETHER SAVE WAS SEEN;                    01589000 T  0003!015512
PRT(316) = P3
                     P4,          COMMENT TELLS WHETHER AUXMEM WAS SEEN;                  01589500 T  0003!015512
PRT(317) = P4
                     VONF,        COMMENT VALUE OR OWN FIELD OF ELBAT WORD;               01590000 T  0003!015512
PRT(320) = VONF
                     FORMALF,     COMMENT FORMAL FIELD OF ELBAT WORD;                     01591000 T  0003!015512
```

```
PRT(321) = FORMALF
                    PTOG,       COMMENT TELLS THAT FORMAL PARAPART IS BEING PROCESSD;          01592000 T  0003:0155:2
PRT(322) = PTOG
               SPECTOG,                                                                        01593000 T  0003:0155:2
PRT(323) = SPECTOG
                    STOPENTRY,  COMMENT THIS MAKES THE ENTRY PROCEDURE ENTER ONLY              01594000 T  0003:0155:2
PRT(324) = STOPENTRY
                                        ONE ID AND THEN EIXT;                                  01595000 T  0003:0155:2
                    AJUMP;      COMMENT TELLS WHETHER A JUMP IS HANGING;                       01596000 T  0003:0155:2
PRT(325) = AJUMP
               BOOLEAN STOPDEFINE;                                                             01597000 T  0003:0155:2
PRT(326) = STOPDEFINE
               REAL CORESZ; % CORE ESTIMATE NEEDED FOR SORT.                                   01597100 T  0003:0155:2
PRT(327) = CORESZ
               INTEGER MAXSAVE;                                                                01598000 T  0003:0155:2
PRT(330) = MAXSAVE
                         COMMENT THIS CONTAINS THE SIZE OF THE MAXIMUM SAVE ARRAY              01599000 T  0003:0155:2
                         DECLARED. IT IS USED TO HELP DETERMINE STORAGE REQUIREMENTS           01600000 T  0003:0155:2
                         FOR THE PROGRAM PARAMETER BLOCK FOR THE OBJECT PROGRAM;               01601000 T  0003:0155:2
                                                                                               01602000 T  0003:0155:2
                REAL
                    KLASSF,     COMMENT CLASS IN LOW ORDER 7 BITS;                             01603000 T  0003:0155:2
PRT(331) = KLASSF
                    ADDRSF,     COMMENT ADDRESS IN LOW ORDER 11 BITS;                          01604000 T  0003:0155:2
PRT(332) = ADDRSF
                    LEVELF,     COMMENT LVL IN LOW ORDER 5 BITS;                               01605000 T  0003:0155:2
PRT(333) = LEVELF
                    LINKF,      COMMENT LINK IN LOW ORDER 13 BITS;                             01606000 T  0003:0155:2
PRT(334) = LINKF
                    INCRF,      COMMENT INCR CN LOW ORDER 8 BITS;                              01607000 T  0003:0155:2
PRT(335) = INCRF
                    PROINFO,    COMMENT CONTAINS ELBAT WORD FOR PROCEDURE BEING                01608000 T  0003:0155:2
PRT(336) = PROINFO
                                        DECLARED;                                              01609000 T  0003:0155:2
                    G,          COMMENT GLOBAL TEMPORARY FOR BLOCK;                            01610000 T  0003:0155:2
PRT(337) = G
                    TYPEV,      COMMENT USED TO CARRY CLASS OF IDENTIFIER                      01611000 T  0003:0155:2
PRT(340) = TYPEV
                                        BEING DECLARED;                                        01612000 T  0003:0155:2
                PROADO,           COMMENT CONTAINS ADDRESS OF PROCEDURE BEING                  01613000 T  0003:0155:2
PRT(341) = PROADO
                                        DECLARED;                                              01614000 T  0003:0155:2
                    MARK ,      COMMENT CONTAINS INDEX INTO INFO WHERE FIRST WORD               01615000 T  0003:0155:2
PRT(342) = MARK
                                        OF ADDITIONAL INFO FOR A PROCEDURE ENTRY;              01616000 T  0003:0155:2
                    PJ,         COMMENT FORMAL PARAMETER COUNTER;                              01617000 T  0003:0155:2
PRT(343) = PJ
                     J,         COMMENT ARRAY COUNTER;                                         01618000 T  0003:0155:2
PRT(344) = J
                    LASTINFO,   COMMENT INDEX TO LAST ENTRY IN INFO;                           01619000 T  0003:0155:2
PRT(345) = LASTINFO
                    NEXTINFO,   COMMENT INDEX FOR NEXT ENTRYIN INFO;                           01620000 T  0003:0155:2
PRT(346) = NEXTINFO
                    GLOBALNINFOO,COMMENT MAINTAINS VALUE OF NINFOO FROM BLOCK ON A             01620100 C  0003:0155:2
PRT(347) = GLOBALNINFOO
                                        GLOBAL LEVEL SO THAT THE PROCEDURE "ENTRY"             01620200 C  0003:0155:2
                                        CAN CHECK FOR DUPLICATE DECLARATIONS;   %118-          01620300 C  0003:0155:2
                    OLDNINFOO,  COMMENT REMEMBERS OLD VALUE OF GLOBALNINFOO;    %118-          01620400 C  0003:0155:2
```

```
PRT(350) = OLDNINFOO
                    FIRSTX,       COMMENT RELATIVE ADD OF FIRST EXECUTABLE CODE                    01621000 T  0003:015:2
PRT(351) = FIRSTX
                                  IN BLOCK, INITIALIZED TO 4095 EACH TIME;                         01622000 T  0003:015:2
                    SAVEL;        COMMENT SAVE LOCATION FOR FIXUPS IN BLOCK;                       01623000 T  0003:015:2
PRT(352) = SAVEL
              INTEGER NCII;       COMMENT THIS CONTAINS THE COUNT OF CONSTANTS                      01624000 T  0003:015:2
PRT(353) = NCII
                                  ENTERED IN INFO AT ANY GIVEN TIME;                               01625000 T  0003:015:2
               REAL FILETHING;  COMMENT HOLDS LINKS FOR STREAM RELEASES ;                          01625100 T  0003:015:2
PRT(354) = FILETHING
              PROCEDURE UNHOOK;FORWARD;                                                            01626000 T  0003:015:2
PRT(355) = UNHOOK
              PROCEDURE MAKEUPACCUM;FORWARD;                                                       01627000 T  0003:015:2
PRT(356) = MAKEUPACCUM
              DEFINE PURPT=[4:8]#,SECRET=2#;                                                       01628000 T  0003:015:2
               COMMENT THESE DEFINES GIVE THE NAMES OF THE WORD MODE OPERATORS.  THE               01629000 T  0003:015:2
                  NUMBERS REFER TO THE APPROPRIATE SECTION OF THE PRODUCT SPECS.  THE              01630000 T  0003:015:2
                  FULL NAME IS ALSO GIVEN;                                                        01631000 T  0003:015:2
               DEFINE                                                                             01632000 T  0003:015:2
                 ADD =  16#,     COMMENT (0101) 7,4,2,1    ADD;                                    01633000 T  0003:015:2
                 BBC =  22#,     COMMENT (0131) 7,4,5,4    BRANCH BACKWARD CONDITIONAL;            01634000 T  0003:015:2
                 BBW = 534#,     COMMENT (4131) 7,4,5,2    BRANCH BACKWARD;                        01635000 T  0003:015:2
                 BFC =  38#,     COMMENT (0231) 7,4,5,3    BRANCH FORWARD CONDITIONAL;             01636000 T  0003:015:2
                 BFW = 550#,     COMMENT (4231) 7,4,5,1    BRANCH FORWARD;                         01637000 T  0003:015:2
                 CDC = 168#,     COMMENT (1241) 7,4,10,4   CONSTRUCT DESCRIPTOR CALL;              01638000 T  0003:015:2
                 CHS = 134#,     COMMENT (1031) 7,4,7,11   CHANGE SIGN;                            01639000 T  0003:015:2
                 COC =  40#,     COMMENT (0241) 7,4,10,3   CONSTRUCT OPERAND CALL;                 01640000 T  0003:015:2
                 COM = 130#,     COMMENT (1011) 7,4,10,5   COMMUNICATION OPERATOR;                 01641000 T  0003:015:2
                 DEL =  10#,     COMMENT (0051) 7,4,9,3    DELETE;                          %A     01642000 T  0003:015:2
                 DUP = 261#,     COMMENT (2025) 7,4,9,2    DUPLICATE;                              01643000 T  0003:015:2
                 EQL = 581#,     COMMENT (4425) 7,4,4,3    EQUAL;                                  01644000 T  0003:015:2
                 LBC = 278#,     COMMENT(2131) 7,4,5,9     GO BACKWARD CONDITIONAL;                01645000 T  0003:015:2
                 LBU = 790#,     COMMENT(6131) 7,4,5,7     GO BACKWARD (WORD);                     01646000 T  0003:015:2
                 GEQ =  21#,     COMMENT (0125) 7,4,4,2    GREATER THAN OR EQUAL TO;               01647000 T  0003:015:2
                 LFC = 294#,     COMMENT(2231) 7,4,5,8     GO FORWARD CONDITIONAL;                 01648000 T  0003:015:2
                  LFU = 806#,    COMMENT(6231) 7,4,5,6     GO FORWARD (WORD);                      01649000 T  0003:015:2
                 GTR =  37#,     COMMENT (0225) 7,4,4,1    GREATER THAN;                           01650000 T  0003:015:2
                 IDV = 384#,     COMMENT (3001) 7,4,2,5    INTEGER DIVIDE;                         01651000 T  0003:015:2
                 INX =  24#,     COMMENT (0141) 7,4,10,2   INDEX;                                  01652000 T  0003:015:2
                 ISD = 532#,     COMMENT (4121) 7,4,6,3    INTEGER STORE DESTRUCTIVE;              01653000 T  0003:015:2
                 ISN = 548#,     COMMENT (4221) 7,4,6,4    INTEGER STORE NON-DESTRUCT;             01654000 T  0003:015:2
                 LEQ = 533#,     COMMENT (4125) 7,4,4,4    LESS THAN OR EQUAL TO;                  01655000 T  0003:015:2
                 LND =  67#,     COMMENT (0415) 7,4,3,1    LOGICAL AND;                            01656000 T  0003:015:2
                 LNG =  19#,     COMMENT (0115) 7,4,3,4    LOGICAL NEGATE;                         01657000 T  0003:015:2
                 LOD = 260#,     COMMENT (2021) 7,4,10,1   LOAD OPERATOR;                          01658000 T  0003:015:2
                 LOR =  35#,     COMMENT (0215) 7,4,3,2    LOGICAL OR;                             01659000 T  0003:015:2
                 LQV = 131#,     COMMENT (1015) 7,4,3,3    LOGICAL EQIVALENCE;                     01660000 T  0003:015:2
                 LSS = 549#,     COMMENT (4225) 7,4,4,5    LESS THAN;                              01661000 T  0003:015:2
               MDS = 515#,     COMMENT (4015) 7,4,7,7    SET FLAG BIT;                           01661100 T  0003:015:2
                 MKS =  72#,     COMMENT (0441) 7,4,8,1    MARK STACK;                             01662000 T  0003:015:2
                 MUL =  64#,     COMMENT (0401) 7,4,2,3    MULTIPLY;                               01663000 T  0003:015:2
                 NEQ =  69#,     COMMENT (0425) 7,4,4,6    NOT EQUAL TO;                           01664000 T  0003:015:2
                 NOP =  11#,     COMMENT (0055) 7,4,7,1    NO OPERATION;                           01665000 T  0003:015:2
                 PRL =  18#,     COMMENT (0111) 7,4,10,6   PROGRAM RELEASE;                        01666000 T  0003:015:2
                 PRTE=  12#,     COMMENT (0061) 7,4,10,9   EXTEND PRT;                             01667000 T  0003:015:2
                 RDV = 896#,     COMMENT (7001) 7,4,2,6    REMAINDER DIVIDE;                       01668000 T  0003:015:2
                 RTN =  39#,     COMMENT (0235) 7,4,8,3    RETURN NORMAL;                          01669000 T  0003:015:2
```

```
                    RTS = 167#,     COMMENT (1235) 7,4,8,4    RETURN SPECIAL;
                    SND = 132#,     COMMENT (1021) 7,4,6,2    STORE NON-DESTRUCTIVE;
                    SSN =  70#,     COMMENT (0431) 7,4,7,10   SET SIGN NEGATIVE;
                    SSP = 582#,     COMMENT (4431) 7,4,7,10   SET SIGN PLUS;
                    STD =  68#,     COMMENT (0421) 7,4,6,1    STORE DESTRUCTIVE;
                    SUB =  48#,     COMMENT (0301) 7,4,2,2    SUBTRACT;
                    XCH = 133#,     COMMENT (1025) 7,4,9,1    EXCHANGE;
                    XIT =  71#,     COMMENT (0435) 7,4,8,2    EXIT;
                    ZP1 = 322#;     COMMENT (2411) 7,4,10,8   CONDITIONAL HALT;
                  COMMENT  THESE DEFINES ARE USED BY EMITD;
                  DEFINE
                    DIA =  45#,     COMMENT (XX55) 7,4,7,1    DIAL A;
                    DIB =  49#,     COMMENT (XX61) 7,4,7,2    DIAL B;
                    TRB =  53#;     COMMENT (XX65) 7,4,7,3    TRANSFER BITS;
                REAL MAXSTACK,STACKCTR;
PRT(357) = MAXSTACK
PRT(360) = STACKCTR
                INTEGER MAXROW;
PRT(361) = MAXROW
                        COMMENT THIS CONTAINS THE MAXIMUM ROW SIZE OF ALL NON-SAVE
                        ARRAYS DECLARED, ITS USE IS LIKE THAT OF MAXSAVE;
              INTEGER SEGSIZEMAX; COMMENT CONTAINS MAX SEGMENT SIZE;
PRT(362) = SEGSIZEMAX
              INTEGER F;
PRT(363) = F
               STREAM PROCEDURE MOVECODE(EDOC,TEDOC);
PRT(364) = MOVECODE
                   BEGIN LOCAL T1,T2,T3;
                        SI←EDOC;T1←SI;SI←TEDOC;T2←SI;SI←LOC EDOC;SI←SI+3;DI←LOC T3;
                        DI←DI+5;SKIP 3 DB;15(IF SB THEN DS←1 SET ELSE DS←1 RESET;
                        SKIP 1 SB);SI←LOC EDOC;DI←LOC T2; DS← 5 CHR;3(IF SB THEN DS←
                        1  SET ELSE DS ← 1 RESET; SKIP 1 SB);DI←T3;SI←LOC T2;DS←WDS;
                        DI←LOC T3;DI←DI+5;SKIP 3 DB;SI←LOC TEDOC;SI←SI+3;15(IF SB
                        THEN DS←1 SET ELSE DS←1 RESET;SKIP 1 SB);SI← LOC TEDOC;DI←LOC
                        T1; DS←5 CHR;3(IF SB THEN DS←1 SET ELSE DS←1 RESET;SKIP 1 SB);
                        DI←T3;SI←LOC T1;DS←WDS
                   END;


                        REAL NLO,NHI,TLO,THI;
PRT(365) = NLO
PRT(366) = NHI
PRT(367) = TLO
PRT(370) = THI

                        BOOLEAN DPTOG;
PRT(371) = DPTOG
                        COMMENT  THE ABOVE THINGS ARE TEMP STORAGE FOR DOUBLE NOS;
              DEFINE FZERO=896#;
              REAL T1,T2,N,K,AKKUM;
PRT(372) = T1
PRT(373) = T2
PRT(374) = N
PRT(375) = K
PRT(376) = AKKUM
              BOOLEAN STOPGSP;
```

```
01670000 T  0003:015:2
01671000 T  0003:015:2
01671100 T  0003:015:2
01672000 T  0003:015:2
01673000 T  0003:015:2
01674000 T  0003:015:2
01675000 T  0003:015:2
01676000 T  0003:015:2
01677000 T  0003:015:2
01678000 T  0003:015:2
01679000 T  0003:015:2
01680000 T  0003:015:2
01681000 T  0003:015:2
01682000 T  0003:015:2
01683000 T  0003:015:2

01684000 T  0003:015:2

01685000 T  0003:015:2
01686000 T  0003:015:2
01687000 T  0003:015:2

01688000 T  0003:015:2

01688010 T  0003:015:2

01688020 T  0003:015:2
01688030 T  0003:015:0
01688040 T  0003:015:3
01688050 T  0003:015:3
01688060 T  0003:016:3
01688070 T  0003:016:3
01688080 T  0003:016:3
01688090 T  0003:016:2
01688100 T  0003:016:3
01688110 T  0003:017:1

01689000 T  0003:017:1

01690000 T  0003:017:1

01691000 T  0003:017:1
01692000 T  0003:017:1
01693000 T  0003:017:1

01694000 T  0003:017:1
```

```
PRT(377) = STOPGSP
                INTEGER BUP;                                                    01695000 T  0003:0170:1
PRT(400) = BUP
                        COMMENT UNIQUE GLOBAL TEMP FOR BLOCK;                   01696000 T  0003:0170:1
                ARRAY GTA1[0:10];                                               01697000 T  0003:0170:1
PRT(401) = GTA1
                BOOLEAN ARRAY SPRT[0:31];                                       01698000 T  0003:0173:3
PRT(402) = SPRT
                    COMMENT  SPRT IS TO BE CONSIDERED TO BE AN ARRAY OF 32 32 BIT   01699000 T  0003:0176:0
                             FIELDS.  THE 32 BITS ARE IN THE LOW ORDER PART OF EACH 01700000 T  0003:0176:0
                             WORD.  THE BIT IS ON IF AND ONLY IF THE CORRESPONDING  01701000 T  0003:0176:0
                             PRT CELL HAS A PERMANENT ASSIGNMENT;                   01702000 T  0003:0176:0
                INTEGER PRTI,PRTIMAX;                                           01703000 T  0003:0176:0
PRT(403) = PRTI
PRT(404) = PRTIMAX
                    COMMENT  PRTIMAX GIVES NEXT PRT CELL AVAILABLE FOR PERMANENT ASSIGN-  01704000 T  0003:0176:0
                             MENT.  PRTI GIVES NEXT PRT CELL POSSIBLY AVAILABLE FOR 01705000 T  0003:0176:0
                             TEMPORARY ASSIGNMENT;                              01706000 T  0003:0176:0
                DEFINE ALPHASIZE = [12:6]#;  COMMENT ALPHASIZE IS THE DEFINE FOR THE BIT  01707000 T  0003:0176:0
                                             POSITION IN THE SECOND WORD OF INFO WHICH    01708000 T  0003:0176:0
                                             CONTAINS THE LENGTH OF ALPHA;                01709000 T  0003:0176:0
                DEFINE EDOCINDEX = L.[36:3],L.[39:7]#;  COMMENT EDOCINDEX IS THE WORD     01710000 T  0003:0176:0
                                                        PORTION OF L SPLIT INTO A ROW AND 01711000 T  0003:0176:0
                                                        COLUMN INDEX FOR EDOC;           01712000 T  0003:0176:0
                DEFINE CPLUS1 = 769#;  COMMENT SEE COMMENT AT CPLUS2 DEFINE;            01713000 T  0003:0176:0
                DEFINE CPLUS2 = 770#;  COMMENT CPLUS1 AND CPLUS2 ARE EXPLICIT CONSTANTS 01714000 T  0003:0176:0
                                       USED IN THE GENERATION OF C-RELATIVE CODE;       01715000 T  0003:0176:0
                PROCEDURE FLAG(ERRNUM); VALUE ERRNUM; INTEGER ERRNUM; FORWARD;          01716000 T  0003:0176:0
PRT(405) = FLAG
                ALPHA PROCEDURE B2D(B); VALUE B; REAL B; FORWARD;                       01717000 T  0003:0176:0
PRT(406) = B2D
                BOOLEAN MACROID;                                                01717800 T  0003:0176:0
PRT(407) = MACROID
                REAL PROCEDURE FIXDEFINEINFO(T); VALUE T; REAL T; FORWARD;      01717900 T  0003:0176:0
PRT(410) = FIXDEFINEINFO
                PROCEDURE DEFINEPARAM(D,N); VALUE D,N; INTEGER D,N; FORWARD;    01717950 T  0003:0176:0
PRT(411) = DEFINEPARAM
                PROCEDURE ERR (ERRNUM); VALUE ERRNUM; INTEGER ERRNUM; FORWARD;  01718000 T  0003:0176:0
PRT(412) = ERR
                REAL PROCEDURE TAKE(X); VALUE X; INTEGER X; FORWARD;            01718100 T  0003:0176:0
PRT(413) = TAKE
                PROCEDURE PUT(W,X); VALUE W,X; REAL W,X; FORWARD                ;01718200 T  0003:0176:0
PRT(414) = PUT
                INTEGER PROCEDURE GIT(L); VALUE L; REAL L; FORWARD;             01719000 T  0003:0176:0
PRT(415) = GIT
                BOOLEAN LISTMODE;                                               01720000 T  0003:0176:0
PRT(416) = LISTMODE
                    COMMENT LISTMODE IS A VARIABLE USED BY FORSTMT TO DECEIDE IF A LIST  01721000 T  0003:0176:0
                            IS BEING GENERATED OR A STATEMENT;                  01722000 T  0003:0176:0
                INTEGER LSTR;                                                   01723000 T  0003:0176:0
PRT(417) = LSTR
                    COMMENT LSTR GIVES THE LOCATION OF FIRST SYLABLE OF A LIST.  IT IS  01724000 T  0003:0176:0
                            USED BY LISTELEMENT TO COMPUTE VALUES TO STORE IS LSTRTN;   01725000 T  0003:0176:0
                PROCEDURE SCANNER;  FORWARD;                                    01730000 T  0003:0176:0
PRT(420) = SCANNER
                    COMMENT MKABS CONVERTS A DESCRIPTOR  TO AN ABSOLUTE ADDRESS;    01732000 T  0003:0176:0
                REAL STREAM PROCEDURE MKABS(A);                                 01733000 T  0003:0176:0
PRT(421) = MKABS
```

```
                    BEGIN DI ← A; MKABS ← DI END MKABS;                                    01734000 T  0003:017610

               STREAM PROCEDURE MOVE(W)"WORDS FROM"(A)"TO"(B); VALUE W;                    01735000 T  0003:017811
PRT(422) = MOVF
                    BEGIN LOCAL T;                                                         01736000 T  0003:017811
                         SI←LOC W; DI←LOC T; SI←SI+6; DI←DI+7; DS←CHR;                     01736100 T  0003:017912
                         SI←A; DI←B; T(DS←32 WDS; DS←32 WDS); DS←W WDS;                    01736200 T  0003:018010
                    END MOVF;                                                              01736300 T  0003:018312


               STREAM PROCEDURE ZEROUT(DEST,NDIV32,NMOD32);                                01737000 T  0003:018312
PRT(423) = ZEROUT
                                   VALUE NDIV32,NMOD32 ;                                   01737050 T  0003:018312
                    BEGIN  DI := DEST;                                                     01737100 T  0003:018312
                           NDIV32(32(DS :=8 LIT"0"));                                      01737150 T  0003:018410
                           NMOD32(DS := 8 LIT"0");                                         01737200 T  0003:018712
                    END;                                                                   01737250 T  0003:018913


               COMMENT  "BLANKET" BLANKS OUT N+1 WORDS IN "THERE";                         01737300 T  0003:019010
               STREAM PROCEDURE BLANKET(N,THERE); VALUE N;                                 01737350 T  0003:019010
PRT(424) = BLANKET
                    BEGIN                                                                  01737400 T  0003:019010
                    DI:=THERE; DS:=8 LIT" "; SI:=THERE; DS:=N WDS;                         01737450 T  0003:019010
                    END BLANKET;                                                           01737500 T  0003:019210


                                                                                           01738000 T  0003:019211
                                                                                           01739000 T  0003:019211
                                                                                           01740000 T  0003:019211
               PROCEDURE STEPIT; FORWARD;                                                  01741000 T  0003:019211
PRT(425) = STEPIT
                    COMMENT SEQCHANGE WILL CONV A MACHING NO. TO PROPER OUTPUT FORM;        01741100 T  0003:019211
               STREAM PROCEDURE CHANGESEQ(VAL, OLDSEQ);      VALUE OLDSEQ;                 01741200 T  0003:019211
PRT(426) = CHANGESEQ
                         BEGIN                                                             01741300 T  0003:019211
                              DI ← OLDSEQ; SI←VAL ;  DS ← 8 DEC                            01741400 T  0003:019312
                         END;                                                              01741500 T  0003:019313


               STREAM PROCEDURE SEQUENCEWARNING(L);                                        01742100 T  0003:019410
PRT(427) = SEQUENCEWARNING
                    BEGIN DI:=L; DI:=DI-8; DS:=24 LIT "SEQUENCE WARNING<<<<<<<<"; END;     01742110 T  0003:019410
```

```
                BOOLEAN STREAM PROCEDURE NONBLANK(FCR); VALUE FCR;                    01742200 T  0003:019:0
PRT(430) = NONBLANK
                    COMMENT NONBLANK SCANS CARD FOR ALL BLANKS--                       01742300 T  0003:019:0
                            TRUE IF ANY VISIBLE CHARACTER ;                           01742400 T  0003:019:0
                        BEGIN                                                         01742500 T  0003:019:0
                            LABEL NED;                                                01742600 T  0003:019:0
                            SI←FCR;                                                   01742700 T  0003:019:0
                            TALLY←0;                                                  01742800 T  0003:019:0
                            2(36(IF SC ≠ " " THEN JUMP OUT 2 TO NED; SI← SI+1));      01742900 T  0003:019:1
                            TALLY←63;                                                 01743000 T  0003:020:2
                        NED: TALLY←TALLY+1;                                           01743100 T  0003:020:2
                            NONBLANK←TALLY                                            01743200 T  0003:020:0
                        END NONBLANK;                                                 01743300 T  0003:020:1



                INTEGER FAULTLEVEL; COMMENT THIS IS FOR THE RUN-TIME ERROR KLUDGE--    01750000 T  0003:020:3
PRT(431) = FAULTLEVEL
                                    GIVES THE LOWEST LEVEL AT WHICH THERE IS AN ACTIVE 01751000 T  0003:020:3
                                    FAULT DECL OR LABEL USED IN A FAULT STATEMENT;    01752000 T  0003:020:3
                BOOLEAN FAULTOG; COMMENT FAULTSTMT USES THIS TO TELL DEXP TO  WORRY    01753000 T  0003:020:3
PRT(432) = FAULTOG
                                    ABOUT FAULTLEVEL;                                 01754000 T  0003:020:3
                INTEGER SFILENO;  COMMENT FILENO OF FIRST SORT FILE;                  01755000 T  0003:020:3
PRT(433) = SFILENO
               STREAM PROCEDURE GETVOID(VP,NCR,VR,LCR,SEQ); VALUE NCR;                01756000 T  0003:020:3
PRT(434) = GETVOID
                    BEGIN                                                             01757000 T  0003:020:3
                    LABEL L,TRANS;                                                    01758000 T  0003:020:0
                    LOCAL N;                                                          01759000 T  0003:020:0
                    SI:=SEQ; DI:=LCR; DI:=DI"1; DS:=LIT"%"; % PUT "%" IN CC 72.       01759100 T  0003:020:0
                    DS:=WDS; % RESTORE SEQ. NO. FOR $VOID(T) CARDS.                   01759200 T  0003:020:2
                    SI:=LCR; DI:=LOC N; DS:=CHR; % SAVE COL. 73                       01760000 T  0003:020:3
                    SI:=NCR; DI:=VP; DS:=8 LIT "0";                                   01761000 T  0003:020:0
                    2(36(IF SC=" " THEN SI:=SI+1 ELSE JUMP OUT 2 TO L));              01762000 T  0003:020:0
                    SI:=LCR; TALLY:=8; GO TRANS;% NO VOID RANGE FOUND, USE 73-80.     01763000 T  0003:021:1
                L:                                                                    01764000 T  0003:021:3
                    IF SC="'" THEN                                                    01765000 T  0003:021:3
                        BEGIN                                                         01766000 T  0003:021:1
                        SI:=SI+1; DI:=LCR; DS:=1 LIT"'"; % STOPPER FOR SCAN           01767000 T  0003:021:1
                        NCR:=SI; % TEMP. STORAGE, SINCE NCR IS "LOCAL" TO GETVOID.    01768000 T  0003:021:0
                        8(IF SC="'" THEN JUMP OUT ELSE                                01769000 T  0003:021:0
                            BEGIN  TALLY:=TALLY+1; SI:=SI+1 END));                    01770000 T  0003:021:3
                        END                                                           01771000 T  0003:021:2
                    ELSE BEGIN                                                        01772000 T  0003:021:2
                        NCR:=SI; % TEMP. STORAGE, SINCE NCR IS "LOCAL" TO GETVOID.    01773000 T  0003:021:2
                        DI:=LCR; DS:=1 LIT" "; % STOPPER FOR SCAN                     01774000 T  0003:021:3
                        8(IF SC=" " THEN JUMP OUT ELSE                                01775000 T  0003:021:0
                            BEGIN  TALLY:=TALLY+1; SI:=SI+1 END));                    01776000 T  0003:021:3
                        END;                                                          01777000 T  0003:022:2
                TRANS:                                                                01778000 T  0003:022:2
                    SI:=LOC N; DI:=LCR; DS:=CHR; % RESTORE COLUMN 73                  01779000 T  0003:022:2
                    SI:=NCR; DI:=VP; DI:=DI+8; % RESTORE POINTERS.                    01780000 T  0003:022:3
                    N:=TALLY; DI:=DI"N; DS:=N CHR;                                    01781000 T  0003:022:1
                    DI:=DI"8; VP:=DI; % I.E., "LOC VP":=DI.                           01782000 T  0003:023:3
                    DI:=VR; SI:=LOC VP; DS:=WDS;% ADDRESS OF VOID RANGE.              01783000 T  0003:022:0
                    END OF GETVOID;                                                   01784000 T  0003:022:2
```

```
                REAL VOIDCR,VOIDPLACE;                                                                  01785000 T  0003:022:2
PRT(435) = VOIDCR
PRT(436) = VOIDPLACE
                BOOLEAN SORTMERGETOG;                                                                   01786000 T  0003:022:2
PRT(437) = SORTMERGETOG
                 FORMAT PRINTSEGNO(X88,"START OF SEGMENT ********** ",I4),                              01800000 T  0003:022:2
                                                                                                START OF SEGMENT **********    4
PRT(440) = PRINTSEGNO
                        PRINTSIZE(X88,I4," IS ",I4," LONG, NEXT SEG ",I4),                              01801000 T  0004:022:2
                        BUG(X24,4(A4,X2));                                                              01802000 T  0004:022:2
                                                                                                    4 IS   29 LONG, NEXT SEG    3
                PROCEDURE DATIME;                                                                       01820000 T  0003:022:2
PRT(441) = DATIME
                    BEGIN                                                                               01821000 T  0003:022:2
                    INTEGER H,MIN,Q; ALPHA N1,N2;                                                       01822000 T  0003:022:2
                                                                                                START OF SEGMENT **********    5
STACK(F+2) = H
STACK(F+3) = MIN
STACK(F+4) = Q
STACK(F+5) = N1
STACK(F+6) = N2
                    ALPHA STREAM PROCEDURE DATER(DATE); VALUE DATE;                                     01823000 T  0005:000:0
PRT(442) = DATER
                        BEGIN                                                                           01824000 T  0005:000:0
                        DI:=LOC DATER; SI:=LOC DATE; SI:=SI+2;                                          01825000 T  0005:000:0
                        2(DS:=2 CHR; DS:=LIT"/"); DS:=2 CHR;                                            01826000 T  0005:000:1
                        END OF DATER;                                                                   01827000 T  0005:002:0
```

```
                        H:=TIME1 DIV 216000; MIN:=(TIME1 DIV 3600) MOD 60;                              01828000 T  0005:000:2
                        N1:=CODE.MFID; N2:=CODE.FID;                                                    01828500 T  0005:000:2
                        WRITE(LINE,                                                                     01829000 T  0005:001:3
                     $ SET OMIT = NOT ALGOL                                                             01829900 T  0005:001:2
                    "XVI,0,122"                                                             %123-       01831000 P  0005:001:2
                                                                                                          <<<<<<<<<<<<<<<<<<<<
            ERROR NUMBER 141 -- XVI,0,12,                                                               01832000 T  0005:001:3
                              ,"  ",A6,"DAY, ",0,",  ",I2,":",A2,X1,A3,                                01832500 T  0005:001:3
                             ////X45,A1,A6,"/",A1,A6,/X45,15("=")//>,                                    01833000 T  0005:001:3
                              TIME(6),DATER(TIME(5)),12×REAL(Q:=H MOD 12=0)+Q,                          01834000 T  0005:001:3
                              Q:=MIN MOD 10+(MIN DIV 10)×64,                                            01835000 T  0005:001:3
                              IF H≥12 THEN "PM," ELSE "AM,",                                            01835500 T  0005:001:3
                              N1,[6:6],N1,N2,[6:6],N2);
                        IF MERGETOG THEN % INDICATE NAME OF SOURCE FILE.                        %120-       01835600 C  0005:001:3
                          WRITE(LINE,<X40,"SOURCE FILE: ",A1,A6,"/",A1,A6,//>,                  %120-       01835700 C  0005:001:6:1
PRT(443) = *FORMAT DESCRIPTOR*
                                                                                                    6 IS   14 LONG, NEXT SEG    5
                          (N1:=TAPE.MFID),[6:6],N1,(N2:=TAPE.FID),[6:6],N2);                    %120-       01835800 C  0005:001:9:3
PRT(444) = *LIST, LABEL, OR SEGMENT DESCRIPTOR*
PRT(445) = OUTPUT(W)
                        NOHEADING:=FALSE;                                                               01836000 T  0005:004:3:2
                        END OF DATIME;                                                                  01837000 T  0005:004:4:0
```

```
                DEFINE DOT= BEGIN IF ELCLASS = PERIOD THEN DOTIT END#;                                  01841000 T  0003:022512
                COMMENT  THIS SECTION CONTAINS ALL CODE PERTAINENT TO READING CARDS                     02000000 T  0003:022512
                         AND SCANNING THEM;                                                             02001000 T  0003:022512
                    BOOLEAN STREAM PROCEDURE LOOK(ACC1,DIR,ROW,STRTPOS,STOPOS);                         02001020 T  0003:022512
PRT(446) = LOOK
                                        VALUE          ROW                      ;                       02001030 T  0003:022512
                BEGIN COMMENT LOOK DOES THE ACTUAL DIRECTORY SEARCH.  IT                                02001040 T  0003:022512
                     REPORTS TRUE IF THE ITEM WAS NOT FOUND IN THE DIRECTORY                            02001050 T  0003:022610
                     ;                                                                                  02001060 T  0003:022610
                     LOCAL DPPOS,TEMP,LGTH;                                                             02001070 T  0003:022610
                     LABEL LOOP,EXIT;                                                                   02001080 T  0003:022610
                     SI←DIR;              ROW(SI←SI+8);      DPPOS←SI;                              02001090 T  0003:022610
                     DI←LOC TEMP;         DS←WDS;            SI←TEMP;                               02001100 T  0003:022810
                     SI←SI+8;                                                                           02001110 T  0003:022811
                LOOP:DI ← LOC LGTH;       DI←DI+7;           DS←CHR;                                02001120 T  0003:022912
                     DI←ACC1;             DI←DI+2;           SI←SI-1;                               02001130 T  0003:022913
                     IF SC = DC                                                                         02001140 T  0003:023011
                     THEN BEGIN COMMENT THE LENGTHS ARE EQUAL;                                          02001150 T  0003:023011
                               IF LGTH SC = DC                                                          02001160 T  0003:023112
                               THEN BEGIN COMMENT FOUND IT;                                             02001170 T  0003:023210
                                          DI←STRTPOS;DS←5 LIT "0"; DS←3 CHR;                            02001180 T  0003:023210
                                          IF SC = "0"                                                   02001190 T  0003:023410
                                          THEN BEGIN COMMENT WE MAY BE IN THE                           02001200 T  0003:023411
                                                     WRONG ROW;                                         02001210 T  0003:023411
                                               SI←SI+1;DI←LOC LOOK;                                     02001220 T  0003:023411
                                               IF 3 SC = DC                                             02001230 T  0003:023513
                                               THEN BEGIN COMMENT WE ARE                                02001240 T  0003:023610
                                                          IN THE WRONG                                  02001250 T  0003:023610
                                                          ROW;                                          02001260 T  0003:023610
                                                          SI←DPPOS;                                     02001270 T  0003:023610
                                                          SI←SI+8;                                      02001280 T  0003:023712
                                                          DPPOS←SI;                                     02001290 T  0003:023712
                                                          DI←LOC TEMP;                                  02001300 T  0003:023713
                                                          DS←WDS;                                       02001310 T  0003:023713
                                                          SI←TEMP;                                      02001320 T  0003:023810
                                                    END                                                 02001330 T  0003:023810
                                               ELSE SI←SI-4;                                            02001340 T  0003:023810
                                          END;                                                          02001350 T  0003:023811
                                     DI←LOC LGTH; DI←DI+7; DS←CHR;                                      02001360 T  0003:023811
                                       SI←SI+ LGTH;                                                     02001360 T  0003:023913
                                     DI←STOPOS; DS←5 LIT"0";                                            02001375 T  0003:024010
                                     DS←3 CHR;  GO TO EXIT;                                             02001380 T  0003:024112
                               END;                                                                     02001390 T  0003:024113
                          SI←SI+3;                                                                      02001400 T  0003:024113
                     END                                                                                02001410 T  0003:024210
                ELSE BEGIN COMMENT THE LENGTHS ARE NOT EQUAL;                                           02001420 T  0003:024210
                   SI←SI-1;                                                                             02001430 T  0003:024210
                          IF SC = "0"                                                                   02001440 T  0003:024211
                          THEN BEGIN COMMENT MAY BE NEW ROW;                                            02001450 T  0003:024211
                                     SI←SI+1;    DI←LOC LOOK;                                           02001460 T  0003:024312
                                     IF 3 SC = DC                                                       02001470 T  0003:024410
                                     THEN BEGIN COMMENT CHANGE ROWS;                                    02001480 T  0003:024410
                                                SI←DPPOS;SI←SI+8;DPPOS←SI;                              02001490 T  0003:024411
```

```
                                                    DI←LOC TEMP;      DS←WDS;          02001500 T  00031024!3
                                                    SI←TEMP;                           02001510 T  00031024!0
                                                END                                    02001520 T  00031024!1
                                            ELSE BEGIN COMMENT IT IS NOT HERE;         02001530 T  00031024!1
                                                    TALLY←1; LOOK←TALLY;               02001540 T  00031024!1
                                                    GO TO EXIT;                        02001550 T  00031024!2
                                                END;                                   02001560 T  00031024!3
                                    GO TO LOOP;                                        02001563 T  00031024!3
PRT(447) = *LIST, LABEL, OR SEGMENT DESCRIPTOR*
                                        END;                                           02001565 T  00031024!3
                                    SI←SI+LGTH; SI←SI+4; COMMENT POSITION TO NEXT ID.; 02001568 T  00031024!3
                                      END;                                             02001570 T  00031024!1
                                GO TO LOOP;                                            02001580 T  00031024!1
                            EXIT;                                                      02001590 T  00031024!1
                            END LOOK;                                                  02001600 T  00031024!2


                    %*************************************************************** 02001605 C  00031025!0
                    %                                                               %116- 02001610 C  00031025!0
                    %       MISCELLANEOUS CROSS REFERENCE PROCEDURES                %116- 02001615 C  00031025!0
                    %                                                               %116- 02001620 C  00031025!0
                    %*************************************************************** 02001630 C  00031025!0
                    %                                                               %116- 02001635 C  00031025!0
                    PROCEDURE CROSSREFIT(INDEX,SEQNO,REFTYPE);                      %116- 02001640 C  00031025!0
PRT(450) = CROSSREFIT
                      VALUE INDEX,SEQNO,REFTYPE;                                    %116- 02001645 C  00031025!0
                      REAL  INDEX,SEQNO,REFTYPE;                                    %116- 02001650 C  00031025!0
                    BEGIN                                                           %116- 02001655 C  00031025!0
                     IF XREFINFO[INDEX].IDNOF ≠ 0 THEN % SAVE                       %116- 02001660 C  00031025!0
                     BEGIN                                                          %116- 02001665 C  00031025!3
                      IF XREFPT > 29 THEN % NO SLOTS LEFT IN ARRAY, WRITE IT OUT.   %116- 02001670 C  00031025!4
                        BEGIN                                                       %116- 02001675 C  00031025!2
                          WRITE(DSK2,30,XREFAY2[*]);                                %116- 02001680 C  00031025!3
                          XREFPT := 0;                                              %116- 02001685 C  00031025!3
                        END;                                                        %116- 02001690 C  00031026!1
                      XREFAY2[XREFPT] := SEQNO & REFTYPE TYPEREF & XREFINFO[INDEX]  %116- 02001695 C  00031026!1
                                                          REFIDNOF;                 %116- 02001700 C  00031026!1
                      XREFPT := XREFPT + 1; % EVEN THOUGH THE ARRAY MAY BE FULL NOW WE %116- 02001705 C  00031026!0
                         /                  % CANT WRITE IT OUT BECAUSE SOME ROUTINES %116- 02001710 C  00031026!3
                                            % WILL LOOK BACK AT THE ENTRY WE JUST PUT %116- 02001715 C  00031026!3
                                            % IN AND FIX IT UP.                      %116- 02001720 C  00031026!3
                     END;                                                           %116- 02001725 C  00031026!3
                    END OF CROSSREFIT;                                              %116- 02001730 C  00031026!3



                    %                                                               %116- 02001735 C  00031026!3
                    PROCEDURE CROSSREFDUMP(INDEX);                                  %116- 02001740 C  00031026!3
PRT(451) = CROSSREFDUMP
                      VALUE INDEX;                                                  %116- 02001745 C  00031026!3
                      REAL  INDEX;                                                  %116- 02001750 P  00031026!3
                    BEGIN                                                           %116- 02001755 C  00031026!3
                      STREAM PROCEDURE MOVEXREFINFO(S,D,N);                         %116- 02001760 P  00031026!3
```

```
PRT(452) = MOVEXREFINFO
             VALUE N;                                                                %116-        02001765 P  0007:000010
            BEGIN                                                                    %116-        02001770 P  0007:000010
              SI := D; DI := D; DS := 8 LIT " "; DS := 7 WDS; % BLANK RECORD         %116-        02001775 C  0007:000010
              SI := S; SI := SI + 3; DI := D; DS := N CHR; % MOVE IDENTIFIER         %116-        02001780 P  0007:000210
            END OF MOVEXREFINFO;                                                     %116-        02001785 C  0007:000312



            %                                                                        %116-        02001790 P  0007:000313
            IF XREFINFO[INDEX].IDNOF ≠ 0 THEN % DUMP IT                              %116-        02001795 C  0007:000313
            BEGIN                                                                    %116-        02001800 P  0007:000713
             MOVEXREFINFO(INFO[INDEX,LINKR,INDEX,LINKC+1],XREFAY1[*],                %116-        02001805 C  0007:000810
                          TAKE(INDEX+1),[12:6]);                                     %116-        02001810 P  0007:001113
             XREFAY1[8] := XREFINFO[INDEX];                                          %116-        02001815 C  0007:001313
             XREFAY1[9] := TAKE(INDEX); % FLBAT WORD                                 %116-        02001820 P  0007:001713
             WRITE(DSK1,10,XREFAY1[*]);                                              %116-        02001821 C  0007:001912
             XREFINFO[INDEX] := 0;                                                   %116-        02001822 C  0007:002313
            END;                                                                     %116-        02001825 P  0007:002712
           END OF CROSSREFDUMP;                                                      %116-        02001830 P  0007:002712
                                                                                                7 IS   28 LONG, NEXT SEG    3



              REAL STREAM PROCEDURE CONV(ACCUM,SKP,N); VALUE SKP,N;                                02001831 T  0003:026713
PRT(453) = CONV
                  BEGIN                                                                            02001832 T  0003:026713
                      SI← ACCUM; SI←SI+SKP;SI←SI+3;DI←LOC CONV; DS ← N OCT                         02001833 T  0003:026810
                      END CONV;                                                                    02001834 T  0003:026913



            COMMENT  OCTIZE REFORMATS ACCUM FOR OCTAL CONSTANTS;                                   02001836 T  0003:027011
            BOOLEAN STREAM PROCEDURE OCTIZE(S,D,SKP,CNT); VALUE SKP,CNT;                           02001838 T  0003:027011
PRT(454) = OCTIZE
                BEGIN                                                                            02001840 T  0003:027011
                SI:=S; SI:=SI+3; DI:=D; SKP(DS:=3 RESET); % RIGHT JUSTIFY.                       02001842 T  0003:027112
                CNT(IF SC≥"8" THEN TALLY:=1 ELSE IF SC<"0" THEN TALLY:=1; SKIP 3 SB;             02001844 T  0003:027312
                    3(IF SB THEN DS:=SET ELSE DS:=RESET; SKIP SB));                              02001846 T  0003:027610
                %                                                                                02001848 T  0003:027811
                %                                                                                02001850 T  0003:027811
                OCTIZE:=TALLY; % "1" = NON OCTAL CHARACTER.                                       02001852 T  0003:027811
                END OCTIZE;                                                                      02001854 T  0003:027811



            COMMENT  HEXIZE REFORMATS ACCUM FOR HEXADECIMAL CONSTANTS;                             02001856 T  0003:027913
            BOOLEAN STREAM PROCEDURE HEXIZE(S,D,SKP,CNT); VALUE SKP,CNT;                           02001858 T  0003:027913
PRT(455) = HEXIZE
                BEGIN  LOCAL T1,T2,TEMP2,TEMP1; LABEL AGIN;                                      02001860 T  0003:027913
            COMMENT  LOCAL VARIABLES ARE LOCATED IN REVERSE ORDER FROM THE                       02001862 T  0003:028010
                     WAY THEY ARE DECLARED IN STREAM PROCEDURES;                                 02001864 T  0003:028010
                DI:=LOC TEMP1; CNT(DS:=LIT"1"); % IN CASE A CHAR=A,B,C,D,OR F.                   02001866 T  0003:028010
                SI:=S; SI:=SI+3; DI:=LOC TEMP1; % WE MAY OVERFLOW INTO TEMP2.                    02001868 T  0003:028210
```

```
                CNT(IF SC<"0" THEN IF SC≥"A" THEN IF SC≤"F" THEN % WORK HARD.          02001870 T  0003:028?:1
                     BEGIN                                                             02001872 T  0003:028?:0
                     T1:=SI; T2:=DI; DI:=T1; SI:=T2; % FLIP, MAN.                      02001874 T  0003:028?:0
                     DS:=3 RESET; SI:=T1; DI:=T2; % FLIP BACK.                         02001876 T  0003:028?:3
                     DS:=1 ADD; DI:=DI-1; SKIP 2 DB; DS:=1 SET; SKIP 3 DB;             02001878 T  0003:028??:1
                     GO AGIN;                                                          02001880 T  0003:028?:3
                     END;                                                              02001882 T  0003:029?:0
                     IF SC<"0" THEN TALLY:=1; DS:=CHR; % < 0 = NON-HEX CHARACTER.      02001884 T  0003:029?:0
           AGIN:                                                                       02001886 T  0003:029?:2
                );                                                                   02001888 T  0003:029?:2
                SI:=LOC TEMP1; DI:=D; SKP(DS:=4 RESET); % RIGHT ADJUST CONSTANT.      02001890 T  0003:029?:2
                CNT(SKIP 2 SB;                                                       02001892 T  0003:029?:2
                  4(IF SB THEN DS:=SET ELSE DS:=RESET; SKIP SB));% FINAL CONVERT.     02001894 T  0003:029?:1
                HEXIZE:=TALLY; % "1" IF PROGRAMMER GOOFED.                           02001896 T  0003:029?:1
                END HEXIZE;                                                          02001898 T  0003:029?:2



           COMMENT  PUTSEQNO PUTS THE SEQUENCE NUMBER OF THE CARD-IMAGE              02002000 T  0003:029?:0
                    CURRENTLY BEING SCANNED INTO THE INFO TABLE IN CASE              02003000 T  0003:029?:0
                    IT IS NEEDED FOR FUTURE REFERENCE;                               02004000 T  0003:029?:0
           STREAM PROCEDURE PUTSEQNO(INFO,LCR); VALUE LCR;                           02005000 T  0003:029?:0
PRT(456) = PUTSEQNO
                BEGIN  DI:=INFO; SI:=LCR; DS:=WDS;  END PUTSEQNO;                    02006000 T  0003:029?:0
```

```
           COMMENT  TURNONSTOPLIGHT TURNS THE LIGHT "RED" ON THE "CORNER".           02007000 T  0003:029?:2
                    I.E., THE PURPOSE OF THIS ROUTINE IS TO INSERT A PER-            02008000 T  0003:029?:2
                    CENT SIGN IN COLUMN 73 AS AN END OF CARD SENTINEL FOR            02009000 T  0003:029?:2
                    THE SCANNER;                                                     02010000 T  0003:029?:2
           STREAM PROCEDURE TURNONSTOPLIGHT(RED,CORNER); VALUE RED,CORNER;           02011000 T  0003:029?:2
PRT(457) = TURNONSTOPLIGHT
                BEGIN  DI:=CORNER; SI:=LOC CORNER; SI:=SI-1; DS:=CHR  END;           02012000 T  0003:029?:2



             COMMENT ADDER COMPUTES SEQUENCE NUMBERS FOR LIBRARY FUNCTIONS.          02013010 T  0003:030?:0
           IT WILL EITHER ADD THE NUMBER IN SUM TO THE NUMBER IS SEQLOC STORING       02013020 T  0003:030?:0
           THE RESULT IN SEQLOC OR SUBTRACT THE NUMBER IN SUM FROM THE               02013030 T  0003:030?:0
           NUMBER IN SEQLOC AND STORE THE RESULT IN SEQLOC,DEPENDING ON THE          02013040 T  0003:030?:0
           VARIABLE AD;                                                              02013050 T  0003:030?:0
                STREAM PROCEDURE ADDER(SUM,SEQLOC,AD,DESCRP);                        02013060 T  0003:030?:0
PRT(460) = ADDER
                    VALUE AD,DESCRP;                                                 02013065 T  0003:030?:0
                         BEGIN                                                       02013070 T  0003:030?:0
                              LOCAL HOLD,ZONEP;                                      02013073 T  0003:030?:2
                              DI←LOC ZONEP; SI←SUM; DS←8 ZON;                        02013074 T  0003:030?:2
                              COMMENT SAVED ZONE PART OF THE SEQ.NO.;                02013075 T  0003:030?:3
                              DI←SUM; DI←DI+7; DS←2 RESET;                           02013076 T  0003:030?:3
                              COMMENT HAVE ZEROED OUT SIGN VALUE OF SEQ.NO.;         02013077 T  0003:030?:1
                              SI←LOC DESCRP; SI←SI+7;                                02013078 T  0003:030?:1
                              IF SC="1" THEN BEGIN DI←LOC HOLD; SI←SEQLOC;           02013080 T  0003:030?:2
                                   DS←WDS; DI←HOLD; END                              02013085 T  0003:030?:1
```

```
                                        ELSE DI←SEQLOC;                                          02013090 T  0003↓030↓2
                        COMMENT DI IS NOW POINTING TO THE SEQNUMBER;                             02013091 T  0003↓030↓3
                            HOLD←DI; DI←DI+7; DS←2 RESET; DI←HOLD;                               02013095 T  0003↓030↓3
                  SI ← LOC AD;                                                                   02013100 T  0003↓030↓1
                  SI ← SI + 7;                                                                   02013110 T  0003↓030↓1
                IF SC = "1" THEN BEGIN SI← SUM; DS←8 ADD; END                                    02013120 T  0003↓030↓2
                            ELSE BEGIN SI← SUM; DS←8 SUB; END;                                   02013130 T  0003↓030↓1
                                    SI←LOC ZONEP; DI←HOLD; DS←8 ZON;                             02013135 T  0003↓030↓2
                                    SI←LOC ZONEP; DI←SUM; DS←8 ZON;                              02013136 T  0003↓031↓0
                                COMMENT MOVE IN ZONE PORTION TO RESULT SEQ.NO.;                   02013137 T  0003↓031↓1
                END ADDER;                                                                       02013140 T  0003↓031↓1


        COMMENT SEARCHLIB IS RESPONSIBLE FOR SEARCHING THE LIBRARY TAPES FOR                     02013150 T  0003↓031↓2
        COMPILABLE QUANTITIES. THE PARAMETER INDICATES THAT WE ARE ENTERING                      02013155 T  0003↓031↓2
        A LIBRAY CALL IF TRUE, ELSE WE ARE EXITING.;                                             02013160 T  0003↓031↓2
        PROCEDURE SEARCHLIB(DOLLAR); VALUE DOLLAR; BOOLEAN DOLLAR;                               02013165 T  0003↓031↓2
PRT(461) = SEARCHLIB
            BEGIN
            LABEL  EXIT,EXITOUT, NOPARTIAL;                                                      02013170 T  0003↓031↓2
                                                                                                 02013175 T  0003↓031↓2
                                                                                        START OF SEGMENT **********    8
            PROCEDURE FLAGIT(N); VALUE N; INTEGER N;                                             02013176 T  0008↓000↓0
PRT(462) = FLAGIT
                BEGIN
                BOOLEAN TL,TS;                                                                   02013177 T  0008↓000↓0
                                                                                                 02013178 T  0008↓000↓0
                                                                                        START OF SEGMENT **********    9
STACK(F+2) = TL
STACK(F+3) = TS
                TL:=LISTOG; TS:=SINGLTOG; LISTOG:=FALSE; SINGLTOG:=FALSE;                        02013179 T  0009↓000↓0
                Q:=ACCUM[1]; FLAG(N);                                                            02013180 T  0009↓000↓0
                LISTOG:=TL; SINGLTOG:=TS;                                                        02013181 T  0009↓000↓3
                END FLAGIT;                                                                      02013183 T  0009↓001↓2
                                                                                        9 IS   14 LONG, NEXT SEG    8


  IF DOLLAR THEN                                                                                 02013184 T  0008↓000↓0
    BEGIN COMMENT WE ARE ON A DOUBLE DOLLAR CARD;                                                02013185 T  0008↓000↓0
RESULT←COUNT←ACCUM[1] ←0; SCANNER;                                                               02013190 T  0008↓000↓0
RESULT←COUNT←ACCUM[1] ←0; SCANNER;                                                               02013195 T  0008↓000↓1
      IF ACCUM[1] > "1←0000" AND ACCUM[1] < "1D0000" THEN                                        02013200 T  0008↓000↓3
         FILEINX:=ACCUM[1].[21:3] ELSE BEGIN                                                     02013205 T  0008↓000↓0
   COMMENT ERROR 500 - ILLEGAL LIBRARY NAME;                                                     02013210 T  0008↓000↓1
          FLAGIT(500); GO EXIT;                                                                  02013219 T  0008↓001↓2
          END;                                                                                   02013222 T  0008↓001↓2
      FILEINX ← FILEINX -1;                                                                      02013225 T  0008↓001↓0
     'IF DIRECTORY[GT1←3×FILEINX,0]=0 THEN                                                       02013230 T  0008↓001↓0
      BEGIN COMMENT MUST READ DIRECTORY;                                                         02013235 T  0008↓001↓3
          GT3←MKABS(LIBRARY[FILEINX](0));                                                        02013240 T  0008↓001↓0
          MOVE(56,LIBRARY[FILEINX](0),DIRECTORY[GT1,0]);                                         02013245 T  0008↓001↓1
      GT2←DIRECTORY[GT1,0]; DIRECTORY[FILEINX×3,0] ← -2;                                         02013250 T  0008↓002↓3
          WHILE GT2 ← GT2-1 > 0 DO                                                               02013255 T  0008↓002↓2
          BEGIN                                                                                  02013260 T  0008↓003↓3
            READ(LIBRARY[FILEINX]);                                                              02013265 T  0008↓003↓0
                                                                                                 02013270 T  0008↓003↓0
```

```
          MOVE(56,LIBRARY[FILEINX](0),DIRECTORY[GT1+GT1+1,0]);          02013275 T  0008:004011
        END;                                                            02013280 T  0008:004712
      END;                                                              02013285 T  0008:004713
    RESULT:=ACCUM[1]:= COUNT:=0; SCANNER; COMMENT GET THE PROG.ID.;     02013290 T  0008:004713
    IF LOOK(ACCUM[1],DIRECTORY,3*FILEINX,   GT1,GT2) THEN               02013295 T  0008:005010
      BEGIN COMMENT ERROR 501 = ITEM NOT IN DIRECTORY;                  02013300 T  0008:005411
        FLAGIT(501); GO EXIT;                                           02013305 T  0008:005512
      END;                                                              02013310 T  0008:005610
    WHILE LCR.[33:15] = NCR.[33:15] # 1 OR NCR.[30:3] # 7               02013313 T  0008:005610
    DO BEGIN                                                            02013315 T  0008:006010
        IF EXAMIN(NCR) = "[" THEN GO TO EXITOUT    ;                    02013317 T  0008:006112
            RESULT:=5; SCANNER;                                         02013318 T  0008:006313
    END;                                                                02013319 T  0008:006411
    GO TO NOPARTIAL;                                                    02013320 T  0008:006512
EXITOUT: BEGIN COMMENT WE HAVE A PARTIAL LIBRARY OPERATION;             02013325 T  0008:006513
      RESULT:= ACCUM[1]:= COUNT:=0; SCANNER; COMMENT SPACE PAST "[" ;   02013330 T  0008:006610
      RESULT:= ACCUM[1]:= COUNT:=0; SCANNER; COMMENT GET START POINT;   02013335 T  0008:006811
      IF RESULT # 3 THEN                                                02013340 T  0008:007113
        BEGIN COMMENT ERROR 502 = IMPROPER START POINT;                 02013345 T  0008:007210
        FLAGIT(502); GO EXIT;                                           02013350 T  0008:007211
        END;                                                            02013355 T  0008:007410
      GT1 := GT1 + CONV(ACCUM[1],0,ACCUM[1],[12:6]) = 1;                02013360 T  0008:007410
      RESULT:= ACCUM[1]:= COUNT:=0; SCANNER;                            02013365 T  0008:007810
      IF RESULT # 2 THEN                                                02013370 T  0008:008112
        BEGIN COMMENT ERROR 503 = NO SEPERATOR;                         02013375 T  0008:008113
        FLAGIT(503); GO EXIT;                                           02013380 T  0008:008210
        END;                                                            02013385 T  0008:008313
      RESULT:= ACCUM[1]:= COUNT:=0; SCANNER; COMMENT GET LENGTH;        02013390 T  0008:008313
      IF RESULT # 3 THEN                                                02013395 T  0008:008610
        BEGIN COMMENT ERROR 504 = IMPROPER LENGTH;                      02013400 T  0008:008712
        FLAGIT(504); GO EXIT;                                           02013405 T  0008:008713
        END;                                                            02013410 T  0008:008811
      GT2 := GT1 + CONV(ACCUM[1],0,ACCUM[1],[12:6]);                    02013410 T  0008:008811
      RESULT:= ACCUM[1]:= COUNT:=0; SCANNER;                            02013420 T  0008:009211
      IF ACCUM[1] # "1]0000" THEN                                       02013425 T  0008:009512
        BEGIN COMMENT ERROR 505 = NO RIGHT BRACKET;                     02013430 T  0008:009610
        FLAGIT(505); GO EXIT;                                           02013435 T  0008:009611
        END;                                                            02013440 T  0008:009912
      WHILE LCR.[33:15] = NCR.[33:15] # 1 OR NCR.[30:3] # 7             02013445 T  0008:009912
        DO BEGIN RESULT := 5; SCANNER END;                              02013446 T  0008:010210
      END;                                                              02013450 T  0008:010512
NOPARTIAL: COMMENT NOW SET UP THE LINKS;                                02013475 T  0008:010512
   LIBARRAY[LIBINDEX].LSTUSD := LASTUSED;                               02013480 T  0008:010512
     LIBARRAY[LIBINDEX].FILEINDEX := FILEINX;                           02013490 T  0008:010713
   LIBARRAY[LIBINDEX].STOPPOINT := FINISHPT;                            02013495 T  0008:011010
    LIBARRAY[LIBINDEX].NEXTENTRY := RECOUNT-1;                          02013497 T  0008:011211
    FINISHPT := GT2;                                                    02013500 T  0008:011513
    IF LIBINDEX>0 THEN DIRECTORY[(LIBARRAY[LIBINDEX-3],FILEINDEX)       02013505 T  0008:011610
    *3,0] := RECOUNT -1;                                                02013510 T  0008:011810
    RECOUNT:=GT1;                                                       02013515 T  0008:012113
    IF EXAMIN(LCR) # "%" THEN                                           02013516 T  0008:012211
     PUTSEQNO(INFO[LASTSEQROW,LASTSEQUENCE],LCR);                       02013517 T  0008:012410
 MOVE(1,INFO[LASTSEQROW,LASTSEQUENCE],LIBARRAY[LIBINDEX+1]);            02013520 T  0008:012611
 MOVE(1,INFO[LASTSEQROW,LASTSEQUENCE],SEQSUM);                          02013525 T  0008:013010
    IF LASTUSED≤2 OR LASTUSED=5 THEN GTI1:=0                            02013526 T  0008:013211
        ELSE IF MAXLTLCR.[33:15]-NCR.[33:15]<11 THEN                    02013527 T  0008:013411
    GTI1:=MKABS(LIBRARY[FILEINX](0)) ELSE GTI1:=(NCR+2).[33:15];        02013528 T  0008:013810
```

```
    LIBARRAY[LIBINDEX+2].NCRLINK←GTI1.[33:15]; COMMENT GTI1=NCR;          02013530 T  0008:0145:3
    IF LASTUSED≤2 OR LASTUSED=5 THEN                                      02013533 T  0008:0148:1
    LIBARRAY[LIBINDEX+2].LCRLINK←0 ELSE                                   02013534 T  0008:0150:1
        LIBARRAY[LIBINDEX+2].LCRLINK←GTI1.[33:15]+10;                     02013535 T  0008:0153:3
    IF LIBINDEX> 0 THEN IF CARDCALL THEN BEGIN                            02013536 T  0008:0157:3
          LASTUSED←5; LIBARRAY[LIBINDEX].NEXTENTRY←                       02013537 T  0008:0159:3
                      LIBARRAY[LIBINDEX].NEXTENTRY -1;                    02013538 T  0008:0161:2
          END ELSE BEGIN                                                  02013539 T  0008:0164:0
          LASTUSED←6; FIRSTIMEX← TRUE; END                                02013540 T  0008:0164:1
ELSE BEGIN IF LASTUSED=3 THEN FIRSTIMEX:=TRUE; LASTUSED:=5; END;          02013541 T  0008:0166:0
    LIBINDEX ← LIBINDEX + 3;                                              02013542 T  0008:0169:2
 END                                                                      02013545 T  0008:0170:1
    ELSE                                                                  02013550 T  0008:0170:1
 BEGIN COMMENT WE DID NOT COME FROM DOUBLE DOLLAR SO UNLINK;              02013555 T  0008:0170:1
    LIBINDEX ← LIBINDEX -3;                                               02013560 T  0008:0171:2
    RECOUNT←RECOUNT-1;                                                    02013563 T  0008:0172:0
    LASTUSED ← LIBARRAY[LIBINDEX].LSTUSD;                                 02013565 T  0008:0173:3
    IF LASTUSED = 1 THEN MEDIUM := "C ";                          %112-   02013566 P  0008:0175:2
    IF LIBINDEX > 0 THEN BEGIN GTI1←LIBARRAY[LIBINDEX].NEXTENTRY;         02013567 T  0008:0177:2
        DIRECTORY[FILEINX×3,0]←RECOUNT; RECOUNT←GTI1+2; END ELSE           02013568 T  0008:0179:3
    DIRECTORY[FILEINX×3,0] ← RECOUNT;                                     02013570 T  0008:0183:3
    IF LIBINDEX > 0 THEN                                                  02013575 T  0008:0187:3
        FILEINX ← LIBARRAY[LIBINDEX -3].FILEINDEX;                        02013580 T  0008:0188:0
    FINISHPT ← LIBARRAY[LIBINDEX].STOPPOINT;                              02013600 T  0008:0190:1
    IF LIBINDEX ≠ 0 THEN                                                  02013610 T  0008:0192:0
        MOVE(1,LIBARRAY[LIBINDEX -3 +1], SEQSUM);                         02013615 T  0008:0193:2
    IF LASTUSED≤2 OR LASTUSED=5 THEN  NCR:=MKABS(CBUFF[0]) ELSE           02013617 T  0008:0196:0
    NCR ← LIBARRAY[LIBINDEX+2].NCRLINK;                                   02013620 T  0008:0200:1
    IF LASTUSED≤2 OR LASTUSED=5 THEN  LCR:=MKABS(CBUFF[9]) ELSE           02013621 T  0008:0203:2
       LCR ← LIBARRAY[LIBINDEX+2].LCRLINK;                                02013625 T  0008:0207:3
       NORELEASE←TRUE;                                                    02013627 T  0008:0210:0
       IF LASTUSED=6 THEN FIRSTIMEX←TRUE;                                 02013628 T  0008:0210:1
    END OF UNLINK;                                                        02013630 T  0008:0212:1
    IF LIBINDEX = 0 THEN                                                  02013635 T  0008:0212:1
       BEGIN COMMENT GOING BACK TO OUTSIDE WORLD;                         02013640 T  0008:0213:3
       SEQSUM←0;                                                          02013645 T  0008:0214:0
       END                                                                02013650 T  0008:0214:1
    ELSE                                                                  02013655 T  0008:0214:1
       BEGIN                                                              02013660 T  0008:0214:1
       GT1←(GTI1←(DIRECTORY[FILEINX×3,0]+3)/5)×5+1;                       02013665 T  0008:0215:2
       GT2←(GTI1←(RECOUNT-3)/5)×5+1;                                      02013670 T  0008:0220:0
       GT3 ←(GT2 - GT1)DIV 5;                                             02013675 T  0008:0223:2
       SPACE(LIBRARY[FILEINX],GT3);                                       02013680 T  0008:0225:2
PRT(463) = INPUT(W)
                                                                          02013681 T  0008:0226:1
                                                                          02013682 T  0008:0228:1
       READ(LIBRARY[FILEINX]);                                            02013685 T  0008:0228:1
                                                                          02013690 T  0008:0232:1
                                                                          02013693 T  0008:0232:1
                                                                          02013695 T  0008:0232:1
       MOVE(1,LIBRARY[FILEINX](0),GTI1);                                  02013697 T  0008:0232:1
       IF GTI1≠GT2 AND GTI1 ≠ 0 THEN                                      02013699 T  0008:0237:2
          BEGIN COMMENT ERROR 507 MEANS TAPE POSITIONING ERROR;           02013701 T  0008:0239:2
             FLAG(507);   GO TO EXIT;                                     02013702 T  0008:0239:3
          END;                                                            02013703 T  0008:0240:1
    LTLCR←MKABS(LIBRARY[FILEINX](10))+(GTI1←(((RECOUNT-1) MOD 5) ×11));   02013705 T  0008:0240:1
       MAXLTLCR←MKABS(LIBRARY[FILEINX](0))+54;                            02013710 T  0008:0248:0
```

```
                    ADDER(SEQSUM,LTLCR,TRUE,TRUE);                                  02013713 T  0008:025:2
                    IF LASTUSED= 6 THEN BEGIN                                       02013714 T  0008:025:2
                    NCR←LCR←MKABS(LIBRARY[FILEINX](0));                             02013715 T  0008:025:1
                    PUTSEQNO(GT1,LCR);                                              02013716 T  0008:026:3
                    TURNONSTOPLIGHT("%",LCR);                                       02013717 T  0008:026:1
                    END;        END;                                                02013718 T  0008:026:3
                EXIT: END SEARCHLIB;                                                02013720 T  0008:026:3
                                                                                  8 IS  265 LONG, NEXT SEG    3


            COMMENT WRITNEW TRANSFERS THE CARD IMAGE TO THE NEWTAPE BUFFER          02014000 T  0003:031:2
                    AND REPORTS IF THE CARD MIGHT BE CONTROL CARD;                  02015000 T  0003:031:2
        BOOLEAN STREAM PROCEDURE WRITNEW(NEW,FCR); VALUE FCR;                       02016000 T  0003:031:2
PRT(464) = WRITNEW
            BEGIN SI←FCR; IF SC≠"$" THEN TALLY←1;                                   02017000 T  0003:031:2
                    DI←NEW; DS←10 WDS; WRITNEW←TALLY                                02018000 T  0003:032:0
                END WRITNEW;                                                        02019000 T  0003:032:1


                                                                                    02020000 T  0003:031:3
                                                                                    02021000 T  0003:031:3
                                                                                    02022000 T  0003:031:3
                                                                                    02023000 T  0003:031:3
                                                                                    02041000 T  0003:031:3
                                                                                    02042000 T  0003:031:3
                                                                                    02043000 T  0003:031:3
                                                                                    02044000 T  0003:031:3
                                                                                    02045000 T  0003:031:3
                                                                                    02046000 T  0003:031:3
                                                                                    02047000 T  0003:031:3
                                                                                    02047050 T  0003:031:3
                                                                                    02047055 T  0003:031:3
                                                                                    02047060 T  0003:031:3
                                                                                    02047065 T  0003:031:3
                                                                                    02047070 T  0003:031:3
                                                                                    02047075 T  0003:031:3
                                                                                    02048000 T  0003:031:3
                                                                                    02049000 T  0003:031:3
                                                                                    02050000 T  0003:031:3
                                                                                    02051000 T  0003:031:3
                                                                                    02052000 T  0003:031:3
                                                                                    02053000 T  0003:031:3
                                                                                    02054000 T  0003:031:3
                                                                                    02055000 T  0003:031:3
                                                                                    02055100 T  0003:031:3
                                                                                    02055200 T  0003:031:3
                                                                                    02055300 T  0003:031:3
                                                                                    02056000 T  0003:031:3
                                                                                    02057000 T  0003:031:3
                                                                                    02058000 T  0003:031:3
                                                                                    02059000 T  0003:031:3
                                                                                    02060000 T  0003:031:3
COMMENT  EQUAL COMPARES COUNT CHARACTERS LOCATED AT A AND B FOR                     02061000 T  0003:031:3
```

```
                        EQUALITY.  THIS ROUTINE IS USED IN THE LOOK-UP OF ALPHA                  02061500 T  0003:031313
                        QUANTITIES IN THE DIRECTORY;                                            02062000 T  0003:031313
                    BOOLEAN STREAM PROCEDURE EQUAL(COUNT,A,B); VALUE COUNT;                     02062500 T  0003:031313
PRT(465) = EQUAL
                        BEGIN                                                                   02063000 T  0003:031313
                        TALLY:=1; S1:=A; DI:=B;                                                 02063500 T  0003:031410
                        IF COUNT SC=DC THEN EQUAL:=TALLY                                        02064000 T  0003:031411
                        END EQUAL;                                                              02064500 T  0003:031513



                    PROCEDURE READACARD;  FORWARD;                                              02065000 T  0003:031611
PRT(466) = READACARD
                    PROCEDURE DOLLARCARD; FORWARD;                                              02065500 T  0003:031611
PRT(467) = DOLLARCARD
                    BOOLEAN PROCEDURE BOOLEXP; FORWARD;                                         02065600 T  0003:031611
PRT(470) = BOOLEXP
                    PROCEDURE SCANNER;                                                          02066000 T  0003:031611
                        BEGIN                                                                   02066500 T  0003:031611
                    COMMENT  "SCAN" IS THE STREAM PROCEDURE WHICH DOES THE ACTUAL SCANNING.     02067000 T  0003:031611
                             IT IS DRIVEN BY A SMALL WORD MODE PROCEDURE CALLED "SCANNER",      02067500 T  0003:031611
                             WHICH CHECKS FOR A QUANTITY BEING BROKEN ACROSS A CARD. "SCAN"     02068000 T  0003:031611
                             IS CONTROLLED BY A VARIABLE CALLED "RESULT". "SCAN" ALSO           02068500 T  0003:031611
                             INFORMS THE WORLD OF ITS ACTION BY MEANS OF THE SAME VARIABLE.     02069000 T  0003:031611
                             HENCE THE VARIABLE "RESULT" IS PASSED BY BOTH NAME AND VALUE.      02069500 T  0003:031611
                             THE MEANING OF "RESULT" AS INPUT IS:                               02070000 T  0003:031611
                                VALUE    MEANING                                                02070500 T  0003:031611
                                =====    ========================================               02071000 T  0003:031611
                                  0      INITIAL CODE - DEBLANK AND START TO FETCH THE          02071500 T  0003:031611
                                         NEXT QUANTITY.                                         02072000 T  0003:031611
                                  1      CONTINUE BUILDING AN IDENTIFIER (INTERRUPTED BY        02072500 T  0003:031611
                                         END-OF-CARD BREAK).                                    02073000 T  0003:031611
                                  2      LAST QUANTITY BUILT WAS SPECIAL CHARACTER. HENCE,      02073500 T  0003:031611
                                         EXIT (INTERRUPTION BY END-OF-CARD BREAK IS NOT         02074000 T  0003:031611
                                         IMPORTANT).                                            02074500 T  0003:031611
                                  3      CONTINUE BUILDING A NUMBER (INTERRUPTED BY END-OF-     02075000 T  0003:031611
                                         CARD BREAK).                                           02075500 T  0003:031611
                                  4      LAST THING WAS AN ERROR (COUNT EXCEEDED 63). HENCE.    02076000 T  0003:031611
                                         EXIT (INTERRUPTION BY END-OF-CARD BREAK NOT            02076500 T  0003:031611
                                         IMPORTANT).                                            02077000 T  0003:031611
                                  5      GET NEXT CHARACTER AND EXIT.                           02077500 T  0003:031611
                                  6      SCAN A COMMENT.                                        02078000 T  0003:031611
                                  7      DEBLANK ONLY.                                          02078500 T  0003:031611
                             THE MEANING OF "RESULT" AS OUTPUT IS:                              02079000 T  0003:031611
                                VALUE    MEANING                                                02079500 T  0003:031611
                                =====    ========================================               02080000 T  0003:031611
                                  1      AN IDENTIFIER WAS BUILT.                               02080500 T  0003:031611
                                  2      A SPECIAL CHARACTER WAS OBTAINED.                      02081000 T  0003:031611
                                  3      A NUMBER (INTEGER) WAS BUILT.                          02081500 T  0003:031611
                             "SCAN" PUTS ALL STUFF SCANNED (EXCEPT FOR COMMENTS AND             02082000 T  0003:031611
                             DISCARDED BLANKS) INTO "ACCUM" (CALLED "ACCUMULATOR"               02082500 T  0003:031611
                             FOR THE REST OF THIS DISCUSSION).                                  02083000 T  0003:031611
                             "COUNT" IS THE VARIABLE THAT GIVES THE NUMBER OF CHARACTERS        02083500 T  0003:031611
                             "SCAN" HAS PUT INTO THE "ACCUMULATOR".  SINCE "SCAN" NEEDS         02084000 T  0003:031611
                             THE VALUE SO THAT IT CAN PUT MORE CHARACTERS INTO THE "ACCUM-      02084500 T  0003:031611
                             ULATOR" AND NEEDS TO UPDATE "COUNT" FOR THE OUTSIDE WORLD.         02085000 T  0003:031611
```

```
                    "COUNT" IS PASSED BY BOTH NAME AND VALUE.  IT IS ALSO                    02085500 T  0003:031611
                    CONVENIENT TO HAVE (63-COUNT). THIS IS CALLED "COMCOUNT".                02086000 T  0003:031611
                    "NCR" (NEXT CHARACTER TO BE SCANNED) IS ALSO PASSED BY                   02086500 T  0003:031611
                    NAME AND VALUE SO THAT IT MAY BE UPDATED.                                02087000 T  0003:031611
                    "ST1" AND "ST2" ARE TEMPORARY STORAGES WHICH ARE EXPLICITLY              02087500 T  0003:031611
                    PASSED TO "SCAN" IN ORDER TO OBTAIN THE MOST USEFULL STACK               02088000 T  0003:031611
                    ARRANGEMENT.                                                             02088500 T  0003:031611
                    ;                                                                        02089000 T  0003:031611
                STREAM PROCEDURE SCAN(NCR,COUNTV,ACCUM,COMCOUNT,RESULT,RESULTV,              02089500 T  0003:031611
PRT(471) = SCAN                                                                    START OF SEGMENT **********   10
                                     COUNT,ST2,NCRV,ST1);                                    02090000 T  0010:000010
                                VALUE COUNTV, COMCOUNT,RESULTV,ST2,NCRV,ST1;                 02090500 T  0010:000010
                    BEGIN                                                                    02091000 T  0010:000010
                    LABEL DEBLANK,NUMBERS,IDBLDR,GNC,K,EXIT,FINIS,L,ERROR,                   02091500 T  0010:000010
                          COMMENTS,COMMANTS;                                                 02092000 T  0010:000010
                    DI:=RESULT; DI:=DI+7; SI:=NCRV;                                          02092500 T  0010:000010
            COMMENT SETUP "DI" FOR A CHANGE IN "RESULT" AND "SI" FOR A LOOK AT               02093000 T  0010:000011
                    THE BUFFER;                                                              02093500 T  0010:000011
                    CI:=CI+RESULTV;    % SWITCH ON VALUE OF RESULT;                         02094000 T  0010:000011
                    GO DEBLANK;  % 0 IS INITIAL CODE.                                        02094500 T  0010:000112
                    GO IDBLDR;   % 1 IS ID CODE.                                           02095000 T  0010:000113
                    GO FINIS;    % 2 IS SPECIAL CHARACTER CODE.                            02095500 T  0010:000113
                    GO NUMBERS;  % 3 IS NUMBER CODE.                                       02096000 T  0010:000210
                    GO FINIS;    % 4 IS ERROR CODE.                                        02096500 T  0010:000210
                    GO GNC;      % 5 IS GET NEXT CHARACTER CODE.                           02097000 T  0010:000211
                    GO COMMANTS; % 6 IS COMMENT CODE.                                      02097500 T  0010:000211
                                 % 7 IS DEBLANK ONLY CODE.                                 02098000 T  0010:000312
                    IF SC=" " THEN                                                           02098500 T  0010:000312
                K:       BEGIN  SI:=SI+1; IF SC=" " THEN GO K  END;                          02099000 T  0010:000313
                    GO FINIS;                                                                02099500 T  0010:000512
                DEBLANK:                                                                     02100000 T  0010:000512
                    IF SC=" " THEN                                                           02100500 T  0010:000512
                L:       BEGIN  SI:=SI+1; IF SC=" " THEN GO L  END;                          02101000 T  0010:000611
            COMMENT IF WE ARRIVE HERE WE HAVE A NON-BLANK CHARACTER;                         02101500 T  0010:000810
                    NCRV:=SI;                                                                02102000 T  0010:000810
                    IF SC ≥ "0" THEN GO NUMBERS;                                             02102500 T  0010:000810
                    IF SC=ALPHA THEN GO IDBLDR;                                              02103000 T  0010:000912
            COMMENT IF WE ARRIVE HERE WE HAVE A SPECIAL CHARACTER (OR GNC);                  02103500 T  0010:000913
                GNC:                                                                         02104000 T  0010:000913
                    DS:=LIT"2"; TALLY:=1; SI:=SI+1; GO EXIT;                                 02104500 T  0010:000913
                COMMANTS:                                                                    02105000 T  0010:001112
                    IF SC≠";" THEN                                                           02105500 T  0010:001112
                         BEGIN                                                               02106000 T  0010:001211
                COMMENTS:                                                                    02106500 T  0010:001211
                         SI:=SI+1;                                                           02107000 T  0010:001312
                         IF SC > "%" THEN GO COMMENTS;                                       02107500 T  0010:001312
                         IF SC < ";" THEN GO COMMENTS;                                       02108000 T  0010:001410
            COMMENT CHARACTERS BETWEEN % AND SEMICOLON ARE HANDLED BY WORD-                  02108500 T  0010:001411
                    MODE PART OF COMMENT ROUTINE;                                            02109000 T  0010:001411
                         END;                                                                02109500 T  0010:001411
                    GO FINIS;                                                                02110000 T  0010:001411
                IDBLDR:                                                                      02110500 T  0010:001512
                    TALLY:=63; DS:=LIT "1";                                                  02111000 T  0010:001512
                    COMCOUNT(TALLY:=TALLY+1;                                                 02111500 T  0010:001513
                             IF SC=ALPHA THEN SI:=SI+1 ELSE JUMP OUT TO EXIT);               02112000 T  0010:001712
                    TALLY:=TALLY+1;                                                          02112500 T  0010:001811
```

```
                    IF SC=ALPHA THEN                                                    02113000 T  0010:0019:2
                         BEGIN                                                          02113500 T  0010:0019:3
               ERROR:                                                                   02114000 T  0010:0019:3
                         DI:=DI-1; DS:=LIT "4"; GO EXIT;                                02114500 T  0010:0020:0
                         END                                                            02115000 T  0010:0021:2
                    ELSE GO EXIT;                                                       02115500 T  0010:0021:2
               COMMENT  IF WE ARRIVE AT ERROR WE HAVE MORE THAN 63 CHARACTERS           02116000 T  0010:0021:3
                        IN AN IDENTIFIER OR NUMBER;                                     02116500 T  0010:0021:3
               NUMBERS:                                                                 02117000 T  0010:0021:3
                    TALLY:=63; DS:=LIT "3";                                             02117500 T  0010:0021:3
PRT(472) = *LIST, LABEL, OR SEGMENT DESCRIPTOR*
                    COMCOUNT(TALLY:=TALLY+1;                                            02118000 T  0010:0022:1
                             IF SC <"0"THEN JUMP OUT TO EXIT; SI:=SI+1);                02118500 T  0010:0024:0
                    GO ERROR;                                                           02119000 T  0010:0025:3
               EXIT:                                                                    02119500 T  0010:0025:3
                    ST1:=TALLY; % "ST1" CONTAINS NUMBER OF CHARACTERS WE ARE            02120000 T  0010:0025:3
                                % GOING TO MOVE INTO THE "ACCUMULATOR".                 02120500 T  0010:0026:0
                    TALLY:=TALLY+COUNTV; ST2:=TALLY;                                    02121000 T  0010:0026:0
                    DI:=COUNT; SI:=LOC ST2; DS:=WDS;                                    02121500 T  0010:0027:2
               COMMENT  THIS CODE UPDATED "COUNT";                                      02122000 T  0010:0027:3
                    DI:=ACCUM; SI:=SI-3; DS:=3 CHR;                                     02122500 T  0010:0027:3
               COMMENT  THIS CODE PLACES "COUNT" IN "ACCUM" AS WELL;                    02123000 T  0010:0028:1
                    DI:=DI+COUNTV; % POSITION "DI" PAST CHARACTERS ALREADY              02123500 T  0010:0028:1
                                   % IN THE "ACCUMULATOR", IF ANY.                      02124000 T  0010:0029:2
                    SI:=NCRV; DS:=ST1 CHR;                                              02124500 T  0010:0029:2
               COMMENT  MOVE CHARACTERS INTO "ACCUM";                                   02125000 T  0010:0029:3
               FINIS:                                                                   02125500 T  0010:0029:3
                    DI:=NCR; ST1:=SI; SI:=LOC ST1; DS:=WDS;                             02126000 T  0010:0029:3
PRT(473) = *LIST, LABEL, OR SEGMENT DESCRIPTOR*
               COMMENT  RESET "NCR" TO LOCATION OF NEXT CHARACTER TO BE SCANNED;        02126500 T  0010:0031:2
                    END OF SCAN;                                                        02127000 T  0010:0031:2


    LABEL L;%                                                                           02127500 T  0010:0031:2
L:                                                                                      02128000 T  0010:0031:2
    SCAN(NCR,COUNT,ACCUM[1],63-COUNT,RESULT,                                            02128500 T  0010:0032:0
         RESULT,COUNT,0,NCR,0);                                                         02129000 T  0010:0034:1
    IF NCR=LCR THEN                                                                     02129500 T  0010:0036:0
         BEGIN                                                                          02130000 T  0010:0037:2
         READACARD;                                                                     02130500 T  0010:0037:3
         IF LIBINDEX#0 THEN                                                             02131500 T  0010:0038:0
              IF RECOUNT=FINISHPT THEN                                                  02132000 T  0010:0038:1
                 BEGIN                                                                  02132500 T  0010:0040:0
                 SEARCHLIB(FALSE);                                                      02133000 T  0010:0040:1
                 READACARD;                                                             02133500 T  0010:0041:2
                 NORELEASE:=FALSE;                                                      02134000 T  0010:0041:3
                 END;                                                                   02134500 T  0010:0042:1
         GO TO L; % GO DIRECTLY TO L, DO NOT PASS GO,                                   02135500 T  0010:0042:1
                  % DO NOT COLLECT $200,                                                02136000 T  0010:0043:2
         END;                                                                           02136500 T  0010:0043:2
    END SCANNER;                                                                        02137000 T  0010:0043:2
                                                                                  10 IS   44 LONG, NEXT SEG    3


DEFINE WRITELINE = IF SINGLTOG THEN WRITE(LINE,15,LIN[*])                               02181000 T  0003:0316:1
```

```
                                ELSE WRITE(LINE[DBL],15,LIN[*])#,                                    02181250 T  0003:031611
                    MAKCAST     = BEGIN                                                              02181500 T  0003:031611
                                  CARDCALL:=IF LASTUSED=5 THEN TRUE ELSE FALSE;                      02181750 T  0003:031611
                                  SEARCHLIB(TRUE);                                                   02182000 T  0003:031611
                                  END #,                                                             02182250 T  0003:031611
                    PRINTCARD   = BEGIN                                                              02182500 T  0003:031611
                                  EDITLINE(LIN,FCR,L,[36:10],                          %114-       02182750 P  0003:031611
                                           SGNO,L,[45:2],MEDIUM,OMITTING);             %114-       02182760 C  0003:031611
                                  IF NOHEADING THEN DATIME; WRITELINE;                               02183000 T  0003:031611
                                  END #;                                                             02183250 T  0003:031611
                STREAM PROCEDURE EDITLINE(LINE,NGR,R,S,L,SYMBOL,OMIT);                   %114-       02183500 P  0003:031611
PRT(474) = EDITLINE
                VALUE NCR,R,S,L,SYMBOL,OMIT;                                             %114-       02183750 P  0003:031611
                 BEGIN                                                                               02184000 T  0003:031611
                 DI := LINE; DS := 16 LIT " ";                                                       02184250 T  0003:031712
                 SI := NCR; DS := 9 WDS;                                                             02184500 T  0003:031913
                 DS := 8 LIT " ";                                                                    02184750 T  0003:032010
                 DS := WDS; % SEQUENCE NUMBER.                                                       02185000 T  0003:032112
                 DS:=LIT" "; SI:=LOC SYMBOL; SI:=SI+6;                                               02185250 T  0003:032113
                 DS:=2 CHR; DS:=LIT" ";                                                              02185500 T  0003:032211
                 SI:=LOC R; SI:=SI+4;                                                                02185750 T  0003:032312
                 IF SC=" " THEN DS:=12 LIT" " ELSE                                       %114-       02186000 P  0003:032313
                     BEGIN                                                               %114-       02186250 P  0003:032610
                     SI:=LOC S; DS:=4 DEC; DS:=LIT ":";                                  %114-       02186300 C  0003:032610
                     SI:=LOC R; DS:=4 DEC; DS:=LIT ":";                                  %114-       02186400 C  0003:032712
                     SI:=LOC L; DS:=1 DEC; DS:=LIT " ";                                  %114-       02186500 P  0003:032810
                     END;                                                                %114-       02186600 C  0003:032912
                 OMIT(DI := DI - 12; DS := 12 LIT "  !OMIT:    "; DI := LINE;          %114-       02186750 P  0003:032912
                      DS := 8 LIT "  !OMIT;");                                           %114-       02186760 C  0003:033211
                END EDITLINE;                                                                        02187000 T  0003:033410



            COMMENT  COMPARE COMPARES SEQUENCE NUMBERS OF TAPE AND CARD. IF                          02187250 T  0003:033410
                     TAPE IS SMALLER THEN RESULT = 0 ELSE IF CARD IS SMALLER                         02187500 T  0003:033410
                     RESULT = 1 ELSE RESULT = 2;                                                     02187750 T  0003:033410
            REAL STREAM PROCEDURE COMPARE(TAPE,CARD); VALUE TAPE,CARD;                               02188000 T  0003:033410
PRT(475) = COMPARE
                BEGIN                                                                                02188250 T  0003:033410
                SI := TAPE; DI := CARD;                                                              02188500 T  0003:033512
                IF 8 SC ≥ DC THEN                                                                    02188750 T  0003:033513
                    BEGIN                                                                            02189000 T  0003:033610
                    SI := SI-8; DI := DI-8; TALLY := 1;                                              02189250 T  0003:033610
                    IF 8 SC = DC THEN TALLY := 2                                                     02189500 T  0003:033712
                    END;                                                                             02189750 T  0003:033713
                COMPARE := TALLY                                                                     02190000 T  0003:033810
                END COMPARE;                                                                         02190250 T  0003:033810



            PROCEDURE OUTPUTSOURCE;                                                                  02190500 T  0003:033912
PRT(476) = OUTPUTSOURCE
                BEGIN                                                                                02190750 T  0003:033912
                LABEL LCARD,LTAPE,AWAY;                                                              02191000 T  0003:033912
                                                                                  START OF SEGMENT ********** 11
```

```
            SWITCH SW:=LCARD,LCARD,LTAPE,AWAY,LCARD,LTAPE;                                  02191250 T  0011:0000:0
            IF SEQTOG THEN % RESEQUENCING,                                                  02191500 T  0011:0006:0
                 BEGIN                                                                      02191750 T  0011:0007:3
                 IF TOTALNO = -10 OR NEWBASE THEN                                           02192000 T  0011:0008:0
                      BEGIN                                                                 02192250 T  0011:0009:3
                      NEWBASE := FALSE; GTI1:= TOTALNO:=BASENUM                             02192500 T  0011:0010:0
                      END                                                                   02192750 T  0011:0011:2
                 ELSE GTI1:= TOTALNO:= TOTALNO + ADDVALUE;                                  02193000 T  0011:0012:0
                 CHANGESEQ(GTI1,LCR);                                                       02193250 T  0011:0014:1
                 END;                                                                       02193500 T  0011:0015:3
            IF NEWTOG THEN                                                                  02193750 T  0011:0015:3
            IF INSERTDEPTH > 0 AND INSERTCOP=1 OR INSERTDEPTH=0 THEN      %107-             02193800 C  0011:0016:0
              IF WRITNEW(LIN,FCR) THEN WRITE(NEWTAPE,10,LIN[*]);                            02194000 T  0011:0020:1
            IF OMITTING THEN IF NOT LISTATOG THEN GO AWAY;                                  02194250 T  0011:0027:3
            GO SW[LASTUSED];                                                                02194500 T  0011:0030:0
       LCARD:                                                                               02194750 T  0011:0032:0
            IF LISTER OR LISTPTOG THEN PRINTCARD;                                           02195000 T  0011:0032:0
            GO AWAY;                                                                        02195250 T  0011:0049:2
       LTAPE:                                                                               02195500 T  0011:0049:3
            IF LISTER THEN PRINTCARD;                                                       02195750 T  0011:0050:0
       %    GO AWAY;                                                                        02196000 T  0011:0066:0
       AWAY:                                                                                02196250 T  0011:0066:0
            END OUTPUTSOURCE;                                                               02196500 T  0011:0067:2
                                                                                  11 IS   68 LONG, NEXT SEG    3


            PROCEDURE BEGINPRINT;                                                           02196510 T  0003:0339:2
PRT(477) = BEGINPRINT
            BEGIN                                                                           02196520 T  0003:0339:2
                 STREAM PROCEDURE STUFF(N,L); VALUE N;                                      02196530 T  0003:0339:2
                                                                                  START OF SEGMENT **********   12
PRT(500) = STUFF
                 BEGIN                                                                      02196540 T  0012:0000:0
                      DI:=L; DS:=8 LIT " "; SI:=L; DS:=13 WDS;                              02196550 T  0012:0000:0
                      SI:=LOC N; DS:=8 DEC;                                                 02196560 T  0012:0002:0
                 END;                                                                       02196570 T  0012:0002:1



                 STUFF(BEGINSTACK[BSPOINT],LIN);                                            02196580 T  0012:0002:1
                 IF NOHEADING THEN DATIME; WRITELINE;                                       02196590 T  0012:0004:1
            END BEGINPRINT;                                                                 02196610 T  0012:0016:0
                                                                                  12 IS   17 LONG, NEXT SEG    3



       PROCEDURE READACARD;                                                                 02196750 T  0003:0339:2
       COMMENT  READACARD READS CARDS FROM EITHER THE CARD READER OR THE                    02197000 T  0003:0339:2
                TAPE MERGING AS REQUESTED AND CREATING A NEW TAPE AND                       02197250 T  0003:0339:2
                LISTING IF REQUESTED.  READACARD ALSO INSERTS A PERCENT                     02197500 T  0003:0339:2
                SIGN AS AN END OF CARD SENTINEL IN COLUMN 73 AND SETS                       02197750 T  0003:0339:2
                FCR,NCR,LCR,TLCR, AND CLCR;                                                 02198000 T  0003:0339:2
            BEGIN                                                                           02198250 T  0003:0339:2
            PROCEDURE READTAPE(LCR,MAXLCR,LIB); VALUE LIB; BOOLEAN LIB;                     02198500 T  0003:0339:2
```

```
                                                                                                                START OF SEGMENT **********   13
PRT(501) = READTAPE
                                                        REAL LCR, MAXLCR;                                                        02198750 T  0013:0000:0
                BEGIN                                                                                           %105-            02198755 C  0013:0000:0
                LABEL ENDREADTAPE, EOFT;                                                                        %105-            02198760 C  0013:0000:0
                                                                                                                START OF SEGMENT **********   14
                        IF LIB THEN                                                                                              02199000 T  0014:0000:0
                             BEGIN                                                                                               02199250 T  0014:0000:0
                             RECOUNT:=RECOUNT+1;                                                                                 02199500 T  0014:0000:1
                             IF LCR:=LCR+11>MAXLCR THEN                                                                          02199750 T  0014:0002:0
                                  BEGIN                                                                                          02200000 T  0014:0004:0
                                  READ(LIBRARY[FILEINX]);                                                                        02200250 T  0014:0004:1
                                  MAXLCR:=46+LCR:=MKABS(LIBRARY[FILEINX](0))+10;                                                 02200500 T  0014:0008:1
                                  END;                                                                                           02200750 T  0014:0015:3
                             ADDER(SEQSUM,LCR,TRUE,TRUE);                                                                        02201000 T  0014:0015:3
                             END                                                                                                 02201250 T  0014:0017:3
                        ELSE BEGIN                                                                                               02201500 T  0014:0017:3
                READ (TAPE, 10, TBUFF[*])[EOFT];                                                                %105-            02201750 P  0014:0018:0
PRT(502) = EOFT
PRT(503) = GO TO SOLVER
                MAXLCR:=LCR:=MKABS(TBUFF[9]);                                                                   %105-            02202000 P  0014:0023:2
                GO TO ENDREADTAPE;                                                                              %105-            02202010 C  0014:0026:0
                EOFT:                                                                                           %105-            02202020 C  0014:0026:1
                DEFINEARRAY[25]:="ND;END,"& "E"[1:43:5];                                                        %105-            02202030 C  0014:0027:2
                DEFINEARRAY[34]:="9999" & "9999"[1:25:23];                                                      %105-            02202040 C  0014:0029:2
                TLCR:= MKABS(DEFINEARRAY[34]);                                                                  %105-            02202050 C  0014:0031:3
                PUTSEQNO (DEFINEARRAY[33],TLCR-8);                                                              %105-            02202060 C  0014:0033:3
                TURNONSTOPLIGHT("%", TLCR-8);                                                                   %105-            02202070 C  0014:0035:2
                ENDREADTAPE:                                                                                    %105-            02202080 C  0014:0036:1
                END;                                                                                            %105-            02202090 C  0014:0037:2
                        END READTAPE;                                                                                            02202250 T  0014:0037:2
                                                                                                                14 IS   43 LONG, NEXT SEG   13


                PROCEDURE SEQCOMPARE(TLCR,CLCR, LIB); VALUE LIB; BOOLEAN LIB;                                                    02202500 T  0013:0000:0
PRT(504) = SEQCOMPARE
                                                        REAL TLCR, CLCR ;                                                        02202750 T  0013:0000:0
                BEGIN                                                                                                            02203000 T  0013:0000:0
                MEDIUM:="C "; % CARD READER.                                                                                     02203250 T  0013:0000:0
                IF GT1:=COMPARE(TLCR,CLCR)=0 THEN  % TAPE HAS LOW SEQUENCE NUMB                                                  02203500 T  0013:0000:1
                     BEGIN                                                                                                       02203750 T  0013:0003:2
                     LCR:=TLCR; LASTUSED:=IF LIB THEN 6 ELSE 3;                                                                  02204000 T  0013:0003:3
                     MEDIUM:=IF LIB THEN "CA"+FILEINX ELSE "T ";%CA,CB,CC,OR T,                                                  02204250 T  0013:0006:1
                     END                                                                                                         02204500 T  0013:0009:3
                ELSE BEGIN                                                                                                       02204750 T  0013:0009:3
                     IF GT1 ≠ 1 THEN % TAPE AND CARD HAVE SAME SEQ                                                               02205000 T  0013:0013:2
                          BEGIN                                                                                                  02205250 T  0013:0013:3
                          MEDIUM:="P "; % CARD PATCHES TAPE.                                                                     02205500 T  0013:0014:0
                               IF LIB THEN IF FINISHPT=RECOUNT=1 THEN                                                            02207750 T  0013:0015:2
                               LASTCRDPATCH:=TRUE ELSE                                                                           02208000 T  0013:0017:2
                               READTAPE(LTLCR,MAXLTLCR,TRUE) ELSE                                                                02208250 T  0013:0018:0
                               READTAPE(TLCR,MAXTLCR,FALSE);                                                                     02208500 T  0013:0021:2
                          END;                                                                                                   02208750 T  0013:0023:2
                     LCR:=CLCR;                                                                                                  02209000 T  0013:0023:2
                     LASTUSED:=IF LIB THEN 5 ELSE 2;                                                                             02209250 T  0013:0024:0
```

```
                    END;
                END OF SEQCOMPARE;


            LABEL CARDONLY, CARDLAST, TAPELAST,  EXIT, FIRSTTIME,
                  EOF, USETHESWITCH,
                  COMPAR,XIT,LIBEND, LIBTLAST,LIBCLAST;
            LABEL COPYLIB, COPYEOF;                                          %107-
            SWITCH USESWITCH := CARDONLY,CARDLAST,TAPELAST,FIRSTTIME,
                                LIBCLAST, LIBTLAST, COPYLIB;                 %107-
            BOOLEAN DOLLAR2TOG;
STACK(F+2) = DOLLAR2TOG
            IF ERRORCOUNT≥ERRMAX THEN ERR(611); % ERR LIMIT EXCEEDED - STOP,
        USETHESWITCH:
            GO TO USESWITCH[LASTUSED];
            MOVE(1,TEXT[LASTUSED,LINKR,LASTUSED,LINKC],
                 DEFINEARRAY[DEFINEINDEX-2]);
            LASTUSED := LASTUSED + 1;
            NCR := LCR-1;
            GO TO XIT;
        FIRSTTIME:
            READ(CARD,10,CBUFF[*]);
            FCR:=NCR:=(LCR:=MKABS(CBUFF[9]))-9;
            MEDIUM:="C ";
            IF EXAMIN(FCR)≠"$" AND LISTER THEN PRINTCARD;
            PUTSEQNO(INFO[LASTSEQROW,LASTSEQUENCE],LCR);
            CARDNUMBER:=CONV(INFO[LASTSEQROW,LASTSEQUENCE-1],5,8);
            TURNONSTOPLIGHT("%",LCR);
            GO XIT;
        COMMENT  WE HAVE JUST INITIALIZED CARD INPUT;
        CARDONLY:
            IF NORELEASE THEN GO TO EXIT; READ(CARD,10,CBUFF[*]);
            LCR := MKABS(CBUFF[9]); GO EXIT;
        CARDLAST:
            IF NORELEASE THEN GO TO EXIT; READ(CARD,10,CBUFF[*])[EOF];
PRT(505) = EOF
            CLCR := MKABS(CBUFF[9]);
            GO COMPAR;
        EOF:
            DEFINEARRAY[25]:="ND;END,"&"E"[1:43:5];
            DEFINEARRAY[34]:="9999"&"9999"[1:25:23];
            CLCR:=MKABS(DEFINEARRAY[34]);
            PUTSEQNO(DEFINEARRAY[33],CLCR-8);
            TURNONSTOPLIGHT("%",CLCR-8);
        %
            GO COMPAR;
        COMMENT  THIS RELEASES THE PREVIOUS CARD FROM CARD READER AND
                 SETS UP CLCR;
        TAPELAST:
            READTAPE(TLCR,MAXTLCR,FALSE); GO TO COMPAR;
        COMMENT  THIS RELEASES THE PREVIOUS CARD FROM TAPE AND SETS UP TLCR;
        LIBCLAST:
            IF FIRSTIMEX THEN
                BEGIN FIRSTIMEX:=FALSE; GO COMPAR END;
            READ(CARD,10,CBUFF[*])[EOF];
            CLCR := MKABS(CBUFF[9]);
```

```
02209500 T  0013:002610
02209750 T  0013:002610

02210000 T  0013:002611
02210250 T  0013:002611
02210500 T  0013:002611
02210600 C  0013:002611
02210750 T  0013:002611
02211000 P  0013:002913
02211250 T  0013:003313
02211500 T  0013:003313
02211750 T  0013:003610
02212000 T  0013:003610
02212250 T  0013:003810
02212500 T  0013:004112
02212750 T  0013:004210
02213000 T  0013:004313
02213250 T  0013:004411
02213500 T  0013:004512
02213750 T  0013:004610
02214000 T  0013:005010
02214100 T  0013:005313
02214200 T  0013:005411
02214250 T  0013:007410
02214260 T  0013:007610
02214500 T  0013:008010
02214750 T  0013:008112
02215000 T  0013:008113
02215250 T  0013:008113
02215500 T  0013:008210
02215750 T  0013:008712
02216000 T  0013:008913
02216250 T  0013:009010

02216500 T  0013:009610
02216750 T  0013:009810
02217000 T  0013:009811
02217250 T  0013:009912
02217500 T  0013:010112
02217750 T  0013:010313
02218000 T  0013:010513
02218250 T  0013:010712
02218400 T  0013:010811
02218500 T  0013:010811
02218750 T  0013:011210
02219000 T  0013:011210
02219250 T  0013:011210
02219500 T  0013:011210
02219750 T  0013:011313
02220000 T  0013:011313
02220250 T  0013:011410
02220500 T  0013:011410
02220750 T  0013:011610
02221000 T  0013:012112
```

```
                    IF LASTCRDPATCH THEN                                                                    02221250 T  0013:0123:2
                        BEGIN                                                                               02221500 T  0013:0123:3
                        LASTCRDPATCH:=FALSE;                                                                02221750 T  0013:0124:0
                        RECOUNT:=RECOUNT+1;                                                                 02222000 T  0013:0124:1
                        GO TO XIT                                                                           02222250 T  0013:0126:0
                        END;                                                                                02222500 T  0013:0126:1
                    GO TO COMPAR;                                                                           02222750 T  0013:0126:1
                LIBTLAST:                                                                                   02223000 T  0013:0127:2
                    IF FIRSTIMEX THEN                                                                       02223250 T  0013:0127:2
                        BEGIN FIRSTIMEX:=FALSE; GO TO COMPAR END;                                           02223500 T  0013:0127:2
                    READTAPE(LTLCR,MAXLTLCR,TRUE);                                                          02223750 T  0013:0129:2
                    IF RECOUNT=FINISHPT THEN GO TO XIT;                                                     02224000 T  0013:0130:0
                    GO COMPAR;                                                          %107-               02224010 C  0013:0131:3
                COPYLIB:                                                                %107-               02224020 C  0013:0132:0
                    READ(LF[INSERTINX:=INSERTINX+1],10,LBUFF[*])[COPYEOF];              %107-               02224030 C  0013:0132:0
PRT(506) = COPYEOF
                    READ SEEK(LF[INSERTINX+1]);                                         %107-               02224032 C  0013:0141:3
                    IF(CMPD(INSERTSEQ,LBUFF[9]) = 0) THEN GO COPYEOF;                   %107-               02224040 C  0013:0148:1
                    LCR:=MKABS(LBUFF[9]);                                               %107-               02224050 C  0013:0152:1
                    GO TO EXIT;                                                         %107-               02224060 C  0013:0154:1
                COPYEOF:                                                                %107-               02224070 C  0013:0155:2
                    CLOSE(LF,RELEASE);                                                  %107-               02224080 C  0013:0155:2
                    IF((INSERTDEPTH:=INSERTDEPTH-1)  =  0) THEN                         %107-               02224090 C  0013:0156:1
                        BEGIN LASTUSED:=SAVECARD; MEDIUM:=MEDIUM.[24:12];               %107-               02224100 C  0013:0158:1
                        GO USETHESWITCH;                                                %107-               02224102 C  0013:0161:2
                        END;                                                            %107-               02224104 C  0013:0161:3
                    FILL LF WITH INSERTMID, INSERTFID;                                  %107-               02224110 C  0013:0161:3
                    GO COPYLIB;                                                         %107-               02224120 C  0013:0167:2
                COMPAR:                                                                                     02224250 T  0013:0167:3
                    IF LASTUSED = 2 OR LASTUSED = 3 THEN SEQCOMPARE(TLCR,CLCR,FALSE)                        02224500 T  0013:0168:0
                        ELSE   SEQCOMPARE(LTLCR,CLCR,TRUE);                                                 02224750 T  0013:0171:2
            EXIT:                                                                                           02225000 T  0013:0173:2
                NCR := FCR:= LCR + 9;                                                                       02225250 T  0013:0174:0
            COMMENT  SETS UP NCR AND FCR;                                                                   02225500 T  0013:0175:3
                IF CHECKTOG AND EXAMIN(FCR)≠"$" THEN % $-CARDS DON"T COUNT.                                 02225750 T  0013:0175:3
                  IF COMPARE(MKABS(INFO[LASTSEQROW,LASTSEQUENCE]),LCR)=1 THEN                               02226000 T  0013:0178:1
                    IF SEQERRTOG THEN BEGIN FLAG(610);                                                      02226250 T  0013:0183:2
                                      NUMSEQUENCEERRORS:=NUMSEQUENCEERRORS+1;END                            02226300 T  0013:0185:3
               ELSE BEGIN % SEQUENCE WARNING                                                                02226500 T  0013:0186:1
                    BLANKET(14,LIN);                                                                        02226750 T  0013:0187:2
                    SEQUENCEWARNING(LIN[13]);                                                               02227000 T  0013:0188:1
                    IF NOHEADING THEN DATIME; WRITELINE;                                                    02227250 T  0013:0189:3
                    IF NOT LISTER THEN PRINTCARD;                                                           02227500 T  0013:0201:2
                   NUMSEQUENCEERRORS:=NUMSEQUENCEERRORS+1;                                                  02227600 T  0013:0217:3
                    END;                                                                                    02227750 T  0013:0218:1
                IF EXAMIN(FCR)="$" THEN                                                                     02228250 T  0013:0218:1
                    BEGIN                                                                                   02228500 T  0013:0220:1
                    IF LISTPTOG OR PRINTDOLLARTOG THEN PRINTCARD;                                           02228750 T  0013:0221:2
                    IF EXAMIN(NCR:=NCR+32768)="$" THEN MAKCAST ELSE DOLLARCARD;                             02229000 T  0013:0238:1
                        NORELEASE := FALSE;                                                                 02229100 T  0013:0247:3
            COMMENT  DONT FORGET THAT NCR IS NOT WORD MODE, BUT CHAR. MODE POINTER;                         02229250 T  0013:0248:0
                    GO USETHESWITCH;                                                                        02229500 T  0013:0248:0
                    END;                                                                                    02229750 T  0013:0248:1
                IF EXAMIN(FCR)=" " THEN                                                                     02230000 T  0013:0248:1
                  IF DOLLAR2TOG:=EXAMIN(FCR+32768)="$" THEN                                                 02230100 T  0013:0250:1
                    BEGIN                                                                                   02230250 T  0013:0253:3
                    OUTPUTSOURCE;                                                                           02230500 T  0013:0254:0
```

```
                    IF EXAMIN(NCR:=NCR+65536)="$" THEN MAKCAST ELSE                    02230750 T  0013:025:1
          DOLLARCARD;                                                                 02231000 T  0013:026:3
          END;                                                                        02231250 T  0013:026:1
     IF VOIDING OR VOIDTAPE THEN                                                      02231500 T  0013:026:1
          BEGIN                                                                       02231750 T  0013:026:0
          IF COMPARE(LCR,VOIDCR)=0 THEN                                               02232000 T  0013:026:1
               BEGIN                                                                  02232250 T  0013:026:1
               IF VOIDTAPE AND LASTUSED=3 OR NOT VOIDTAPE THEN                        02232500 T  0013:026:2
                 GO USETHESWITCH;                                                     02232750 T  0013:027:0
               END                                                                    02233000 T  0013:027:2
          ELSE BEGIN                                                                  02233250 T  0013:027:2
               VOIDCR:=VOIDPLACE:=0;                                                  02233500 T  0013:027:3
               VOIDING:=FALSE; VOIDTAPE:=FALSE                                        02233750 T  0013:027:1
               END;                                                                   02234000 T  0013:027:2
          END;                                                                        02234250 T  0013:027:0
     CARDCOUNT:=CARDCOUNT+1;                                                          02234500 T  0013:027:0
     IF DOLLAR2TOG THEN                                                               02234600 T  0013:027:3
               BEGIN DOLLAR2TOG:=NORELEASE:=FALSE; GO USETHESWITCH;END;               02234650 T  0013:027:3
     PUTSEQNO(INFO[LASTSEQROW,LASTSEQUENCE],LCR);                                     02234750 T  0013:028:0
          CARDNUMBER:=IF SEQTOG THEN TOTALNO+ADDVALUE ELSE                            02234800 T  0013:028:0
                    CONV(INFO[LASTSEQROW,LASTSEQUENCE-1],5,8);                        02234900 T  0013:028:1
     OUTPUTSOURCE;                                                                    02235000 T  0013:029:0
     IF OMITTING THEN GO USETHESWITCH;                                                02235250 T  0013:029:1
%                                                                                     02235500 T  0013:029:0
     TURNONSTOPLIGHT("%",LCR);                                                        02235750 T  0013:029:0
     IF BUILDLINE THEN                                                                02236000 T  0013:029:2
       IF LASTADDRESS ≠ (LASTADDRESS := L.[36:10]) THEN                               02236250 T  0013:029:3
            BEGIN                                                                     02236500 T  0013:029:3
            ENILSPOT := LASTADDRESS   & CARDNUMBER[10:20:28];                         02236750 T  0013:029:0
            IF (ENILPTR := ENILPTR+1)≥1023 THEN                                       02237000 T  0013:030:0
                 BEGIN FLAG(80);  ENILPTR := 512;  END;                               02237250 T  0013:030:0
            END;                                                                      02237500 T  0013:030:0
     XIT:                                                                             02237750 T  0013:030:0
          END READACARD;                                                              02238000 T  0013:030:0
                                                                                  13 IS  309 LONG, NEXT SEG    3


          PROCEDURE INCLUDECARD;                                          %107-       02238100 C  0003:033:2
PRT(507) = INCLUDECARD
          BEGIN                                                           %107-       02238110 C  0003:033:2
          REAL V;                                                         %107-       02238118 C  0003:033:2
                                                                                START OF SEGMENT **********   15
STACK(F+2) = V
          LABEL FEXIT,AGAIN,GETEM,EOF,EXIT,DONTSCAN;                      %107-       02238120 C  0015:000:0
          REAL STREAM PROCEDURE SCNN(A,B); VALUE B;                       %107-       02238130 C  0015:000:0
PRT(510) = SCNN
               BEGIN                                                      %107-       02238140 C  0015:000:0
               SI:=A; DI:=LOC SCNN; DS:=8 LIT"0          ";               %107-       02238150 C  0015:000:0
               DI:=DI+7; SI:=SI+3; DS:=B CHR;                             %107-       02238160 C  0015:000:3
               END;                                                       %107-       02238170 C  0015:000:1


          STREAM PROCEDURE MVE(A,B,C,D); VALUE B,C;                       %107-       02238180 C  0015:000:3
PRT(511) = MVE
```

```
                BEGIN                                                                %107-      02238190 C  0015:000313
                SI:=A; SI:=SI+3; DI:=D; C(DS:="LIT"0"); DS:=B CHR;                   %107-      02238200 C  0015:000410
                END;                                                                 %107-      02238210 C  0015:000712


            STREAM PROCEDURE MVEWD(A,B); VALUE A;                                    %107-      02238212 C  0015:000712
PRT(512) = MVEWD
            BEGIN SI:=A; DI:=B; DS:=10 WDS; END;                                     %107-      02238214 C  0015:000712


            DEFINE SKAN = BEGIN                                                      %107-      02238220 C  0015:000912
                          COUNT:=RESULT:=ACCUM[0]:=0;                                %107-      02238230 C  0015:000912
                          SCANNER;                                                   %107-      02238240 C  0015:000912
                          V:=SCNN(ACCUM[1],MIN(COUNT,7));                            %107-      02238250 C  0015:000912
                          END#;                                                      %107-      02238260 C  0015:000912
            DEFINE ERR(ERR1) = BEGIN FLAG(ERR1); GO TO EEXIT; END#;                  %107-      02238270 C  0015:000912
            IF((INSERTDEPTH:=INSERTDEPTH+1) > INSERTMAX) THEN ERR(612);              %107-      02238280 C  0015:000912
            INSERTMID:=INSERTFID:=INSERTINX:=INSERTCOP:=0;                           %107-      02238290 C  0015:001211
            INSERTSEQ:="9999"&"9999"[1:23];                                          %107-      02238300 C  0015:001913
        AGAIN:                                                                       %107-      02238330 C  0015:002211
            SKAN;                                                                    %107-      02238340 C  0015:002312
        DONTSCAN:                                                                    %107-      02238342 C  0015:003010
            IF V="%      " THEN GO GETEM;                                            %107-      02238350 C  0015:003112
            IF V="/      " THEN GO AGAIN;                                            %107-      02238360 C  0015:003210
            IF RESULT=3 THEN % SEQ RANGE                                             %107-      02238370 C  0015:003313
            BEGIN                                                                    %107-      02238380 C  0015:003410
            MVE(ACCUM[1],COUNT:=MIN(COUNT,8),8-COUNT,INSERTINX);                     %107-      02238385 C  0015:003411
            SKAN;                                                                    %107-      02238390 C  0015:004011
            IF V="-      " THEN                                                      %107-      02238400 C  0015:004810
                BEGIN                                                                %107-      02238410 C  0015:004811
                SKAN;                                                                %107-      02238420 C  0015:004912
                IF RESULT # 3 THEN ERR(614);                                         %107-      02238430 C  0015:005611
                MVE(ACCUM[1],COUNT:=MIN(COUNT,8),8-COUNT,INSERTSEQ);                 %107-      02238440 C  0015:006312
            END ELSE GO TO DONTSCAN;                                                 %107-      02238450 C  0015:006912
            GO AGAIN;                                                                %107-      02238460 C  0015:006912
            END; % SEQ RANGE                                                         %107-      02238470 C  0015:006913
        IF V="+      " THEN % WE HAVE COPY FORM                                      %107-      02238480 C  0015:006913
            BEGIN                                                                    %107-      02238490 C  0015:007010
            SKAN;                                                                    %107-      02238500 C  0015:007011
            IF V="COPY   " THEN                                                      %107-      02238510 C  0015:007810
                IF EXAMIN(LCR-9)="$" THEN                                            %107-      02238512 C  0015:007811
                    INSERTCOP:=INSERTINFO[INSERTDEPTH-1,4]                           %107-      02238514 C  0015:008113
                ELSE ERR(617)                                                        %107-      02238520 C  0015:008411
            ELSE ERR(616);                                                           %107-      02238522 C  0015:008912
            GO AGAIN;                                                                %107-      02238530 C  0015:009112
            END;                                                                     %107-      02238540 C  0015:009113
        IF INSERTMID=0 THEN INSERTMID:=V                                             %107-      02238550 C  0015:009113
        ELSE IF INSERTFID=0 THEN INSERTFID:=V ELSE ERR(616);                         %107-      02238552 C  0015:009512
        GO AGAIN;                                                                    %107-      02238555 C  0015:010210
    GETEM:                                                                           %107-      02238560 C  0015:010211
        IF NOT BOOLEAN(INSERTCOP) AND NEWTOG THEN                                    %107-      02238570 C  0015:010312
        IF EXAMIN(FCR) = "$" THEN % ONLY IF "$" IS IN COLUMN ONE                     %107-      02238572 C  0015:010513
        IF BOOLEAN(INSERTINFO[INSERTDEPTH-1,4]) THEN % ONLY IF LAST HAD COPY         %107-      02238574 C  0015:010713
```

```
                        BEGIN MVEWD(FCR,LBUFF[0]);                                          %107-      02238580 C  0015:011:0
                        PUTSEQNO(LBUFF[9],MKABS(INFO[LASTSEQROW,LASTSEQUENCE]));            %107-      02238582 C  0015:011:3
                        WRITE(NEWTAPE,10,LBUFF[*]);                                         %107-      02238590 C  0015:011:2
                        END;                                                                %107-      02238600 C  0015:011:2
                    IF INSERTMID=0 THEN ERR(613);                                           %107-      02238602 C  0015:011:2
                    IF INSERTFID=0 THEN INSERTFID:=TIME(=1);                                %107-      02238610 C  0015:012:1
                    IF INSERTFID=0 THEN                                                     %107-      02238620 C  0015:012:3
                        BEGIN INSERTFID:=INSERTMID; INSERTMID:=0; END;                      %107-      02238630 C  0015:012:3
                    IF INSERTDEPTH > 1 THEN CLOSE(LF,RELEASE);                              %107-      02238640 C  0015:013:2
                    FILL LF WITH INSERTMID,INSERTFID;                                       %107-      02238650 C  0015:013:0
                    READ(LF[0],10,LBUFF[*])[EEXIT];     % DO THE FOLLOWING SO THAT      %107-      02238652 C  0015:014:3
PRT(513) = EEXIT
                    INSERTMID:=LF,MFID;                 % IF THE OPERATOR IL=ED US      %107-      02238654 C  0015:015:0
                    INSERTFID:=LF,FID;                  % WE WILL HAVE THE PROPER NAMES.  02238656 C  0015:015:1
                    V:=-1;                                                                  02238658 C  0015:015:2
                    IF INSERTINX > 0 THEN                                               %107-      02238660 C  0015:016:0
                        BEGIN                                                           %107-      02238670 C  0015:016:0
                        DO READ(LF[V:=V+1],10,LBUFF[*])[EEXIT]                          %107-      02238680 C  0015:016:1
                        UNTIL CMPD(INSERTINX,LBUFF[9]) ≤ 1;                             %107-      02238690 C  0015:016:3
                        V:=V-1;                                                         %107-      02238700 C  0015:017:0
                        END;                                                            %107-      02238702 C  0015:017:3
                    INSERTINX:=V;                                                       %107-      02238704 C  0015:017:3
                    IF INSERTDEPTH = 1 THEN                                             %107-      02238710 C  0015:017:3
                    BEGIN SAVECARD:=LASTUSED; LASTUSED:=7; MEDIUM:="L "& MEDIUM[24:12];             02238720 C  0015:017:0
                    END;                                                                %107-      02238730 C  0015:018:0
                    GO TO EXIT;                                                         %107-      02238760 C  0015:018:0
                EEXIT:                                                                  %107-      02238770 C  0015:018:0
                    IF((INSERTDEPTH:=INSERTDEPTH-1)  >  0) THEN                         %107-      02238780 C  0015:018:0
                        BEGIN                                                           %107-      02238790 C  0015:018:3
                        CLOSE(LF,RELEASE);                                              %107-      02238800 C  0015:018:0
                        FILL LF WITH INSERTMID,INSERTFID;                               %107-      02238810 C  0015:019:0
                        END;                                                            %107-      02238820 C  0015:019:3
                EXIT:                                                                   %107-      02238830 C  0015:019:3
                    Q:="1%0000";                                                        %107-      02238832 C  0015:019:0
                    END;                                                                %107-      02238840 C  0015:019:1
                                                                                              15 IS  203 LONG, NEXT SEG    3


                REAL PROCEDURE CONVERT;                                                           02248000 T  0003:033:2
PRT(514) = CONVERT
                    BEGIN   REAL  T;  INTEGER  N;                                                 02249000 T  0003:033:2
                                                                                        START OF SEGMENT **********   16
STACK(F+3) = T
STACK(F+4) = N

                    TLO←0;   THI←                                                                 02250000 T  0016:000:0
                        T← CONV(ACCUM[1],TCOUNT,N←(COUNT=TCOUNT)MOD 8);                           02251000 T  0016:000:1
                        FOR N← TCOUNT+N STEP 8 UNTIL COUNT= 1 DO                                  02252000 T  0016:000:5:2
                    IF DPTOG THEN                                                                 02253000 T  0016:001:0
                    BEGIN                                                                         02254000 T  0016:001:0
                    DOUBLE(THI,TLO,100000000,0,0,×,CONV(ACCUM[1],N,8),0,+,←,                      02255000 T  0016:001:1
                                THI,TLO);                                                         02256000 T  0016:001:5:2
                    T←THI;                                                                        02257000 T  0016:001:6:0
                    END  ELSE                                                                     02258000 T  0016:001:6:1
                        T←  T×100000000+ CONV(ACCUM[1],N,8);                                      02259000 T  0016:001:6:1
                        CONVERT←T;                                                                02260000 T  0016:002:4:0
                    END;                                                                          02261000 T  0016:002:4:1
```

```
                                                                                                                    16 IS   28 LONG, NEXT SEG    3

                REAL STREAM PROCEDURE FETCH(F); VALUE F;                                                    02262000 T  0003:039:2
PRT(515) = FETCH
                    BEGIN  SI:=F; SI:=SI*8; DI:=LOC FETCH; DS:=WDS  END FETCH;                               02263000 T  0003:039:2



                PROCEDURE DUMPINFO;                                                                         02264000 T  0003:042:0
PRT(516) = DUMPINFO
                    BEGIN                                                                                   02264050 T  0003:042:0
                    ARRAY A[0:14]; INTEGER JEDEN,DWA;                                                       02264100 T  0003:042:0
                                                                                                      START OF SEGMENT **********   17
STACK(F+2) = A
STACK(F+3) = JEDEN
STACK(F+4) = DWA
                    STREAM PROCEDURE OCTALWORDS(S,D,N); VALUE N;                                            02264400 T  0017:001:3
PRT(517) = OCTALWORDS
                        BEGIN                                                                               02264450 T  0017:001:3
                        SI:=S; DI:=D;                                                                       02264500 T  0017:003:2
                        N(2(8(DS:=3 RESET; 3(IF SB THEN DS:=1 SET ELSE                                      02264550 T  0017:003:3
                            DS:=1 RESET; SKIP 1 SB)); DS:=1 LIT " ");DS:=2 LIT" ");                         02264600 T  0017:006:1
                        END OF OCTALWORDS;                                                                  02264650 T  0017:009:2



                    STREAM PROCEDURE ALPHAWORDS(S,D,N); VALUE N;                                            02264700 T  0017:009:2
PRT(520) = ALPHAWORDS
                        BEGIN                                                                               02264750 T  0017:009:2
                        SI:=S; DI:=D;                                                                       02264800 T  0017:010:0
                        N(2(4(DS:=1 LIT" "; DS:=1 CHR); DS:=1 LIT" "); DS:=2 LIT" ");                       02264850 T  0017:010:1
                        END OF ALPHAWORDS;                                                                  02264900 T  0017:014:1



                    IF NOHEADING THEN DATIME;WRITE(LINE[DBL],<//"ELBAT">);                                  02264950 T  0017:014:1
PRT(521) = *FORMAT DESCRIPTOR*

                                                                                                    18 IS    6 LONG, NEXT SEG   17
                    FOR JEDEN:=0 STEP 6 UNTIL 71 DO                                                         02265000 T  0017:019:2
                        BEGIN                                                                               02265050 T  0017:020:0
                        BLANKET(14,A); OCTALWORDS(ELBAT[JEDEN],A,6);                                        02265100 T  0017:020:0
                        WRITE(LINE[DBL],15,A[*]);                                                           02265150 T  0017:023:2
                        END;                                                                                02265200 T  0017:027:2
                    BLANKET(14,A); OCTALWORDS(ELBAT[72],A,4);                                               02265250 T  0017:029:3
                    WRITE(LINE[DBL],15,A[*]);                                                               02265300 T  0017:032:1
                    FOR JEDEN:=0 STEP 1 UNTIL NEXTINFO DIV 256 DO                                           02265350 T  0017:036:1
                        BEGIN                                                                               02265400 T  0017:041:2
                        WRITE(LINE[DBL],<//"INFO[",I2,",*]">,JEDEN);                                        02265450 T  0017:041:2
PRT(522) = *FORMAT DESCRIPTOR*

                                                                                                    19 IS    8 LONG, NEXT SEG   17
PRT(523) = *LIST, LABEL, OR SEGMENT DESCRIPTOR*
```

```
        FOR DWA:=0 STEP 6 UNTIL 251 DO                                          02265500 T  0017:004810
             BEGIN                                                              02265550 T  0017:004912
             BLANKET(14,A); ALPHAWORDS(INFO[JEDEN,DWA],A,6);                    02265600 T  0017:004912
             WRITE(LINE,15,A[*]);                                               02265650 T  0017:005211
             BLANKET(14,A); OCTALWORDS(INFO[JEDEN,DWA],A,6);                    02265700 T  0017:005712
             WRITE(LINE[DBL],15,A[*]);                                          02265750 T  0017:006011
             END;                                                               02265800 T  0017:006512
        BLANKET(14,A); ALPHAWORDS(INFO[JEDEN,252],A,4);                         02265850 T  0017:006712
        WRITE(LINE,15,A[*]);                                                    02265900 T  0017:007112
        BLANKET(14,A); OCTALWORDS(INFO[JEDEN,252],A,4);                         02265950 T  0017:007512
        WRITE(LINE[DBL],15,A[*]);                                               02266000 T  0017:007912
        END;                                                                    02266050 T  0017:008312
    END OF DUMPINFO;                                                            02266100 T  0017:008313
                                                                          17 IS   88 LONG, NEXT SEG    3


DEFINE SKAN = BEGIN                                                             02277000 T  0003:034210
              COUNT:=RESULT:=ACCUM[1]:=0;                                       02278000 T  0003:034210
              SCANNER;                                                          02279000 T  0003:034210
              Q:=ACCUM[1];                                                      02280000 T  0003:034210
              END #;                                                            02281000 T  0003:034210
COMMENT  DOLLARCARD HANDLES THE COMPILER CONTROL CARDS.                         02282000 T  0003:034210
         ALL COMPILER- AND USER-DEFINED OPTIONS ARE KEPT                        02283000 T  0003:034210
         IN THE ARRAY "OPTIONS".                                                02284000 T  0003:034210
         EACH OPTION HAS A TWO-WORD ENTRY:                                      02285000 T  0003:034210
                                                                                02286000 T  0003:034210
           WORD          CONTAINS                                               02287000 T  0003:034210
           ----          --------                                               02288000 T  0003:034210
             1           ENTRY FROM ACCUM[1]: 00XZZZZ, WHERE                    02289000 T  0003:034210
                         X IS THE SIZE OF THE ID AND                            02290000 T  0003:034210
                         ZZZZZ IS THE FIRST FIVE CHARS OF THE ID.               02291000 T  0003:034210
             2           PUSH-DOWN, 47-BIT STACK CONTAINING THE                 02292000 T  0003:034210
                         HISTORY OF THE SETTINGS OF THIS OPTION.                02293000 T  0003:034210
                                                                                02294000 T  0003:034210
                   IN "FINDOPTION", ALL COMPILER-DEFINED OPTIONS ARE USUALLY    02295000 T  0003:034210
                   LOCATED BASED UPON A UNIQUE NUMBER ASSIGNED TO EACH.         02296000 T  0003:034210
                   FOR ALL USER-DEFINED OPTIONS, A SEQUENTIAL TABLE SEARCH IS   02297000 T  0003:034210
                   INITIATED USING "USEROPINX" AS THE INITIAL INDEX INTO THE    02298000 T  0003:034210
                   "OPTIONS" ARRAY. IF THE NUMBER OF COMPILER-DEFINED OPTIONS   02299000 T  0003:034210
                   IS CHANGED, THEN "USEROPINX" MUST BE ACCORDINGLY CHANGED.    02300000 T  0003:034210
                   THE NUMBER OF USER DEFINED OPTIONS ALLOWED CAN BE            02301000 T  0003:034210
                   CHANGED BY CHANGING THE DEFINE "OPARSIZE".                   02302000 T  0003:034210
                   THE VARIABLE "OPTIONWORD" CONTAINS THE CURRENT TRUE OR FALSE 02303000 T  0003:034210
                   SETTING OF ALL OF THE COMPILER-DEFINED OPTIONS, ONE BIT PER  02304000 T  0003:034210
                   OPTION.                                                      02305000 T  0003:034210
                ;                                                               02306000 T  0003:034210
              BOOLEAN PROCEDURE FINDOPTION(BIT); VALUE BIT; INTEGER BIT;        02307000 T  0003:034210
PRT(524) = FINDOPTION                                                           02308000 T  0003:034210
                   BEGIN                                                        02309000 T  0003:034210
                   LABEL FOUND;                                           START OF SEGMENT **********   20
                   REAL ID;                                                     02310000 T  0020:000010
STACK(F+3) = ID                                                                 02311000 T  0020:000010
                   OPINX:=2*BIT-4;                                              02312000 T  0020:000113
                   WHILE ID:=OPTIONS[OPINX:=OPINX+2]#0 DO                       02313000 T  0020:000512
                        IF Q=ID THEN GO FOUND;
```

```
                    OPTIONS[OPINX]:=0; % NEW USER-DEFINED OPTION.                                          02314000 T  0020:0006:1
                FOUND:                                                                          %103-      02315000 P  0020:0008:0
                    IF OPINX +1>OPARSIZE THEN FLAG(602) ELSE % TOO MANY USER OPTIONS                       02316000 P  0020:0008:0
                    FINDOPTION:=BOOLEAN(OPTIONS[OPINX+1]);                                                 02317000 T  0020:0010:1
                    END FINDOPTION;                                                                        02318000 T  0020:0012:1
                                                                                                  20 IS   15 LONG, NEXT SEG    3



                PROCEDURE DOLLARCARD;                                                                      02319000 T  0003:0342:0
                    BEGIN                                                                                  02320000 T  0003:0342:0
                    PROCEDURE SWITCHIT(XBIT); VALUE XBIT; INTEGER XBIT;                                    02321000 T  0003:0342:0
                                                                                                 START OF SEGMENT **********   21
PRT(525) = SWITCHIT
                        BEGIN                                                                              02322000 T  0021:0000:0
                        BOOLEAN B,T;                                                                       02323000 T  0021:0000:0
                                                                                                 START OF SEGMENT **********   22
STACK(F+2) = B
STACK(F+3) = T
                        INTEGER SAVEINX;                                                                   02324000 T  0022:0000:0
STACK(F+4) = SAVEINX
                        LABEL XMODE0,XMODE1,XMODE2,XMODE3,XMODE4,ALONG;                                    02325000 T  0022:0000:0
                        SWITCH SW:=XMODE0,XMODE1,XMODE2,XMODE3,XMODE4;                                     02326000 T  0022:0000:0
                        SETTING:=FINDOPTION(XBIT); SKAN;                                                   02327000 T  0022:0005:3
                        GO SW[XMODE+1];                                                                    02328000 T  0022:0011:2
                    XMODE0: % FIRST OPTION ON CARD, BUT NOT SET, RESET, OR POP.                            02329000 T  0022:0013:3
                        OPTIONWORD:=BOOLEAN(0);                                                            02330000 T  0022:0014:0
                        FOR SAVEINX:=1 STEP 2 UNTIL OPARSIZE DO OPTIONS[SAVEINX]:=0;                       02331000 T  0022:0014:1
                        IF BUILDLINE.[45:1] THEN                                                %108-      02331050 C  0022:0019:3
                           BUILDLINE.[47:1]:=SEQXEQTOG:=FALSE;                                  %108-      02331060 C  0022:0020:0
                        XMODE:=1; IF LASTUSED < 5 AND LASTUSED # 3 THEN LASTUSED:=1;                       02332000 P  0022:0023:2
                    XMODE1: % NOT FIRST OPTION AND NOT BEING SET, RESET, OR POPPED.                        02333000 T  0022:0026:1
                        OPTIONS[OPINX+1]:=REAL(TRUE);                                                      02334000 T  0022:0027:2
                        IF XBIT<USEROPINX THEN OPTIONWORD:=OPTIONWORD & TRUE[XBIT:1];                      02335000 T  0022:0028:1
PRT(526) = DYNAMIC DIALS
                        GO ALONG;                                                                          02336000 T  0022:0032:1
                    XMODE2: % RESET,                                                                       02337000 T  0022:0033:2
                        OPTIONS[OPINX+1]:=REAL(FALSE & SETTING[1:2:46]);                                   02338000 T  0022:0034:0
                        IF XBIT<USEROPINX THEN OPTIONWORD:=OPTIONWORD & FALSE[XBIT:1];                     02339000 T  0022:0036:1
                        GO ALONG;                                                                          02340000 T  0022:0040:1
                    XMODE3: % SET,                                                                         02341000 T  0022:0041:2
                        SAVEINX:=OPINX; % REMEMBER OPTION WE ARE SETTING.                                  02342000 T  0022:0042:0
                        B:=IF Q="1=0000" THEN BOOLEXP ELSE TRUE;                                           02343000 T  0022:0042:1
                        OPTIONS[SAVEINX+1]:=REAL(B & SETTING[1:46]);                                       02352000 T  0022:0045:3
                        IF XBIT<USEROPINX THEN OPTIONWORD:=OPTIONWORD & B [XBIT:1];                        02353000 T  0022:0048:1
                        GO ALONG;                                                                          02354000 T  0022:0052:1
                    XMODE4: % POP,                                                                         02355000 T  0022:0054:0
                        OPTIONS[OPINX+1]:=REAL(B:=SETTING.[1:46]);                                         02356000 T  0022:0054:0
                        IF XBIT<USEROPINX THEN OPTIONWORD:=OPTIONWORD & B [XBIT:1];                        02357000 T  0022:0057:2
                    ALONG:                                                                                 02358000 T  0022:0061:2
                        END SWITCHIT;                                                                      02359000 T  0022:0062:0
                                                                                                  22 IS   65 LONG, NEXT SEG   21


                    LABEL EXIT,AGAIN,SKANAGAIN,LENGTH1,LENGTH2,LENGTH3,LENGTH4,                            02360000 T  0021:0000:0
```

```
                    LENGTH5,LENGTH6,LENGTH7,LENGTH8,LENGTH9,                                  02361000 T  0021:000:0
                    WHATISIT,                                                                 02362000 T  0021:000:0
                    CARDOPTION,MERGEOPTION;                                                   02363000 T  0021:000:0
               SWITCH OPTIONLENGTH:=LENGTH1,WHATISIT,LENGTH3,LENGTH4,LENGTH5,                 02364000 T  0021:000:0
                                  LENGTH6,LENGTH7,LENGTH8,LENGTH9,WHATISIT;  %106-            02365000 P  0021:0002:1
               INTEGER SRESULT,SCOUNT;                                                        02365100 T  0021:0008:0
STACK(F+2) = SRESULT
STACK(F+3) = SCOUNT
          INTEGER SAVEINX;                                                    %108-           02365200 C  0021:0008:0
STACK(F+4) = SAVEINX
               DOLLARTOG:=TRUE;                                                               02366000 T  0021:0008:0
               MOVE(10,ACCUM[0],DEFINEARRAY[0]); % SAVE INFORMATION FOR                       02366100 T  0021:0009:3
               SCOUNT:=COUNT; SRESULT:=RESULT;   % "TABLE" TO RESUME SCAN.                   02366200 T  0021:001:3
               XMODE:=0;                                                                      02367000 T  0021:001:2
               PUTSEQNO(INFO[LASTSEQROW,LASTSEQUENCE],LCR);                                   02368000 T  0021:001:0
               TURNONSTOPLIGHT("%",LCR);                                                      02369000 T  0021:001:0
          SKANAGAIN:                                                                          02370000 T  0021:001:2
               SKAN;                                                                          02371000 T  0021:001:2
          AGAIN:                                                                              02372000 T  0021:002:1
               GO OPTIONLENGTH[MIN(COUNT,10)];                                                02373000 T  0021:002:2
          LENGTH1:                                                                            02374000 T  0021:002:2
               IF Q = "1%0000" THEN GO EXIT;                                                  02375000 T  0021:002:0
               IF Q = "1$0000" THEN                                                           02376000 T  0021:002:2
                    BEGIN SWITCHIT(PRINTDOLLARBIT); GO AGAIN END;                             02377000 T  0021:002:0
               IF Q = "1,0000" THEN GO SKANAGAIN;                                             02378000 T  0021:003:0
               GO WHATISIT;                                                                   02379000 T  0021:003:2
          LENGTH2; % NO OPTIONS OF THIS LENGTH ARE CURRENTLY IMPLEMENTED.                     02380000 T  0021:003:2
          LENGTH3:                                                                            02381000 T  0021:003:2
               IF Q = "3SET00" THEN                                                           02382000 T  0021:003:2
                    BEGIN XMODE:=3; GO SKANAGAIN END;                                         02383000 T  0021:003:3
               IF Q = "3POP00" THEN                                                           02384000 T  0021:003:2
                    BEGIN XMODE:=4; GO SKANAGAIN END;                                         02385000 T  0021:003:3
               IF Q = "3NEW00" THEN                                                           02386000 T  0021:004:2
                    BEGIN                                                                     02387000 T  0021:004:3
                    SWITCHIT(NEWBIT);                                                         02388000 T  0021:004:0
                    IF Q = "4TAPE0" THEN GO SKANAGAIN;                                        02389000 T  0021:004:2
                    GO AGAIN;                                                                 02390000 T  0021:004:0
                    END;                                                                      02391000 T  0021:004:2
               IF Q = "3SEQ00" THEN                                                           02392000 T  0021:004:2
                    BEGIN SWITCHIT(SEQBIT); GO AGAIN END;                                     02393000 T  0021:004:3
               IF Q = "3PRT00" THEN                                                           02394000 T  0021:005:2
                    BEGIN SWITCHIT(PRTBIT); GO AGAIN END;                                     02395000 T  0021:005:3
               IF Q = "3MCP00" THEN                                                           02396000 T  0021:005:2
                    BEGIN SWITCHIT(MCPBIT); GO AGAIN END;                                     02397000 T  0021:005:3
               GO WHATISIT;                                                                   02398000 T  0021:006:2
          LENGTH4:                                                                            02399000 T  0021:006:3
               IF Q = "4LISTO" THEN                                                           02400000 T  0021:006:0
                    BEGIN                                                                     02401000 T  0021:006:1
                    SWITCHIT(LISTBIT);                                                        02402000 T  0021:006:2
                    GO AGAIN;                                                                 02404000 T  0021:006:0
                    END;                                                                      02405000 T  0021:006:0
               IF Q = "4VOIDO" THEN                                                           02406000 T  0021:006:0
                    BEGIN                                                                     02407000 T  0021:006:1
                    IF XMODE=0 THEN                                                           02408000 T  0021:006:2
                         BEGIN                                                                02409000 T  0021:006:0
                         GETVOID(VOIDPLACE,NCR,VOIDCR,LCR,                                    02410000 T  0021:006:1
                                 INFO[LASTSEQROW,LASTSEQUENCE]);                              02411000 T  0021:007:0
                         XMODE:=1; SWITCHIT(VOIDBIT);                                         02412000 T  0021:007:3
```

```
                GO EXIT;                                                                02413000 T  0021:0073:2
                END;                                                                    02414000 T  0021:0075:2
            SWITCHIT(VOIDBIT);                                                          02415000 T  0021:0075:2
            VOIDPLACE:="9999"&"9999"[1:23];%2 B COMPATIBLE W/B-5700 VOIDS               02416000 T  0021:0075:3
            VOIDCR:=MKABS(VOIDPLACE); % AND FAKE OUT READACARD.                         02417000 T  0021:0077:3
            GO AGAIN;                                                                   02418000 T  0021:0079:2
            END;                                                                        02419000 T  0021:0081:2
        IF Q = "4XREFO" THEN BEGIN SWITCHIT(XREFBIT); IF BOOLEAN(XMODE) THEN            02419100 T  0021:0081:2
            DEFINING := BOOLEAN(REAL(DEFINING)&1[1:47:1]);                              02419110 T  0021:0083:2
                                GO AGAIN  END;                                          02419120 T  0021:0085:3
        IF Q = "4BENDO" THEN BEGIN SWITCHIT(BENDBIT); GO AGAIN END;                     02419200 T  0021:0087:2
        IF Q = "4OMITO" THEN                                                            02420000 T  0021:0091:2
            BEGIN SWITCHIT(OMITBIT); GO AGAIN END;                                      02421000 T  0021:0091:3
        IF Q = "4CARDO" THEN                                                            02422000 T  0021:0095:2
            BEGIN                                                                       02423000 T  0021:0095:3
            Q:="4TAPEO"; % FAKE OUT SWITCHIT.                                           02424000 T  0021:0096:0
            SWITCHIT(MERGEBIT);                                                         02425000 T  0021:0097:2
            IF XMODE#2 THEN MERGETOG:=NOT MERGETOG;                                     02425500 T  0021:0097:3
            OPTIONS[2×MERGEBIT-1]:=                 % CARD IS                          02426000 T  0021:0101:3
              REAL(SETTING & (MERGETOG)[47:1]);  % INVERSE OF MERGE.                   02427000 T  0021:0103:2
            IF MERGETOG THEN GO MERGEOPTION;                                            02428000 T  0021:0105:2
CARDOPTION:                                                                             02429000 T  0021:0106:1
            IF LASTUSED<5 THEN LASTUSED:=1;                                             02430000 T  0021:0107:2
            GO AGAIN;                                                                   02431000 T  0021:0109:2
            END;                                                                        02432000 T  0021:0112:0
        IF Q = "4TAPEO" THEN                                                            02433000 T  0021:0112:0
                BEGIN                                                                   02434000 T  0021:0112:1
                SWITCHIT(MERGEBIT);                                                     02435000 T  0021:0113:2
                IF NOT MERGETOG THEN GO CARDOPTION;                                     02436000 T  0021:0114:0
            MERGEOPTION:                                                                02437000 T  0021:0115:2
                IF LASTUSED # 1 THEN GO TO AGAIN;                       %110-           02437500 C  0021:0116:0
                LASTUSED:=2; % NEXT CARD IS READ FROM READER.                           02438000 T  0021:0117:2
                IF MAXTLCR=0 THEN                                                       02439000 T  0021:0118:0
                    BEGIN                                                               02440000 T  0021:0118:1
                    INTEGER STREAM PROCEDURE FEJ(F,T); VALUE T;                         02441000 T  0021:0119:2

PRT(527) = *SEGMENT DESCRIPTOR*                                                                 START OF SEGMENT **********   23

PRT(530) = FEJ
                        BEGIN                                                           02442000 T  0023:0000:0
                        SI:=F; DI:=LOC T; DS:=WDS;                                      02443000 T  0023:0000:0
                        SI:=T; SI:=SI-16; DI:=LOC FEJ; DS:=WDS;                         02444000 T  0023:0000:1
                        END FEJ;                                                        02445000 T  0023:0001:3



                    STREAM PROCEDURE FIX(F,T); VALUE T;                                 02446000 T  0023:0002:1
PRT(531) = FIX
                        BEGIN                                                           02447000 T  0023:0002:1
                        SI:=F; SI:=SI-24; DI:=LOC T; DS:=WDS;                           02448000 T  0023:0003:2
                        DI:=T; DI:=DI+47; SKIP 4 DB; DS:=2 RESET;                       02449000 T  0023:0004:0
                        2(DI:=DI+48); DS:=8 LIT"00#01+0#";                              02450000 T  0023:0005:2
                        END FIX;                                                        02451000 T  0023:0007:2



                    IF GT1:=FEJ(TAPE,0)=10 THEN                                         02452000 T  0023:0007:2
```

```
                              BEGIN                                                                  02453000 T  0023:0010:1
                              REWIND(TAPE); FIX(TAPE,0);                                             02454000 T  0023:0011:2
                              END;                                                                   02455000 T  0023:0014:0
                         MAXTLCR:=GT1+TLCR:=9+MKABS(TBUFF[0]);                                       02456000 T  0023:0014:0
                         READ(TAPE,10,TBUFF[*]); % INITIALIZE TAPE INPUT,                            02457000 T  0023:0017:3
                         LASTUSED:=2;                                                                02458000 T  0023:0022:0
                         END;                                                                        02459000 T  0023:0022:1
PRT(532) = *SEGMENT DESCRIPTOR*

                                                                                                23 IS   24 LONG, NEXT SEG   21
                              GO AGAIN;                                                              02460000 T  0021:0121:2
                              END;                                                                   02461000 T  0021:0121:3
                         IF Q = "4PAGEO" THEN                                                        02462000 T  0021:0121:3
                              BEGIN                                                                  02463000 T  0021:0122:0
                              IF LISTER THEN WRITE(LINE[PAGE]);                                      02464000 T  0021:0122:1
                              GO SKANAGAIN;                                                          02465000 T  0021:0127:3
                              END;                                                                   02466000 T  0021:0129:2
                         IF Q = "4INFOO" THEN                                                        02467000 T  0021:0129:2
                              BEGIN DUMPINFO; GO SKANAGAIN END;                                      02468000 T  0021:0129:3
                         IF Q = "4SEGSO" THEN                                                        02469000 T  0021:0133:2
                              BEGIN SWITCHIT(SEGSBIT); GO AGAIN END;                                 02470000 T  0021:0133:3
                         IF Q = "4NESTO" THEN                                                        02471000 T  0021:0137:2
                              BEGIN SWITCHIT(NESTBIT); GO AGAIN END;                                 02472000 T  0021:0137:3
                         IF Q = "4DECKO" THEN                                                        02473000 T  0021:0141:2
                              BEGIN SWITCHIT(DECKBIT); GO AGAIN END;                                 02474000 T  0021:0141:3
                         GO WHATISIT;                                                                02475000 T  0021:0145:2
                    LENGTH5:                                                                         02476000 T  0021:0145:3
                         IF Q = "5RESET" THEN                                                        02477000 T  0021:0146:0
                              BEGIN XMODE:=2; GO SKANAGAIN END;                                      02478000 T  0021:0146:1
                         IF Q = "5LISTP" THEN                                                        02479000 T  0021:0150:0
                              BEGIN SWITCHIT(LISTPBIT); GO AGAIN; END;                               02480000 T  0021:0150:1
IF Q = "5VOIDT" THEN                                                                                 02481000 T  0021:0154:0
     BEGIN                                                                                           02482000 T  0021:0154:1
     IF XMODE=0 THEN                                                                                 02483000 T  0021:0155:2
          BEGIN                                                                                      02484000 T  0021:0156:0
          GETVOID(VOIDPLACE,NCR,VOIDCR,LCR,                                                          02485000 T  0021:0156:1
                    INFO[LASTSEQROW,LASTSEQUENCE]);                                                  02486000 T  0021:0158:0
          XMODE:=1; SWITCHIT(VOIDTBIT);                                                              02487000 T  0021:0159:3
          GO EXIT;                                                                                   02488000 T  0021:0161:2
          END;                                                                                       02489000 T  0021:0163:2
     SWITCHIT(VOIDTBIT);                                                                             02490000 T  0021:0163:2
     VOIDPLACE:="9999"&"9999"[1:23];%2 B COMPATIBLE W/B-5700 VOIDS                                   02491000 T  0021:0163:3
     VOIDCR:=MKABS(VOIDPLACE); % AND FAKE OUT READACARD,                                             02492000 T  0021:0165:3
     GO AGAIN;                                                                                       02493000 T  0021:0167:2
     END;                                                                                            02494000 T  0021:0169:2
IF Q = "5CHECK" THEN                                                                                 02495000 T  0021:0169:2
     BEGIN SWITCHIT(CHECKBIT); GO AGAIN END;                                                         02496000 T  0021:0169:3
IF Q = "5LIMIT" THEN                                                                                 02497000 T  0021:0173:2
     BEGIN                                                                                           02498000 T  0021:0173:3
     SKAN;                                                                                           02499000 T  0021:0174:0
     IF RESULT#3 THEN % SHOULD BE NUMBER,                                                            02500000 T  0021:0178:0
          BEGIN FLAG(600); GO AGAIN END;                                                             02501000 T  0021:0178:1
     ERRMAX:=CONV(ACCUM[1],0,ACCUM[1].[12:6]);                                                       02502000 T  0021:0182:0
     GO SKANAGAIN;                                                                                   02503000 T  0021:0185:2
     END;                                                                                            02504000 T  0021:0185:3
IF Q = "5PUNCH" THEN                                                                                 02505000 T  0021:0185:3
     BEGIN SWITCHIT(PUNCHBIT); GO AGAIN; END;                                                        02506000 T  0021:0186:1
IF Q = "5PURGE" THEN                                                                                 02507000 T  0021:0190:0
     BEGIN SWITCHIT(PURGEBIT); GO AGAIN; END;                                                        02508000 T  0021:0190:1
```

```
      IF Q = "5LISTA" THEN                                                    02509000 T  0021!0194!0
        BEGIN                                                                 02510000 T  0021!0194!1
        SWITCHIT(LISTABIT);                                                   02511000 T  0021!0195!2
        GO AGAIN;                                                             02513000 T  0021!0196!0
        END;                                                                  02514000 T  0021!0198!0
      IF Q = "5STUFF" THEN                                                    02515000 T  0021!0198!0
        BEGIN SWITCHIT(STUFFBIT); GO AGAIN; END;                              02516000 T  0021!0198!1
      GO WHATISIT;                                                            02517000 T  0021!0202!0
LENGTH6:                                                                      02518000 T  0021!0202!1
      IF Q = "6SEQER" THEN                                                    02519000 T  0021!0203!2
        BEGIN SWITCHIT(SEQERRBIT); GO AGAIN END;                              02520000 T  0021!0203!3
      IF Q = "6SINGL" THEN                                                    02521000 T  0021!0207!2
        BEGIN SWITCHIT(SINGLBIT); GO AGAIN  END;                              02522000 T  0021!0207!3
      IF Q = "6SEQXE" THEN                                                    02523000 T  0021!0211!2
        BEGIN                                                                 02524000 T  0021!0211!3
          IF BUILDLINE.[45:1] THEN                                %108-       02525000 P  0021!0212!0
            BEGIN                                                 %108-       02525001 C  0021!0213!2
              IF XMODE = 0 THEN                                   %108-       02525003 C  0021!0213!3
                BEGIN                                             %108-       02525004 C  0021!0214!0
                  OPTIONWORD:=BOOLEAN(0);                         %108-       02525005 C  0021!0214!1
                  FOR SAVEINX:=1 STEP 2 UNTIL OPARSIZE DO         %108-       02525006 C  0021!0215!3
                    OPTIONS[SAVEINX]:=0;                          %108-       02525007 C  0021!0218!0
                  BUILDLINE.[47:1]:=SEQXEQTOG:=FALSE;             %108-       02525008 C  0021!0221!3
                  IF LASTUSED < 5 THEN LASTUSED:=1;               %108-       02525009 C  0021!0223!3
                  XMODE:=1;                                       %108-       02525010 C  0021!0225!3
                END;                                              %108-       02525011 C  0021!0226!1
              SEQXEQTOG:=XMODE ≠ 2 AND XMODE ≠ 4;                 %108-       02525012 C  0021!0226!1
              BUILDLINE.[47:1]:=SEQXEQTOG;                        %108-       02525013 C  0021!0228!1
            END;                                                  %108-       02526000 P  0021!0230!1
        GO SKANAGAIN;                                                         02527000 T  0021!0230!1
        END;                                                                  02528000 T  0021!0231!2
    IF Q = "6DEBUG" THEN                                                      02529000 T  0021!0231!2
        BEGIN                                                                 02530000 T  0021!0231!3
        SWITCHIT(DEBUGBIT);                                                   02531000 T  0021!0232!0
        IF DEBUGTOG THEN                                                      02533000 T  0021!0233!2
            IF WOP[0]=0 THEN                                                  02534000 T  0021!0233!3
                BEGIN                                                         02535000 T  0021!0235!2
                FILL WOP[*] WITH                                              02536000 T  0021!0235!3
        "LITC","    ",                                                        02537000 T  0021!0236!0
                                                                        START OF SEGMENT **********   24
        "OPDC","DESC",                                                        02538000 T  0021!0237!3
 10,"DFL ", 11,"NOP ", 12,"XRT ", 16,"ADD ", 17,"AD2 ", 18,"PRL ",            02539000 T  0021!0237!3
 19,"LNG ", 21,"GFQ ", 22,"BRC ", 24,"INX ", 35,"LOR ", 37,"GTR ",            02540000 T  0021!0237!3
 38,"BFC ", 39,"RTN ", 40,"COC ", 48,"SUB ", 49,"SB2 ", 64,"MUL ",            02541000 T  0021!0237!3
 65,"ML2 ", 67,"LND ", 68,"STD ", 69,"NEQ ", 71,"XIT ", 72,"MKS ",            02542000 T  0021!0237!3
128,"DIV ",129,"DV2 ",130,"COM ",131,"LQV ",132,"SND ",133,"XCH ",            02543000 T  0021!0237!3
134,"CHS ",167,"RTS ",168,"CDC ",197,"FTC ",260,"LOD ",261,"DUP ",            02544000 T  0021!0237!3
278,"LRC ",280,"SSF ",294,"LFC ",322,"ZP1 ",384,"IDV ",453,"FTF ",            02545000 T  0021!0237!3
515,"MDS ",532,"ISD ",533,"LEQ ",534,"BBW ",548,"ISN ",549,"LSS ",            02546000 T  0021!0237!3
550,"BFW ",581,"EQL ",582,"SSP ",584,"ECM ",709,"CTC ",790,"LBU ",            02547000 T  0021!0237!3
806,"LFU ",896,"RDV ",965,"CTF ",                                             02548000 T  0021!0237!3
        1023,1023,1023,1023,1023,1023,1023,1023,1023,1023,1023, 1023;         02549000 T  0021!0237!3
                                                                        24 IS  130 LONG, NEXT SEG   21
                FILL COP[*] WITH  % CHARACTER MODE MNEMONICS                  02550000 T  0021!0237!3
    "EXC ","NOP ","BSD ","BSS ","RDA ","TRW ","SED ","TDA ",                  02551000 T  0021!0238!0
                                                                        START OF SEGMENT **********   25
    "    ","    ","TBN ","    ","SDA ","SSA ","SFD ","SRD ",                  02552000 T  0021!0239!2
```

```
    "    ","    ","SES ","    ","TEQ ","TNE ","TEG ","TGR ",                                  02553000 T  0021:023:2
    "SRS ","SFS ","    ","    ","TEL ","TLS ","TAN ","BIT ",                                  02554000 T  0021:023:2
    "INC ","STC ","SEC ","CRF ","JNC ","JFC ","JNS ","JFW ",                                02555000 T  0021:023:2
    "RCA ","ENS ","BNS ","RSA ","SCA ","JRC ","TSA ","JRV ",                                02556000 T  0021:023:2
    "CEQ ","CNE ","CEG ","CGR ","BIS ","BIR ","OCV ","ICV ",                                02557000 T  0021:023:2
    "CEL ","CLS ","FSU ","FAD ","TRP ","TRN ","TRZ ","TRS ";                                02558000 T  0021:023:2
                                                                                  25 IS   64 LONG, NEXT SEG   21
        FILL POP[*] WITH                                                                     02559000 T  0021:023:2
 "ZFN ","ZBN ","ZFD ","ZBD ","ISO ",0,"DIA ","DIB ","TRB ","CFL ","CFE "                     02560000 T  0021:023:2
                                                                                  START OF SEGMENT **********   26
                        ;                                                                    02561000 T  0021:023:3
                                                                                  26 IS   11 LONG, NEXT SEG   21
                    END;                                                                     02562000 T  0021:024:2
        GO AGAIN;                                                                            02563000 T  0021:024:2
        END;                                                                                 02564000 T  0021:024:2
    IF Q = "6FORMA" THEN                                                                     02565000 T  0021:024:2
        BEGIN SWITCHIT(FORMATBIT); GO AGAIN; END;                                            02566000 T  0021:024:2
    GO WHATISIT;                                                                             02567000 T  0021:024:3
LENGTH7:                                                                                     02568000 T  0021:024:7:2
    IF Q = "7INCLU" THEN                                                                     02569000 T  0021:024:7:3
        BEGIN INCLUDECARD; GO EXIT; END;                                        %107-        02570000 P  0021:024:8:0
%   IF Q = "7INCLN" THEN                                                        %107-        02571000 P  0021:024:8:1
%       BEGIN SWITCHIT(NEWINCLBIT);  GO AGAIN; END;                                          02572000 T  0021:025:2:0
LENGTH8:                                                                                     02573000 T  0021:025:2:0
    IF Q ="8CODEF" THEN                                                         %106-        02574000 P  0021:025:2:0
        BEGIN SWITCHIT(CODEFILEBIT); GO AGAIN; END;                             %106-        02574100 C  0021:025:2:0
    GO WHATISIT;                                                                %106-        02574200 C  0021:025:2:1
LENGTH9:                                                                        %106-        02574300 C  0021:025:6:0
    IF Q = "9INTRI" THEN                                                                     02575000 T  0021:025:6:1
        BEGIN SWITCHIT(INTBIT);  GO AGAIN; END;                                              02576000 T  0021:025:7:2
WHATISIT:                                                                                    02577000 T  0021:025:7:3
    IF RESULT=3 THEN                                                                         02578000 T  0021:026:1:2
        BEGIN                                                                                02579000 T  0021:026:1:2
        BASENUM:=CONV(ACCUM[1],0,ACCUM[1],[12:6]);                                           02580000 T  0021:026:1:3
        TOTALNO:="10;                                                                        02581000 T  0021:026:2:0
        NEWBASE:=TRUE;                                                                       02582000 T  0021:026:5:3
        GO SKANAGAIN;                                                                        02583000 T  0021:026:6:1
        END;                                                                                 02584000 T  0021:026:7:2
    IF RESULT=2 THEN                                                                         02585000 T  0021:026:7:3
        BEGIN                                                                                02586000 T  0021:026:7:3
        IF Q = "1+0000" THEN                                                                 02587000 T  0021:026:8:1
            BEGIN                                                                            02588000 T  0021:026:9:2
            SKAN;                                                                            02589000 T  0021:026:9:3
            IF RESULT=3 THEN                                                                 02590000 T  0021:027:0:0
             ADDVALUE:=CONV(ACCUM[1],0,ACCUM[1],[12:6])                                      02591000 T  0021:027:4:0
         ELSE FLAG(600); % NUMBER EXPECTED.                                                  02592000 T  0021:027:4:1
            END;                                                                             02593000 T  0021:027:7:2
        GO SKANAGAIN;                                                                        02594000 T  0021:028:0:1
        END;                                                                                 02595000 T  0021:028:0:1
COMMENT  DID NOT RECOGNIZE OPTION;                                                           02596000 T  0021:028:1:2
    IF RESULT#1 THEN % NOT AN IDENTIFIER.                                                    02597000 T  0021:028:1:2
        BEGIN FLAG(601); GO SKANAGAIN END;                                                   02598000 T  0021:028:1:2
    SWITCHIT(USEROPINX); % USEROPINX MEANS A USER-DEFINED OPTION.                            02599000 T  0021:028:2:0
    GO AGAIN;                                                                            02600000 T  0021:028:3:3
 EXIT:                                                                                       02601000 T  0021:028:4:1
    LISTER:=DEBUGTOG OR LISTOG OR LISTATOG;                                                  02602000 T  0021:028:5:2
    MOVE(10,DEFINEARRAY[0],ACCUM[0]); % RESTORE INFORMATION FOR                              02602500 T  0021:028:5:2
                                                                                             02602600 T  0021:028:8:0
```

```
        COUNT:=SCOUNT; RESULT:=SRESULT;    % "TABLE" TO RESUME SCAN.                    02602700 T  00211029010
        DOLLARTOG:=FALSE;                                                              02603000 T  00211029113
        END DOLLARCARD;                                                                02604000 T  00211029211
                                                                                  21 IS  295 LONG, NEXT SEG    3


COMMENT  TABLE IS THE ROUTINE THAT MOST CODE IN THE COMPILER                           02605000 T  00031034210
      USES WHEN IT IS DESIRED TO SCAN ANOTHER LOGICAL QUANTITY.                        02606000 T  00031034210
      THE RESULT RETURNED IS THE CLASS OF THE ITEM DESIRED.                            02607000 T  00031034210
      TABLE MAINTAINS THE VARIABLES I AND NXTELBT AND THE ARRAY                        02608000 T  00031034210
      ELBAT.  ELBAT AND I ARE PRINCIPAL VARIABLES USED FOR                             02609000 T  00031034210
      COMUNICATION BETWEEN TABLE AND THE OUTSIDE WORLD.  NXTELBT                       02610000 T  00031034210
      IS ALMOST EXCLUSIVELY USED BY TABLE. ALTHOUGH AN OCCASION-                       02611000 T  00031034210
      AL OTHER USE IS MADE IN ORDER TO FORGET THAT SOMETHING WAS                       02612000 T  00031034210
      SCANNED.  (SEE. FOR EXAMPLE. COMPOUNDTAIL). FOR FURTHER                         02613000 T  00031034210
      GENERAL DISCUSSION SEE THE DECLARATION OF THESE VARIABLES.                       02614000 T  00031034210
        THE PARAMETER P IS THE ACTUAL INDEX OF THE QUANTITY                            02615000 T  00031034210
      DESIRED (USUALLY I-1, I, OR I+1).                                                02616000 T  00031034210
        THE GENERAL PLAN OF TABLE IS THIS:                                             02617000 T  00031034210
          I) IF P < NXTELBAT GO ON TO III).                                            02618000 T  00031034210
         II) PROCESS ONE QUANTITY.                                                     02619000 T  00031034210
            A) SCAN.                                                                   02620000 T  00031034210
            B) TEST FOR IDENTIFIER, NUMBER, OR SPECIAL CHARACTER.                      02621000 T  00031034210
              1) IDENTIFIER - LOOKUP IN DIRECTORY AND PROCESS                          02622000 T  00031034210
                   IN SPECIAL MANNER IF COMMENT OR DEFINED ID.                         02623000 T  00031034210
              2) NUMBER - PROCESS INTEGER PART, FRACTIONAL PART,                       02624000 T  00031034210
                   AND EXPONENT PART.                                                  02625000 T  00031034210
              3) TEST IF SPECIAL CHARACTER REQUIRES SPECIAL                            02626000 T  00031034210
                   PROCESSING - OTHERWISE GET ELBAT WORD FROM                          02627000 T  00031034210
                   SPECIAL.                                                            02628000 T  00031034210
            C) LOAD ELBAT AND INCREMENT NXTELBT.                                       02629000 T  00031034210
            D) IF ELBAT IS FULL ADJUST ELBAT, NXTELBT, I, AND P.                       02630000 T  00031034210
            E) GO BACK TO I).                                                          02631000 T  00031034210
         III) RETURN WITH CLASS OF ELBAT[P].                                           02632000 T  00031034210
        FURTHER DETAILS ARE GIVEN IN BODY OF TABLE.                                    02633000 T  00031034210
    ;                                                                                  02634000 T  00031034210
    INTEGER PROCEDURE TABLE(P); VALUE P; INTEGER P;                                    02635000 T  00031034210
PRT(533) = TABLE                                                                       
        BEGIN                                                                          02636000 T  00031034210
        LABEL PERCENT,SPECIALCHAR,COMPLETE,COLON,DOT,ATSIGN,QUOTE,                     02637000 T  00031034210
                                                                          START OF SEGMENT **********   27
              STRNGXT,MOVEIT,ARGH,FINISHNUMBER,                                        02638000 T  00271000010
              SCANAGAIN,FPART,EPART,IPART,IDENT,ROSE,COMPOST,DOLLAR,RTPAREN,           02639000 T  00271000010
              CROSSHATCH, DBLDOLLAR;                                                   02640000 T  00271000010
        SWITCH SPECIALSWITCH:=PERCENT,DOLLAR,DOT,ATSIGN,COLON,QUOTE,                   02641000 T  00271000010
                              RTPAREN,CROSSHATCH,DBLDOLLAR;                            02642000 T  00271000211
        SWITCH RESULTSWITCH:=IDENT,SPECIALCHAR,IPART;                                  02643000 T  00271000713
        WHILE P ≥ NXTELBT                                                              02644000 T  00271001211
        DO   BEGIN                                                                     02645000 T  00271001312
    SCANAGAIN:                                                                         02646000 T  00271001410
              COUNT:=RESULT:=ACCUM[1]:=0; SCANNER;                                     02647000 T  00271001512
              GO RESULTSWITCH[RESULT];                                                 02648000 T  00271001713
    ARGH:                                                                              02649000 T  00271001913
              Q:=ACCUM[1]; FLAG(14); GO SCANAGAIN;                                     02650000 T  00271002010
    SPECIALCHAR:                                                                       02651000 T  00271002210
```

```
          GT1:=ACCUM[1].[18:6] - 2;                                                  02652000 T  00271002:2
          ENDTOG:=GT1 = 57 AND ENDTOG;                                               02653000 T  00271002:2
COMMENT   OBTAIN ACTUAL CHARACTER FROM ACCUM;                                        02654000 T  00271002:1
          T:=SPECIAL[GT1&GT1[42:41:3]];                                              02655000 T  00271002:1
COMMENT   NOTICE COMPRESSION TECHNIQUE USED TO SHORTEN TABLE OF                      02656000 T  00271002:1
          ELBAT WORDS FOR SPECIAL CHARACTERS;                                        02657000 T  00271002:1
          IF GT1:=T.INCR = 0 THEN GO COMPLETE;                                       02658000 T  00271002:1
          GO SPECIALSWITCH[GT1];                                                     02659000 T  00271003:2
COMMENT   INCR FIELD OF SPECIAL CHARACTER IS NON-ZERO FOR SPECIAL                    02660000 T  00271003:2
          CHARACTERS REQUIRING SPECIAL HANDLING.  INCR IS SWITCHED                   02661000 T  00271003:2
          ON TO OBTAIN DISCRIMINATION;                                               02662000 T  00271003:2
COLON:    RESULT:=7; SCANNER; COMMENT ELIMINATE BLANKS - CHECKING                    02663000 T  00271003:2
                                      FOR := IN PLACE OF ← ;                         02664000 T  00271003:0
          IF EXAMIN (NCR) = "=" THEN                                                 02665000 T  00271003:0
               BEGIN RESULT:=0; SCANNER; T:=SPECIAL[13]  END;                        02666000 T  00271003:0
          RESULT:=2; GO COMPLETE;                                                    02667000 T  00271003:1
DOT:                                                                                 02668000 T  00271004:0
          IF EXAMIN(NCR)>9 OR ENDTOG THEN GO COMPLETE;                               02680000 T  00271004:0
          NHI:=NLO:=0;                                                               02681000 T  00271004:2
          C:=0; FSAVE:=0; GO FPART;                                                  02682000 T  00271004:0
ATSIGN:                                                                              02683000 T  00271004:0
%         RESULT:=0; SCANNER;                                                        02684000 T  00271004:2
%         IF COUNT>17 THEN GO ARGH;                                                  02685000 T  00271004:2
%         IF OCTIZE(ACCUM[1],COUNT-1,17-COUNT,C) THEN GO ARGH                        02686000 T  00271004:2
%    ELSE GO NUMBEREND;                                                              02687000 T  00271004:2
          NHI:=C:=1;  NLO:=FSAVE:=0;  GO EPART;                                      02688000 T  00271004:2
COMMENT   DOT AND ATSIGN ENTER NUMBER CONVERSION AT CORRECT SPOT;                    02689000 T  00271005:0
QUOTE:                                                                               02690000 T  00271005:0
          COUNT := 0; T := IF STREAMTOG THEN 63 ELSE 8;                  %111-       02691000 P  00271005:0
%                                                                                    02692000 T  00271005:2
%                                                                                    02692500 T  00271005:2
          DO   BEGIN                                                                 02693000 T  00271005:2
               RESULT:=5; SCANNER;                                                   02694000 T  00271005:2
               IF COUNT=T THEN                                                       02695000 T  00271005:0
                    IF EXAMIN(NCR) ≠ """ THEN GO ARGH;                               02696000 T  00271005:2
               END UNTIL EXAMIN(NCR) = """;                                          02697000 T  00271005:3
          IF NOT STREAMTOG AND COUNT=8 AND BOOLEAN(ACCUM[1].[18:1]) THEN             02697500 C  00271006:0
            BEGIN Q := ACCUM[1]; FLAG(254); GO TO SCANAGAIN; END;        %111-       02697600 C  00271006:1
          Q:=ACCUM[1]; RESULT:=5; SCANNER; COUNT:=COUNT-1;                           02698000 T  00271006:3
          IF COUNT<0 THEN COUNT:=COUNT+64;                                           02699000 T  00271006:2
          ACCUM[1]:=Q; RESULT:=4;                                                    02700000 T  00271007:3
STRNGXT:  T:=C:=0;                                                                   02701000 T  00271007:3
          T.CLASS:=STRNGCON;                                                         02702000 T  00271007:2
          IF COUNT < 8 OR (COUNT = 8 AND NOT BOOLEAN                 %111-           02703000 P  00271007:2
            (ACCUM[1].[18:1])) THEN % FLAG BIT NOT SET, FULL WORD CONST.             02703050 C  00271007:1
MOVEIT:                                                                              02704000 T  00271008:0
          MOVECHARACTERS(COUNT,ACCUM[1],3,C,8-COUNT)                                 02705000 T  00271008:2
     ELSE T.CLASS:=STRING;                                                           02705100 T  00271008:1
          T.INCR:=COUNT; GO COMPLETE;                                                02705200 T  00271008:3
%                                                                                    02706000 T  00271008:0
COMMENT   CROSSHATCH HANDLES TWO SITUATIONS;                                         02707000 T  00271008:0
          THE CROSSHATCH AT END OF DEFINE DECLARATIONS AND                           02708000 T  00271008:0
          THE CROSSHATCH AT END OF ALPHA REPRESENTING DEFINED IDS.                   02709000 T  00271008:0
          THE TWO CASES ARE PROCESSED DIFFERENTLY.  THE FIRST CASE                   02710000 T  00271008:0
          MERELY PLACES THE CROSSHATCH IN ELBAT.  THE SECOND CASE                    02711000 T  00271008:0
          CAUSES AN EXIT FROM SCANNING THE ALPHA FOR THE DEFINED ID.                 02712000 T  00271008:0
          FOR A FULL DISCUSSION SEE DEFINEGEN;                                       02713000 T  00271008:0
CROSSHATCH:                                                                          02714000 T  00271008:0
```

rf 02903000

```
                IF DEFINECTR≠0 THEN GO COMPLETE;                                                02715000 T  0027:0088:0
                PUTSEQNO(GT1,LCR);                                                              02716000 T  0027:0089:2
                TURNONSTOPLIGHT(0,LCR);                                                         02717000 T  0027:0090:0
                IF DEFINEINDEX = 0 THEN GO ARGH;                                                02718000 T  0027:0091:2
                LCR:=(GT1:=DEFINEARRAY[DEFINEINDEX-1]) DIV 262144;                              02719000 T  0027:0092:1
                NCR:=GT1 MOD 262144;                                                            02720000 T  0027:0095:2
                LASTUSED:=(T:=DEFINEARRAY[DEFINEINDEX:=DEFINEINDEX-3]).[33:15];                 02721000 P  0027:0096:0
                IF (GT2 := T.[18:15]) ≠ 0 THEN % THIS WAS A PARAMETRIC DEFINE                   02721500 C  0027:0098:1
                  BEGIN % PURGING PARAMETERS FROM DEFSTACKHEAD                          %122-   02722000 P  0027:0101:2
                    GT2 := TAKE(GT2).LINK; % GET POINTER TO NEW DEFSTACKHEAD                    02722500 C  0027:0101:3
                    DO                                                                  %122-   02723000 P  0027:0103:2
                      PUT(TEXT[(NEXTTEXT:=(GT1:=TAKE(DEFSTACKHEAD)).DYNAM-1)                    02723500 C  0027:0104:0
                            .LINKR.NEXTTEXT.LINKC].DEFSTACKHEAD)                        %122-   02724000 P  0027:0106:0
                        % THIS RESTORES THE PREVIOUS ELBAT WORD FOR                     %122-   02724500 C  0027:0109:2
                        % THIS PARAMETER IN CASE OF NESTED DEFINE.                      %122-   02725000 P  0027:0109:2
                    UNTIL                                                               %122-   02725500 C  0027:0109:2
                      GT2 = (DEFSTACKHEAD := GT1.LINK);                                 %122-   02726000 P  0027:0109:3
                  END;                                                                  %122-   02727000 P  0027:0111:3
                GO SCANAGAIN;                                                                   02728000 T  0027:0111:3
DOLLAR:         COMMENT THIS CODE HANDLES CONTROL CARDS;                                        02729000 T  0027:0114:0
                IF GT1:=EXAMIN(NCR)="$" THEN GO DBLDOLLAR ELSE DOLLARCARD;                      02730000 T  0027:0114:0
PERCENT:        IF NCR ≠ FCR THEN READACARD;                                                    02731000 T  0027:0117:2
                IF LIBINDEX≠0 THEN                                                              02732000 T  0027:0119:3
                 IF RECOUNT=FINISHPT THEN                                                       02733000 T  0027:0120:1
                     BEGIN                                                                      02734000 T  0027:0121:3
                     SEARCHLIB(FALSE); READACARD; NORELEASE:=FALSE                              02735000 T  0027:0122:0
                     END;                                                                       02736000 T  0027:0123:3
                GO SCANAGAIN;                                                                   02737000 T  0027:0124:0
COMMENT         MOST PERCENT SIGNS ACTING AS END OF CARD SENTINELS GET TO                       02738000 T  0027:0124:1
                PERCENT.  PERCENT READS THE NEXT CARD AND STARTS OVER.  A                       02739000 T  0027:0124:1
                SIDE EFFECT IS THAT ALL CHARACTERS ON A CARD ARE IGNORED                        02740000 T  0027:0124:1
                AFTER A FREE PERCENT SIGN (ONE NOT IMBEDDED IN A STRING OR                      02741000 T  0027:0124:1
                COMMENT);                                                                       02742000 T  0027:0124:1
COMMENT         MIGHT BE FUNNY COMMA - HANDLE HERE;                                             02743000 T  0027:0124:1
RTPAREN;        RESULT:=7; SCANNER;                                                             02744000 T  0027:0124:1
                IF EXAMIN(NCR) = "'" THEN                                                       02745000 T  0027:0126:0
                     BEGIN                                                                      02746000 T  0027:0128:0
                     RESULT:=0; SCANNER;                                                        02747000 T  0027:0128:1
                     DO   BEGIN                                                                 02748000 T  0027:0129:3
                          RESULT:=5; SCANNER                                                    02749000 T  0027:0130:0
                          END UNTIL EXAMIN(NCR) = "'";                                          02750000 T  0027:0130:1
                     RESULT:=0; SCANNER;                                                        02751000 T  0027:0133:3
                     RESULT:=7; SCANNER;                                                        02752000 T  0027:0134:1
                     IF EXAMIN(NCR) ≠ "(" THEN GO ARGH;                                         02753000 T  0027:0136:0
                     RESULT:=0; SCANNER;  Q:=ACCUM[1];                                          02754000 T  0027:0138:0
                     T:=SPECIAL[24]                                                             02755000 T  0027:0140:1
                     END;                                                                       02756000 T  0027:0140:1
                RESULT:=2; GO COMPLETE;                                                         02757000 T  0027:0141:3
IPART:          TCOUNT:=FSAVE:=0; C:=CONVERT;                                                   02758000 T  0027:0142:1
                RESULT:=7; SCANNER; % DEBLANK.                                                  02759000 T  0027:0145:2
                IF DEFINECTR=0 THEN                                                             02760000 T  0027:0146:1
                 IF (C=3 OR C=4) AND EXAMIN(NCR)="'" THEN %OCTAL OR HEX STRING.                 02761000 T  0027:0147:2
                  IF NOT (ACCUM[0].CLASS=FILEID AND INFO[LASTINFO.    % 1641                   02761500 T  0027:0151:3
                     LINKR. LASTINFO.LINKC]  = ACCUM[0])THEN          % 1641                   02761501 T  0027:0153:3
                     BEGIN    INTEGER SIZ;                                                      02762000 T  0027:0157:2
PRT(534) = *SEGMENT DESCRIPTOR*
```

```
STACK(F+3) = SIZ
                        RESULT:=5; SCANNER; % SKIP QUOTE.                                  02763000 T  0028:0000:0
                        COUNT:=0;                                                          02764000 T  0028:0001:2
                        DO   BEGIN                                                         02765000 T  0028:0002:0
                             RESULT:=5; SCANNER;                                           02766000 T  0028:0002:0
                             IF COUNT > SIZ:=48 DIV C THEN % > 1 WORD LONG.                02767000 T  0028:0003:2
                                  BEGIN  ERR(520); GO SCANAGAIN  END;                      02768000 T  0028:0005:2
PRT(535) = SCANAGAIN
                             END UNTIL EXAMIN(NCR)="'";                                    02769000 T  0028:0008:1
                        Q:=ACCUM[1]; RESULT:=5; SCANNER; COUNT:=COUNT-1;                   02770000 T  0028:0010:1
                        IF C=3 THEN % OCTAL STRING.                                        02771000 T  0028:0014:0
                          IF OCTIZE(ACCUM[1],ACCUM[4],16-COUNT,COUNT) THEN                 02772000 T  0028:0015:2
                            FLAG(521) % NON-OCTAL CHARACTER IN STRING.                     02773000 T  0028:0018:1
                      ELSE ELSE IF HEXIZE(ACCUM[1],ACCUM[4],12-COUNT,COUNT) THEN           02774000 T  0028:0019:3
                             FLAG(521); % NON-HEX CHARACTER IN HEX STRING.                 02775000 T  0028:0024:1
                        T.INCR := COUNT := (C*COUNT-1)DIV 6 + 1; % # OF CHARS.            02776100 C  0028:0026:0
                        T.CLASS:= STRNGCON;                                    %111-       02776200 C  0028:0030:0
                        MOVECHARACTERS(1,ACCUM[4],0,ACCUM[1],3);               %111-       02776300 C  0028:0032:0
                    IF BOOLEAN(ACCUM[1],F18:1]) THEN % FLAG BIT SET.           %111-       02776400 C  0028:0034:0
                      IF STREAMTOG THEN                                        %111-       02776500 C  0028:0035:2
                          T.CLASS := STRING                                    %111-       02776600 C  0028:0036:0
                        ELSE                                                   %111-       02776700 C  0028:0037:2
                          FLAG(254)                                            %111-       02776800 C  0028:0038:0
                        ELSE                                                   %111-       02776900 C  0028:0039:2
                          C := ACCUM[4]; % GET FULL WORD EQUIVALENT OF STRING.              02777000 P  0028:0039:3
                        MOVECHARACTERS(COUNT,ACCUM[4],8-COUNT,ACCUM[1],3);     %111-       02777050 C  0028:0041:2
                        GO TO COMPLETE;                                        %111-       02777100 C  0028:0043:3
PRT(536) = COMPLETE
                        MOVECHARACTERS(8,ACCUM[4],0,ACCUM[1],3);                           02781000 T  0028:0046:0
                        GO COMPLETE;                                                       02782000 T  0028:0048:0
                        END OCTAL OR HEX STRING;                                           02783000 T  0028:0050:1
PRT(537) = *SEGMENT DESCRIPTOR*
                                                                          28 IS   51 LONG, NEXT SEG   27
                    IF DPTOG THEN                                                  02784000 T  0027:0158:0
                        BEGIN NHI:=THI; NLO:=TLO; END;                             02785000 T  0027:0158:0
                    IF EXAMIN(NCR)="," THEN                                        02786000 T  0027:0160:0
                        BEGIN                                                      02787000 T  0027:0162:0
                        RESULT:=0; SCANNER;                                        02788000 T  0027:0162:1
                        C:=1.0* C;                                                 02789000 T  0027:0163:3
                FPART:  TCOUNT:=COUNT;                                             02790000 T  0027:0165:2
                        IF EXAMIN(NCR)≤9 THEN                                      02791000 T  0027:0165:3
                            BEGIN                                                  02792000 T  0027:0167:3
                            RESULT:=0; SCANNER;                                    02793000 T  0027:0168:0
                            IF DPTOG THEN                                          02794000 T  0027:0169:2
                                BEGIN                                              02795000 T  0027:0169:3
                                DOUBLE(CONVERT,TLO,TEN[(COUNT-TCOUNT)MOD 12],      02796000 T  0027:0170:0
                                                   0,/,:=,THI,TLO);                02797000 T  0027:0172:1
                                FOR T:=12 STEP 12 UNTIL COUNT - TCOUNT DO          02798000 T  0027:0174:0
                                  DOUBLE(THI,TLO,TEN[12],0,/,:=,THI,TLO);          02799000 T  0027:0178:1
                                DOUBLE(THI,TLO,NHI,NLO,+,:=,NHI,NLO);              02800000 T  0027:0183:2
                                C:=NHI                                             02801000 T  0027:0185:2
                                END                                                02802000 T  0027:0185:2
                            ELSE C:=CONVERT+C*TEN[FSAVE:=COUNT-TCOUNT];            02803000 T  0027:0186:0
                            END                                                    02804000 T  0027:0189:3
                        END;                                                       02805000 T  0027:0189:3
                    RESULT:=7; SCANNER;                                            02806000 T  0027:0189:3
                    IF EXAMIN(NCR)="@" THEN                                        02807000 T  0027:0191:2
                        BEGIN                                                      02808000 T  0027:0192:1
```

```
FPART:      RESULT:=0; SCANNER;                                                  02809000 T  0027:0193:2
            TCOUNT:=COUNT;                                                       02810000 T  0027:0194:1
            C:=C×1.0;                                                            02811000 T  0027:0195:3
            RESULT:=7; SCANNER;                                                  02812000 T  0027:0197:2
            IF T:=EXAMIN(NCR)>9 THEN                                             02813000 T  0027:0198:0
             IF T="-" OR T = "+" THEN                                            02814000 T  0027:0200:1
                 BEGIN                                                           02815000 T  0027:0202:1
                 RESULT:=0; SCANNER;                                             02816000 T  0027:0203:2
                 TCOUNT:=COUNT;                                                  02817000 T  0027:0204:1
                 END                                                             02818000 T  0027:0205:2
            ELSE FLAG(47);                                                       02819000 T  0027:0205:2
            RESULT:=0; SCANNER;                                                  02820000 T  0027:0207:3
            IF RESULT ≠ 3 THEN FLAG (47);COMMENT NOT A NUMBER;                   02821000 T  0027:0209:2
            Q:=ACCUM[1];                                                         02822000 T  0027:0211:2
            IF GT1:=T:=(IF T="-"THEN -CONVERT ELSE CONVERT)<=46 OR               02823000 T  0027:0212:0
              T>69 THEN FLAG(269)                                                02824000 T  0027:0216:1
            ELSE BEGIN                                                           02825000 T  0027:0218:1
                 T:=TEN[ABS(GT3:=T-FSAVE)];                                      02826000 T  0027:0219:3
                 IF ABS(0&C[42:3:6]&C[1:2:1]+0&T[42:3:6]&GT3[1:1:1]              02827000 T  0027:0221:3
                   + 12) >63 THEN FLAG(269)                                      02828000 T  0027:0226:0
            ELSE IF DPTOG THEN                                                   02829000 T  0027:0228:1
                   IF GT1<0 THEN                                                 02830000 T  0027:0229:3
                      BEGIN                                                      02831000 T  0027:0231:2
                      GT1:=-GT1;                                                 02832000 T  0027:0231:3
                      DOUBLE(NHI,NLO,TEN[GT1 MOD 12],0,/,:=,NHI,NLO);            02833000 T  0027:0232:1
                      FOR GT2:=12 STEP 12 UNTIL GT1 DO                           02834000 T  0027:0235:3
                        DOUBLE(NHI,NLO,TEN[12],0,/,:=,NHI,NLO);                  02835000 T  0027:0237:2
                      END                                                        02836000 T  0027:0242:0
                 ELSE BEGIN                                                      02837000 T  0027:0242:0
                      DOUBLE(NHI,NLO,TEN[GT1 MOD 12],0,×,:=,NHI,NLO);            02838000 T  0027:0242:1
                      FOR GT2:=12 STEP 12 UNTIL GT1 DO                           02839000 T  0027:0245:3
                        DOUBLE( NHI,NLO,TEN[12],0,×,:=,NHI,NLO);                 02840000 T  0027:0247:2
                      END                                                        02841000 T  0027:0252:0
                 ELSE C:=IF GT3<0 THEN C/T ELSE C×T;                             02842000 T  0027:0252:0
                      END;                                                       02843000 T  0027:0256:0
                 END                                                             02844000 T  0027:0256:0
            ELSE IF FSAVE ≠ 0 THEN C:=C/TEN[FSAVE];                              02845000 T  0027:0256:0
            Q:=ACCUM[1]; RESULT:=3;                                              02846000 T  0027:0259:3
FINISHNUMBER:                                                                    02847000 T  0027:0261:2
            T:=0;                                                                02848000 T  0027:0262:0
            IF C.[1:37]=0 THEN                                                   02849000 T  0027:0262:1
                 BEGIN   T.CLASS:=LITNO ; T.ADDRESS:=C  END                      02850000 T  0027:0264:1
            ELSE T.CLASS:=NONLITNO ;                                             02851000 T  0027:0268:1
            GO COMPLETE;                                                         02852000 T  0027:0270:1
COMMENT     THE CODE BETWEEN IDENT AND COMPOST DOES A LOOKUP IN INFO.            02853000 T  0027:0271:2
            IF QUANTITY IS NOT FOUND THE ELBAT WORD EXPECTS TO BE                02854000 T  0027:0271:2
            ZERO.  THE SCRAMBLE FOR APPROPRIATE STACK IS FIRST THING             02855000 T  0027:0271:2
            TO BE DONE.  THEN A CHECK IS MADE, USING SUPERSTACK,                 02856000 T  0027:0271:2
            TO DETERMINE WHETHER THE IDENTIFIER IS ONE OF OUR                    02857000 T  0027:0271:2
            COMMON RESERVED WORDS.  IF IT IS, EXIT IS MADE TO                    02858000 T  0027:0271:2
            COMPLETE, OTHERWISE THE LOOP BETWEEN COMPOST AND                     02859000 T  0027:0271:2
            ROSE IS ENTERED.  THE LAST THING DONE FOR ANY                        02860000 T  0027:0271:2
            IDENTIFIER WHICH IS FOUND IS TO STUFF THE LOCATION                   02861000 T  0027:0271:2
            OF THE ELBATWORD IN INFO INTO THE LINK FIELD.  THIS                  02862000 T  0027:0271:2
            ALLOWS REFERENCE BACK TO INFO FOR ADDITIONAL DATA.                   02863000 T  0027:0271:2
            SHOULD THIS BE REQUIRED.                                ;            02864000 T  0027:0271:2
IDENT:      IF T:=SUPERSTACK[SCRAM:=(Q:=ACCUM[1])MOD 125]≠0 THEN                 02865000 T  0027:0271:2
```

```
                BEGIN                                                                    02866000 T  0027:027:2
                IF INFO[GT1:=T.LINKR+(GT2:=T.LINKC)+1]=Q THEN                            02867000 T  0027:027:3
                     BEGIN                                                               02868000 T  0027:028:0
                     T:=INFO[GT1,GT2]&T[35:35:13];                                       02869000 T  0027:028:1
                     GO COMPLETE                                                         02870000 T  0027:028:2
                     END                                                                 02871000 T  0027:028:3
                END;                                                                     02872000 T  0027:028:3
           IF EXAMINELAST(ACCUM[1], COUNT+2) = 12 THEN T:=DEFSTACKHEAD                   02873000 T  0027:028:3
      ELSE T:=STACKHEAD[SCRAM];                                                          02874000 T  0027:028:2
ROSE: GT1:=T.LINKR;                                                                      02875000 T  0027:028:2
           IF(GT2:=T.LINKC)+GT1= 0 THEN                                                  02876000 T  0027:029:2
                BEGIN T:=0; GO COMPLETE END;                                             02877000 T  0027:029:3
         IF T = INFO[GT1, GT2] THEN BEGIN                                    %101-       02877010 C  0027:029:2
           T:= 0; GO TO COMPLETE END;                                        %101-       02877020 C  0027:029:3
           T:=INFO[GT1,GT2];                                                             02878000 T  0027:029:1
           IF INFO[GT1,GT2+1]&0[1:1:11] ≠ Q THEN GO ROSE;                                02879000 T  0027:030:1
           IF COUNT ≤ 5 THEN GO COMPOST ;                                                02880000 T  0027:030:0
           IF NOT EQUAL(COUNT=5,ACCUM[2],INFO[GT1,GT2+2])THEN GO ROSE;                   02881000 T  0027:030:3
COMPOST: T:=T&GT1[35:43:5]&GT2[40:40:8];                                                 02882000 T  0027:031:0
           IF GT1 ≠1 AND NOT MACROID THEN % NOT RESERVED WORD                %116-       02882100 P  0027:031:1
             XREFIT(T.LINK,CARDNUMBER,NORMALREF); % BUILD XREF ENTRY          %116-       02882200 P  0027:031:0
COMMENT  CHECK HERE FOR COMMENTS AND DEFINED IDS;                                        02883000 T  0027:031:3
           IF NOT ENDTOG THEN                                                            02884000 T  0027:031:3
                BEGIN                                                                    02885000 T  0027:031:0
                IF GT1:=T.CLASS = COMMENTV THEN                                          02886000 T  0027:031:1
                     BEGIN                                                               02887000 T  0027:032:1
                     WHILE EXAMIN(NCR) ≠ ";" DO                                          02888000 T  0027:032:2
                          BEGIN RESULT:=6; COUNT:=0; SCANNER END;                        02889000 T  0027:032:2
                     RESULT:=0;SCANNER;GO SCANAGAIN                                      02890000 T  0027:032:3
                     END                                                                 02891000 T  0027:032:3
                END;                                                                     02892000 T  0027:032:3
           IF STOPDEFINE THEN GO COMPLETE;                                               02893000 T  0027:032:3
           IF GT1 ≠ DEFINEDID THEN GO COMPLETE;                                          02894000 T  0027:032:1
COMMENT  SETUP FOR DEFINED IDS - SEE DEFINEGEN FOR MORE DETAILS;                         02895000 T  0027:032:3
           IF BOOLEAN(T,MON) THEN % THIS IS A PARAMETRIC DEFINE                          02896000 T  0027:032:3
             GT1:=GIT(T:=FIXDEFINEINFO(T)) ELSE GT1:=0;                                  02897000 T  0027:033:1
           IF DEFINEINDEX = 24 THEN                                                      02898000 T  0027:033:1
                BEGIN FLAG(139);GO ARGH END;                                             02899000 T  0027:033:2
           DEFINEARRAY[DEFINEINDEX]:=LASTUSED & GT1[18:33:15];                           02900000 T  0027:033:2
           LASTUSED:=T.DYNAM;                                                            02901000 T  0027:033:1
           DEFINEARRAY[DEFINEINDEX+2]:=262144×LCR+NCR;                                   02902000 T  0027:034:0
           LCR:=(NCR:=MKABS(DEFINEARRAY[DEFINEINDEX+1]))+1;                              02903000 T  0027:034:1
           PUTSEQNO(GT4,LCR);                                                            02904000 T  0027:034:6:0
           TURNONSTOPLIGHT("%",LCR); DEFINEINDEX:=DEFINEINDEX+3;                         02905000 T  0027:034:7:2
           GO PERCENT;                                                                   02906000 T  0027:034:9:3
DBLDOLLAR:                                                                               02907000 T  0027:035:1:2
           MAKCAST; GO SCANAGAIN;                                                        02908000 T  0027:035:1:2
COMPLETE:                                                                                02909000 T  0027:035:5:2
           ELBAT[NXTELBT]:=T;                                                            02910000 T  0027:035:5:2
      IF NOT DEFINING THEN                                                               02910100 T  0027:035:6:0
      IF T.CLASS = BEGINV THEN                                                           02910200 T  0027:035:6:1
           BEGINSTACK[BSPOINT:=BSPOINT+1]:=CARDNUMBER ELSE                               02910300 T  0027:035:8:1
      IF T.CLASS = ENDV THEN                                                             02910400 T  0027:036:1:2
    BEGIN                                                                                02910500 T  0027:036:3:2
           IF LISTER THEN IF BEND THEN BEGINPRINT;                                       02910600 T  0027:036:3:3
           BSPOINT:=BSPOINT - REAL(BSPOINT > 0); % PREVENT INVALID INDEX                 02910700 T  0027:036:6:0
    END;                                                                                 02910800 T  0027:036:7:3
```

```
                    STOPDEFINE:=FALSE; COMMENT ALLOW DEFINES AGAIN;                                  02911000 T  0027:036713
                    IF NXTELBT:=NXTELBT+1 > 74 THEN                                                 02912000 T  0027:036811
                      IF NOT MACROID THEN                                                           02913000 T  0027:037010
                        BEGIN                                                                       02914000 T  0027:037112
                COMMENT  ELBAT IS FULL:  ADJUST IT;                                                 02915000 T  0027:037113
                        MOVE(10,ELBAT[65],ELBAT);                                                   02916000 T  0027:037113
                        I:=I-65; P:=P-65; NXTELBT:=10;                                              02917000 T  0027:037313
                        END                                                                         02918000 T  0027:037712
                    END;                                                                            02919000 T  0027:037712
                IF TABLE:=ELBAT[P].CLASS = COMMENTV THEN                                            02920000 T  0027:037713
                    BEGIN                                                                           02921000 T  0027:038010
                COMMENT  SPECIAL HANDLING OF CONSTANTS FOR SAKE OF FOR STATEMENTS;                  02922000 T  0027:038011
                    C:=INFO[0,ELBAT[P].ADDRESS];                                                    02923000 T  0027:038011
                    ELBAT[P].CLASS:=TABLE:=NONLITNO                                                 02924000 T  0027:038312
                    END;                                                                            02925000 T  0027:038410
                STOPDEFINE:=FALSE;  COMMENT ALLOW DEFINE;                                           02926000 T  0027:038610
                END TABLE ;                                                                         02927000 T  0027:038611
                                                                                          27 IS  394 LONG, NEXT SEG    3



                INTEGER PROCEDURE MOVEANDBLOCK(FROM,SIZE,NAME);                    %106-        02927100 C  0003:034210
PRT(540) = MOVEANDBLOCK
                    VALUE SIZE,NAME; REAL SIZE,NAME; ARRAY FROM[0,0];              %106-        02927110 C  0003:034210
                    BEGIN                                                          %106-        02927120 C  0003:034210
                    INTEGER NSEGS,I,J,K;                                           %106-        02927130 C  0003:034210
                                                                                        START OF SEGMENT **********   29
STACK(F+3) = NSEGS
STACK(F+4) = I
STACK(F+5) = J
STACK(F+6) = K
                    ARRAY A[0:14];                                                 %106-        02927140 C  0029:000010
STACK(F+7) = A
                    SWITCH FORMAT FMT :=                                           %106-        02927150 C  0029:000113
                                                                                        START OF SEGMENT **********   30
PRT(541) = FMT
                        (/,"FILE PARAMETER BLOCK IS CODE FILE SEGMENT",I5,/),       %106-        02927160 C  0030:000113
                        (/,"SEGMENT DICTIONARY IS CODE FILE SEGMENT",I5,/),         %106-        02927170 C  0030:000113
                        (/,"PROGRAM-LINE DICTIONARY IS CODE FILE SEGMENT",I5,/),    %106-        02927180 C  0030:000113
                        (/,"PROGRAM REFERENCE TABLE IS CODE FILE SEGMENT",I5,/),    %106-        02927190 C  0030:000113
                        (/,"SEGMENT-LINE DICTIONARY IS CODE FILE SEGMENT",I5,/),    %106-        02927200 C  0030:000113
                        (/,"POWER OF TEN ARRAY IS CODE FILE SEGMENT",I5,/),         %106-        02927210 C  0030:000113
                        (/,"SEGMENT ZERO",I*,/),                                   %106-        02927220 C  0030:000113
                        (/,"SEGMENT NUMBER",I5," IS CODE FILE SEGMENT",I5,/);      %106-        02927230 C  0030:000113
                                                                                          30 IS  111 LONG, NEXT SEG   29
                    STREAM PROCEDURE OCTALWORDS(N,W,S,D); VALUE N,W;               %106-        02927240 C  0029:000113
PRT(542) = OCTALWORDS
                    BEGIN                                                          %106-        02927250 C  0029:000113
                    D:=D; DS:="LIT" ";                                             %106-        02927260 C  0029:000312
                    S:=LOC N; SI:=SI+6;                                            %106-        02927270 C  0029:000313
                    4(DS:=3 RESET; 3(IF SB THEN DS:=SET ELSE DS:=RESET; SKIP SB));             02927272 C  0029:000410
                    D:=DI-4; DS:=3 FILL;                                           %106-        02927280 C  0029:000712
                    D:=D; DI:=DI+5; DS:=4 LIT" ";                                  %106-        02927290 C  0029:000713
                    SI:=S;                                                         %106-        02927300 C  0029:000811
                    W(2(8(DS:=3 RESET;                                             %106-        02927310 C  0029:000912
                        3(IF SB THEN DS:=SET ELSE DS:=RESET; SKIP SB));            %106-        02927320 C  0029:001011
```

```
                    );
                  DS:=LIT" ");
                DS:=2 LIT" ");
              END OF OCTALWORDS;                                                %106-        02927330 C  0029:0012:1
                                                                                %106-        02927340 C  0029:0013:2
                                                                                %106-        02927350 C  0029:0013:3
                                                                                %106-        02927360 C  0029:0014:1


              %**********     S T A R T     **********                          %106-        02927370 C  0029:0014:1
              NSEGS:=(SIZE+29) DIV 30;                                          %106-        02927380 C  0029:0014:1
              IF DA DIV CHUNK < T:=(DA+NSEGS) DIV CHUNK THEN                    %106-        02927390 C  0029:0016:1
                 DA:=CHUNK*T;                                                   %106-        02927400 C  0029:0019:3
              MOVEANDBLOCK:=DA;                                                 %106-        02927410 C  0029:0021:2
              IF CODEFILE THEN                                                  %106-        02927420 C  0029:0022:0
                 IF NAME≥0 THEN                                                 %106-        02927430 C  0029:0022:1
                    WRITE(LINE,FMT[NAME],DA)                                    %106-        02927440 C  0029:0024:0
PRT(543) = *LIST, LABEL, OR SEGMENT DESCRIPTOR*
                 ELSE                                                           %106-        02927450 C  0029:0033:2
                    WRITE(LINE,FMT[7],ABS(NAME),DA);                            %106-        02927460 C  0029:0033:2
PRT(544) = *LIST, LABEL, OR SEGMENT DESCRIPTOR*
              IF SIZE≠0 THEN                                                    %106-        02927470 C  0029:0045:2
                 BEGIN                                                          %106-        02927480 C  0029:0046:0
                 FOR J:=0 STEP 30 WHILE J < SIZE DO                             %106-        02927490 C  0029:0046:1
                    BEGIN                                                       %106-        02927500 C  0029:0050:1
                    IF (K:=(128-(J MOD 128))) < 30 THEN                         %106-        02927510 C  0029:0050:1
                       BEGIN                                                    %106-        02927520 C  0029:0052:1
                       MOVE(K,FROM[J DIV 128,J MOD 128],CODE(0));               %106-        02927530 C  0029:0053:2
                       MOVE(30-K,FROM[(J DIV 128)+1,0],CODE(K));                %106-        02927540 C  0029:0059:3
                       END                                                      %106-        02927550 C  0029:0066:0
                    ELSE                                                        %106-        02927560 C  0029:0066:0
                       MOVE(30,FROM[J DIV 128,J MOD 128],CODE(0));              %106-        02927570 C  0029:0066:0
                    IF J+30 > SIZE THEN % ZERO OUT UNUSED SECTION               %106-        02927580 C  0029:0073:2
                       BEGIN                                                    %106-        02927590 C  0029:0074:0
                       K:=0;                                                    %106-        02927600 C  0029:0074:1
                       MOVE(1,K,CODE(SIZE-J));                                  %106-        02927610 C  0029:0075:3
                       IF (SIZE-J) < 29 THEN % MORE THAN ONE WORD               %106-        02927612 C  0029:0080:0
                          MOVE(29-SIZE+J,CODE(SIZE-J),CODE(SIZE-J+1));          %106-        02927620 C  0029:0081:3
                       END;                                                     %106-        02927630 C  0029:0091:3
                    IF CODEFILE THEN                                            %106-        02927640 C  0029:0091:3
                       BEGIN                                                    %106-        02927650 C  0029:0092:0
                       FOR K:=0 STEP 5 WHILE K≤25 AND (J+K)≤SIZE DO             %106-        02927660 C  0029:0092:1
                          BEGIN                                                 %106-        02927670 C  0029:0098:0
                          BLANKET(14,A);                                        %106-        02927680 C  0029:0098:0
                          OCTALWORDS(J+K,IF (J+K+5)≤SIZE THEN 5 ELSE            %106-        02927690 C  0029:0099:3
                                SIZE-J-K,CODE(K),A);                            %106-        02927700 C  0029:0103:3
                          WRITE(LINE,15,A[*]);                                  %106-        02927710 C  0029:0108:1
                          END;                                                  %106-        02927720 C  0029:0112:1
                       WRITE(LINE);                                             %106-        02927722 C  0029:0113:2
                       END;                                                     %106-        02927730 C  0029:0117:2
                    WRITE(CODE[DA]); DA:=DA+1;                                  %106-        02927740 C  0029:0117:2
                    END;                                                        %106-        02927750 C  0029:0123:2
                 END;                                                           %106-        02927760 C  0029:0123:3
              END OF MOVEANDBLOCK;                                              %106-        02927770 C  0029:0123:3
                                                                                                29 IS  129 LONG. NEXT SEG    3


COMMENT  NEXTENT IS THE PROCEDURE WHICH SCANS FOR THE FORMAT GENERATOR.                      02928000 T  0003:0342:0
```

```
                    IT USES THE SAME SCANNER AS THE TABLE ROUTINE.  NEXTENT                  02929000 T  0003!034210
                    PLACES EITHER A CHARACTER OR A CONVERTED NUMBER WITH A                   02930000 T  0003!034210
                    NEGATIVE SIGN IN ELCLASS.  NEXTENT SUPPRESSES BLANKS;                    02931000 T  0003!034210
               PROCEDURE NEXTENT;                                                            02932000 T  0003!034210
PRT(545) = NEXTENT
                    BEGIN     LABEL DEBLANK;                                                 02933000 T  0003!034210
                                                                                  START OF SEGMENT **********   31
                    COUNT:=ACCUM[1]:=0; LASTELCLASS:=ELCLASS;                                02934000 T  0031!000010
               DEBLANK;                                                                      02935000 T  0031!000021
                    IF EXAMIN(NCR)=" " THEN                                                  02936000 T  0031!000312
                         BEGIN                                                               02937000 T  0031!000411
                         RESULT:=7; SCANNER;                                                 02938000 T  0031!000512
                         END;                                                                02939000 T  0031!000611
                    IF EXAMIN(NCR) ≤ 9 THEN % WE HAVE A NO. (WORD MODE COLLATING SEQ.)       02940000 T  0031!000611
                         BEGIN                                                               02941000 T  0031!000810
                         RESULT:=3; SCANNER; TCOUNT:=0; Q:=ACCUM[1];                         02942000 T  0031!000811
                         IF COUNT>4 THEN FLAG(140) % INTEGER > 1023.                         02943000 T  0031!001113
                    ELSE  IF ELCLASS:="CONVERT < "1023 THEN FLAG(140) % INTEGER > 1023.      02944000 T  0031!001313
                         END                                                                 02945000 T  0031!001712
               ELSE IF EXAMIN(NCR)="%" THEN                                                  02946000 T  0031!001713
                         BEGIN                                                               02947000 T  0031!001913
                         READACARD; COUNT:=ACCUM[1]:=0; GO DEBLANK;                          02948000 T  0031!002010
                         END                                                                 02949000 T  0031!002312
               ELSE      BEGIN                                                               02950000 T  0031!002312
                         RESULT:=5; SCANNER; % GET NEXT CHARACTER.                           02951000 T  0031!002313
                         Q:=ACCUM[1]; ELCLASS:=ACCUM[1].[18:6]                               02952000 T  0031!002411
                         END                                                                 02953000 T  0031!002610
                    END OF NEXTENT;                                                          02954000 T  0031!002712
                                                                                  31 IS   28 LONG. NEXT SEG    3

               BOOLEAN PROCEDURE BOOLPRIM;       FORWARD;                                    02955000 T  0003!034210
PRT(546) = BOOLPRIM
               PROCEDURE BOOLCOMP(B); BOOLEAN B; FORWARD;                                  02955500 T  0003!034210
PRT(547) = BOOLCOMP
               INTEGER PROCEDURE NEXT;                                                     02956000 T  0003!034210
PRT(550) = NEXT
                    BEGIN                                                                    02956500 T  0003!034210
                    LABEL EXIT;                                                              02957000 T  0003!034210
                                                                                  START OF SEGMENT **********   32
                    INTEGER T;                                                               02957500 T  0032!000010
STACK(F+3) = T
                    DEFINE ERROR = BEGIN  FLAG(603); GO EXIT  END#;                          02958000 T  0032!000010
                    SKAN;                                                                    02958500 T  0032!000010
                    IF RESULT=3 THEN ERROR; % NUMBERS NOT ALLOWED.                           02959000 T  0032!000313
                    IF RESULT=2 THEN % SPECIAL CHARACTER.                                    02959500 T  0032!000610
                         BEGIN                                                               02960000 T  0032!000712
                         T:=IF Q="1,0000" OR Q="1%0000" THEN 20 % FAKE OUT BOOLEXP.          02960500 T  0032!000713
                            ELSE ((T:=Q.[18:6]-2) & T[42:41:3]);                             02961000 T  0032!001010
                         IF T=11 OR T=19 OR T=20 THEN BATMAN:=SPECIAL[T] % (,),OR ;          02961500 T  0032!001313
                    ELSE FLAG(603);                                                          02962000 T  0032!001712
                         GO EXIT                                                             02962500 T  0032!002113
                         END SPECIAL CHARACTERS;                                             02963000 T  0032!002210
               COMMENT  LOOK FOR BOOLEAN OPERATORS, THEN OPTIONS;                            02963500 T  0032!002210
                         T:=     IF Q="3NOTOO" THEN NOTOP                                    02964000 T  0032!002210
```

```
                        ELSE IF Q="3ANDOO" THEN ANDOP                                   02964500 T  0032:0023:3
                        ELSE IF Q="2ORQOO" THEN OROP                                    02965000 T  0032:0025:3
                        ELSE IF Q="3EQVOO" THEN EQVOP                                   02965500 T  0032:0027:3
                        ELSE 0;                                                         02966000 T  0032:0029:3
                    IF T≠0 THEN BATMAN.CLASS:=T                                         02966500 T  0032:0031:2
                ELSE BATMAN:=1 & BOOID[2:7] & REAL(FINDOPTION(1))[1:1]; % OPTION.       02967000 T  0032:0032:1
            EXIT:                                                                       02967500 T  0032:0042:0
                    NEXT:=MYCLASS:=BATMAN.CLASS;                                        02968000 T  0032:0043:2
                END NEXT;                                                               02968500 T  0032:0044:1
                                                                                  32 IS   48 LONG, NEXT SEG    3



                    BOOLEAN PROCEDURE BOOLEXP;                                          02969000 T  0003:0342:0
                        BEGIN                                                           02969500 T  0003:0342:0
                        BOOLEAN B;                                                      02970000 T  0003:0342:0
                                                                                START OF SEGMENT **********   33
STACK(F+3) = B
                        B:=BOOLPRIM;                                                    02970500 T  0033:0000:0
                        WHILE MYCLASS≥EQVOP AND MYCLASS≤ANDOP DO BOOLCOMP(B);           02971000 T  0033:0001:2
                        BOOLEXP:=B                                                      02971500 T  0033:0004:1
                        END BOOLEXP;                                                    02972000 T  0033:0004:1
                                                                                  33 IS    8 LONG, NEXT SEG    3



                    BOOLEAN PROCEDURE BOOLPRIM;                                         02972500 T  0003:0342:0
                        BEGIN                                                           02973000 T  0003:0342:0
                        BOOLEAN B,KNOT;                                                 02973500 T  0003:0342:0
                                                                                START OF SEGMENT **********   34
STACK(F+3) = B
STACK(F+4) = KNOT
                        DEFINE SKIPIT = MYCLASS:=NEXT #;                                02974000 T  0034:0000:0
                        IF KNOT:=(NEXT=NOTOP) THEN SKIPIT;                              02974500 T  0034:0000:0
                        IF MYCLASS=LEFTPAREN THEN                                       02975000 T  0034:0003:2
                            BEGIN                                                       02975500 T  0034:0003:3
                            B:=BOOLEXP;                                                 02976000 T  0034:0004:0
                            IF MYCLASS≠RTPAREN THEN FLAG(604);                          02976500 T  0034:0005:2
                            END                                                         02977000 T  0034:0007:2
                    ELSE IF MYCLASS≠BOOID THEN FLAG(601)                                02977500 T  0034:0007:2
                    ELSE B:=BATMAN<0;                                                   02978000 T  0034:0009:2
                        IF KNOT THEN B:=NOT B; SKIPIT;                                  02978500 T  0034:0011:3
                        BOOLPRIM:=B                                                     02979000 T  0034:0014:0
                        END BOOLPRIM;                                                   02979500 T  0034:0014:0
                                                                                  34 IS   18 LONG, NEXT SEG    3



                    PROCEDURE BOOLCOMP(B); BOOLEAN B;                                   02980000 T  0003:0342:0
                        BEGIN                                                           02980500 T  0003:0342:0
                        REAL OPCLASS;                                                   02981000 T  0003:0342:0
                                                                                START OF SEGMENT **********   35
STACK(F+2) = OPCLASS
                        BOOLEAN T;                                                      02981500 T  0035:0000:0
STACK(F+3) = T
```

```
                        OPCLASS:=MYCLASS;                                                                    02982000 T  00351000010
                        T:=BOOLPRIM;                                                                         02982500 T  00351000011
                        WHILE OPCLASS<MYCLASS DO BOOLCOMP(T);                                                02983000 T  00351000113
                        B:=     IF OPCLASS=ANDOP THEN (B AND T)                                              02983500 T  00351000411
                           ELSE IF OPCLASS=OROP  THEN (B OR T)                                               02984000 T  00351000610
                           ELSE                       (B EQV T);                                             02984500 T  00351000811
                        END BOOLCOMP;                                                                        02985000 T  00351001112
                                                                                                  35 IS   14 LONG. NEXT SEG    3

                        %
                        COMMENT##############################################################################  02985500 T  00031034210
                                      FORWARD DECLARATIONS                                                   02986000 T  00031034210
                        ##############################################################################;      02986500 T  00031034210
                        %                                                                                    02987000 T  00031034210
                         PROCEDURE AEXP; FORWARD;                                                            02987500 T  00031034210
PRT(551) = AEXP                                                                                              03001000 T  00031034210
                         PROCEDURE ARITHSEC; FORWARD;                                                        03002000 T  00031034210
PRT(552) = ARITHSEC
                         PROCEDURE SIMPARITH; FORWARD;                                                       03003000 T  00031034210
PRT(553) = SIMPARITH
                         PROCEDURE ARITHCOMP; FORWARD;                                                       03004000 T  00031034210
PRT(554) = ARITHCOMP
                         PROCEDURE PRIMARY; FORWARD;                                                         03005000 T  00031034210
PRT(555) = PRIMARY
                         PROCEDURE BEXP; FORWARD;                                                            03006000 T  00031034210
PRT(556) = BEXP
                        INTEGER PROCEDURE EXPRSS; FORWARD;                                                   03007000 T  00031034210
PRT(557) = EXPRSS
                        INTEGER PROCEDURE BOOSEC; FORWARD;                                                   03008000 T  00031034210
PRT(560) = BOOSEC
                        PROCEDURE SIMPBOO; FORWARD;                                                          03009000 T  00031034210
PRT(561) = SIMPBOO
                        PROCEDURE BOOCOMP; FORWARD;                                                          03010000 T  00031034210
PRT(562) = BOOCOMP
                        INTEGER PROCEDURE BOOPRIM; FORWARD;                                                  03011000 T  00031034210
PRT(563) = BOOPRIM
                        PROCEDURE RELATION; FORWARD;                                                         03012000 T  00031034210
PRT(564) = RELATION
                        INTEGER PROCEDURE IFEXP; FORWARD;                                                    03013000 T  00031034210
PRT(565) = IFEXP
                        PROCEDURE PARSE; FORWARD;                                                            03014000 T  00031034210
PRT(566) = PARSE
                        PROCEDURE DOTIT; FORWARD;                                                            03015000 T  00031034210
PRT(567) = DOTIT
                        PROCEDURE GENGO(ELBATWORD); VALUE ELBATWORD; REAL ELBATWORD; FORWARD;                03016000 T  00031034210
PRT(570) = GENGO
                        PROCEDURE DEXP; FORWARD;                                                             03017000 T  00031034210
PRT(571) = DEXP
                        PROCEDURE  IFCLAUSE; FORWARD;                                                        03018000 T  00031034210
PRT(572) = IFCLAUSE
                       INTEGER PROCEDURE GET(SYLLABLE);VALUE SYLLABLE; REAL SYLLABLE; FORWARD;               03019000 T  00031034210
PRT(573) = GET
                       INTEGER PROCEDURE GNAT(L); VALUE L; REAL L; FORWARD;                                  03020000 T  00031034210
PRT(574) = GNAT
```

```
                    PROCEDURE PANA; FORWARD;                                                                    03021000 T  0003!034210
PRT(575) = PANA
                    PROCEDURE IFSTMT; FORWARD;                                                                  03022000 T  0003!034210
PRT(576) = IFSTMT
                    PROCEDURE GOGEN(LABELBAT,BRANCHTYPE);                                                       03023000 T  0003!034210
PRT(577) = GOGEN
                              VALUE LABELBAT,BRANCHTYPE;                                                        03024000 T  0003!034210
                              REAL LABELBAT,BRANCHTYPE; FORWARD;                                                03025000 T  0003!034210
                    BOOLEAN PROCEDURE SIMPGO; FORWARD;                                                          03026000 T  0003!034210
PRT(600) = SIMPGO
                    PROCEDURE STMT; FORWARD;                                                                    03027000 T  0003!034210
PRT(601) = STMT
                    PROCEDURE EMIT(SYLLABLE); VALUE SYLLABLE; REAL SYLLABLE; FORWARD;                           03028000 T  0003!034210
PRT(602) = EMIT
                    PROCEDURE PROCSTMT(FROM); VALUE FROM; BOOLEAN FROM; FORWARD;                                03029000 T  0003!034210
PRT(603) = PROCSTMT
                    PROCEDURE STRMPROCSTMT; FORWARD;                                                            03030000 T  0003!034210
PRT(604) = STRMPROCSTMT
                    BOOLEAN PROCEDURE GETINT; FORWARD;                                                          03031000 T  0003!034210
PRT(605) = GETINT
                    INTEGER PROCEDURE DIVIDE(NUMBER,P1,P2); VALUE NUMBER;                                       03032000 T  0003!034210
PRT(606) = DIVIDE
                              INTEGER P1,P2,NUMBER; FORWARD;                                                    03033000 T  0003!034210
                    PROCEDURE CONSTANTCLEAN; FORWARD;                                                           03034000 T  0003!034210
PRT(607) = CONSTANTCLEAN
                    PROCEDURE SCATTERELBAT; FORWARD;                                                            03035000 T  0003!034210
PRT(610) = SCATTERELBAT
                    PROCEDURE EMITB(BRANCH,FROM,TOWARDS); VALUE BRANCH,FROM,TOWARDS;                            03036000 T  0003!034210
PRT(611) = EMITB
                         INTEGER BRANCH,FROM,TOWARDS; FORWARD;                                                  03037000 T  0003!034210
                    PROCEDURE VARIABLE(FROM); REAL FROM; FORWARD;                                               03038000 T  0003!034210
PRT(612) = VARIABLE
                    PROCEDURE IMPFUN; FORWARD;                                                                  03039000 T  0003!034210
PRT(613) = IMPFUN
                    PROCEDURE STREAMSTMT; FORWARD;                                                              03040000 T  0003!034210
PRT(614) = STREAMSTMT
                     PROCEDURE SEGMENTSTART;FORWARD;                                                            03041000 T  0003!034210
PRT(615) = SEGMENTSTART
                     PROCEDURE SEGMENT(SIZE,NO,NOO);                                                            03042000 T  0003!034210
PRT(616) = SEGMENT
                         VALUE SIZE,NO,NOO;                                                                     03043000 T  0003!034210
                         REAL SIZE,NO,NOO;                                                                      03044000 T  0003!034210
                         FORWARD;                                                                               03045000 T  0003!034210
                    INTEGER PROCEDURE BAE; FORWARD;                                                             03046000 T  0003!034210
PRT(617) = BAE
                   REAL PROCEDURE PROGDFSCBLDR(A,B,C);VALUE A,B,C; REAL A,B,C; FORWARD;                         03047000 T  0003!034210
PRT(620) = PROGDFSCBLDR
                    PROCEDURE BANA; FORWARD;                                                                    03048000 T  0003!034210
PRT(621) = BANA
                    PROCEDURE EMITNUM(A); VALUE A; REAL A; FORWARD;                                             03049000 T  0003!034210
PRT(622) = EMITNUM
                    PROCEDURE EMITD(A,B,T); VALUE A,B,T; INTEGER A,B,T; FORWARD;                                03050000 T  0003!034210
PRT(623) = EMITD
                   INTEGER PROCEDURE GETSPACE(S,L); VALUE S,L;                                                  03051000 T  0003!034210
PRT(624) = GETSPACE
                                                    INTEGER L; BOOLEAN S; FORWARD;                              03051001 T  0003!034210
                    PROCEDURE FORSTMT; FORWARD;                                                                 03052000 T  0003!034210
PRT(625) = FORSTMT
```

```
                PROCEDURE F;  FORWARD;                                                                                  03053000 T  00031034210
                                                                                                                        03054000 T  00031034210
PRT(626) = E
                PROCEDURE ENTRY(TYPE); VALUE TYPE;REAL TYPE; FORWARD;                                                   03055000 T  00031034210
PRT(627) = ENTRY
                PROCEDURE FORMATGEN;FORWARD;                                                                            03056000 T  00031034210
PRT(630) = FORMATGEN
                PROCEDURE EXPLICITFORMAT;  FORWARD;                                                                     03056100 T  00031034210
PRT(631) = EXPLICITFORMAT
                BOOLEAN PROCEDURE FORMATPHRASE;  FORWARD;                                                               03056200 T  00031034210
PRT(632) = FORMATPHRASE
                PROCEDURE PUTNBUMP(P1);  VALUE P1;  REAL P1;  FORWARD;                                                  03057000 T  00031034210
PRT(633) = PUTNBUMP
                PROCEDURE JUMPCHKNX;  FORWARD;                                                                          03058000 T  00031034210
PRT(634) = JUMPCHKNX
                PROCEDURE JUMPCHKX;  FORWARD;                                                                           03059000 T  00031034210
PRT(635) = JUMPCHKX
                 PROCEDURE DBLSTMT; FORWARD;                                                                            03060000 T  00031034210
PRT(636) = DBLSTMT
                 PROCEDURE READSTMT;  FORWARD;                                                                          03061000 T  00031034210
PRT(637) = READSTMT
                INTEGER PROCEDURE FILEATTRIBUTEHANDLER(N); VALUE N; REAL N; FORWARD ;                                   03061010 T  00031034210
PRT(640) = FILEATTRIBUTEHANDLER
                 PROCEDURE WRITESTMT; FORWARD;                                                                          03062000 T  00031034210
PRT(641) = WRITESTMT
                 PROCEDURE SPACESTMT; FORWARD;                                                                          03063000 T  00031034210
PRT(642) = SPACESTMT
                 PROCEDURE CLOSESTMT; FORWARD;                                                                          03064000 T  00031034210
PRT(643) = CLOSESTMT
                  PROCEDURE LOCKSTMT;  FORWARD;                                                                         03065000 T  00031034210
PRT(644) = LOCKSTMT
                  PROCEDURE RWNDSTMT;  FORWARD;                                                                         03066000 T  00031034210
PRT(645) = RWNDSTMT
                  PROCEDURE BLOCK(S); VALUE S; BOOLEAN S; FORWARD;                                                      03067000 T  00031034210
PRT(646) = BLOCK
                PROCEDURE PURGE(STOPPER);VALUE STOPPER;REAL STOPPER;FORWARD;                                            03068000 T  00031034210
PRT(647) = PURGE
                PROCEDURE ENTER(TYPEV);                                                                                 03069000 T  00031034210
PRT(650) = ENTER
                          VALUE TYPEV;                                                                                  03070000 T  00031034210
                                INTEGER TYPEV; FORWARD;                                                                 03071000 T  00031034210
                INTEGER PROCEDURE PASSTYPE(P); VALUE P; REAL P; FORWARD;                                                03074000 T  00031034210
PRT(651) = PASSTYPE
                PROCEDURE PASSALPHA(P); VALUE P; REAL P; FORWARD;                                                       03075000 T  00031034210
PRT(652) = PASSALPHA
                 PROCEDURE LISTELEMENT; FORWARD;                                                                        03076000 T  00031034210
PRT(653) = LISTELEMENT
                 REAL PROCEDURE LISTGEN;  FORWARD;                                                                      03077000 T  00031034210
PRT(654) = LISTGEN
                 PROCEDURE UNKNOWNSTMT; FORWARD;                                                                        03078000 T  00031034210
PRT(655) = UNKNOWNSTMT
                 PROCEDURE FAULTSTMT; FORWARD;                                                                          03079000 T  00031034210
PRT(656) = FAULTSTMT
                 PROCEDURE FAULTDEC; FORWARD;                                                                           03080000 T  00031034210
PRT(657) = FAULTDEC
                 PROCEDURE SORTSTMT;     FORWARD;                                                                       03081000 T  00031034210
PRT(660) = SORTSTMT
```

```
                PROCEDURE MERGESTMT;   FORWARD;                                          03082000 T  0003:034210
PRT(661) = MERGESTMT
                PROCEDURE CASESTMT; FORWARD;                                             03083000 T  0003:034210
PRT(662) = CASESTMT
              PROCEDURE HANDLETHETAILENDOFAREADORSPACESTATEMENT;  FORWARD;                03084000 T  0003:034210
PRT(663) = HANDLETHETAILENDOFAREADORSPACESTATEMENT
                ALPHA PROCEDURE BUGGER(S); VALUE S; INTEGER S; FORWARD;                  03100000 T  0003:034210
PRT(664) = BUGGER
                COMMENT THIS SECTION CONTAINS THE EMITTERS.  THEY ARE THE AGENTS WHICH   04000000 T  0003:034210
                          ACTUALLY PRODUCE CODE AND DEBUGING OUTPUT;                     04001000 T  0003:034210
                COMMENT EMITL EMITS A LIT CALL;                                          04002000 T  0003:034210
                PROCEDURE EMITL(LITERAL); VALUE LITERAL; INTEGER LITERAL;                04003000 T  0003:034210
PRT(665) = EMITL
                          EMIT(0&LITERAL[36:38:10]);                                     04004000 T  0003:034210


                COMMENT EMITO EMIT AN OPERATOR;                                          04005000 T  0003:034410
                PROCEDURE EMITO(OPERATOR); VALUE OPERATOR; INTEGER OPERATOR;             04006000 T  0003:034410
PRT(666) = EMITO
                          EMIT(1&OPERATOR[36:38:10]);                                    04007000 T  0003:034410


                COMMENT EMITC IS PRIMARILY FOR USE BY STRMSTMT TO EMIT CHARACTOR MODE    04008000 T  0003:034610
                          OPERATORS.  HOWEVER IT ALSO HANDLES DIA, DIB, AND TRB;         04009000 T  0003:034610
              PROCEDURE EMITC(REPEAT,OPERATOR); VALUE REPEAT,OPERATOR;                   04010000 T  0003:034610
PRT(667) = EMITC
                INTEGER REPEAT,OPERATOR;                                                 04011000 T  0003:034610
                BEGIN                                                                    04012000 T  0003:034610
                  IF REPEAT≥64 THEN FLAG(268);                                           04013000 T  0003:034610
                  EMIT(OPERATOR&REPEAT[36:42:6]) END EMITC;                              04014000 T  0003:034810


              COMMENT EMITV EMITS AN OPERAND CALL.  IF THE ADDRESS IS FOR THE SECOND     04015000 T  0003:035010
                          HALF OF THE PRT, THEN IT ALSO EMITS A PRTE;                    04016000 T  0003:035010
              PROCEDURE EMITV(ADDRESS); VALUE ADDRESS; INTEGER ADDRESS;                  04017000 T  0003:035010
PRT(670) = EMITV
                          BEGIN IF ADDRESS > 1023 THEN EMITO(PRTE);                      04018000 T  0003:035010
                              EMIT(2 & ADDRESS [36:38:10]) END EMITV;                    04019000 T  0003:035210


              COMMENT EMITN EMITS A DESCRIPTOR CALL.  IF THE ADDRESS IS FOR THE          04020000 T  0003:035410
                          SECOND HALF OF THE PRT, THEN IT ALSO EMITS A PRTE;             04021000 T  0003:035410
              PROCEDURE EMITN(ADDRESS); VALUE ADDRESS; INTEGER ADDRESS;                  04022000 T  0003:035410
PRT(671) = EMITN
                          BEGIN IF ADDRESS > 1023 THEN EMITO(PRTE);                      04023000 T  0003:035410
                              EMIT(3 & ADDRESS [36:38:10]) END EMITN;                    04024000 T  0003:035610


              COMMENT EMITPAIR EMITS A LITC ADDRESS FOLLOWED BY OPERATOR.  IF THE        04025000 T  0003:035810
```

```
                        ADDRESS IS FOR THE SECOND HALF OF THE PRT, THEN IT ALSO          04026000 T  00031035810
                        EMITS PRTE;                                                      04027000 T  00031035810
                PROCEDURE EMITPAIR(ADDRESS,OPERATOR);                                    04028000 T  00031035810
PRT(672) = EMITPAIR
                        VALUE ADDRESS,OPERATOR;                                          04029000 T  00031035810
                        INTEGER ADDRESS,OPERATOR;                                        04030000 T  00031035810
                        BEGIN                                                            04031000 T  00031035810
                             EMITL(ADDRESS);                                             04032000 T  00031035810
                             IF ADDRESS > 1023 THEN EMITO(PRTE);                         04033000 T  00031035811
                             EMITO(OPERATOR) END EMITPAIR;                               04034000 T  00031036011


                COMMENT  EMITUP IS RESPONSIBLE FOR COMPILING THE CODE TO RAISE AN        04035000 T  00031036113
                        EXPRESSION TO SOME POWER  IF THE EXPONENT IS A LITERAL           04036000 T  00031036113
                        OR A NEGATIVE LITERAL THEN IN LINE CODE IS COMPILED.  THIS       04037000 T  00031036113
                             CODE CONSISTS OF A SERIES OF DUPS AND MULS.  AS WITH        04038000 T  00031036113
                        EMITLNG CARE MUST BE TAKEN TO AVOID CONFUSION WITH LINKS         04039000 T  00031036113
                        AND CONDITIONAL EXPRESSIONS.  IF THESE SPECIAL CASES DO          04040000 T  00031036113
                        NOT HOLD, THEN A CALL ON AN INTRINSIC PROCEDURE, XTOTHEI,        04041000 T  00031036113
                        IS CONSTRUCTED.  XTOTHEI PRODUCES A SERIES OF MULTIPLIES         04042000 T  00031036113
                        (APROXIMATELY LN I MULTIPLIES) IF I IS AN INTEGER.               04043000 T  00031036113
                        OTHERWISE IT CALLS LN AND EXP;                                   04044000 T  00031036113
                PROCEDURE EMITUP;                                                        04045000 T  00031036113
PRT(673) = EMITUP
                        BEGIN INTEGER BACKUP, CTR;                                       04046000 T  00031036113
                                                                                START OF SEGMENT **********   36
STACK(F+2) = BACKUP
STACK(F+3) = CTR
                            LABEL E;                                                     04047000 T  00361000010
                        IF NOT LINKTOG THEN GO TO E;                                     04048000 T  00361000010
COMMENT CALL XTOTHFI IF LAST THING IS LINK;                                              04049000 T  00361000011
                        IF GET(L-1) = 537 THEN                                           04050000 T  00361000011
COMMENT LAST OPERATOR IS CHS;                                                            04051000 T  00361000211
                            BEGIN BACKUP ← 1; L ← L-1 END;                               04052000 T  00361000211
                        IF(GT4 ← GET(L-1)).[46:2] = 0                                    04053000 T  00361000512
                          THEN BEGIN                                                     04054000 T  00361000712
COMMENT IT IS A LITERAL;                                                                 04055000 T  00361000810
                            BACKUP ← BACKUP+1; L ← L-1;                                  04056000 T  00361000810
                        IF GET(L-1).[39:9] = 153 THEN GO TO E;                           04057000 T  00361001011
COMMENT  CALL XTOTHE IF THE LAST OPERATOR IS A BRANCH;                                   04058000 T  00361001313
                        CTR ← 1;  GT4 ← GT4 DIV 4;                                       04059000 T  00361001313
                        WHILE GT4 DIV 2 ≠ 0                                              04060000 T  00361001513
                         DO BEGIN                                                        04061000 T  00361001611
                            EMITO(DUP);                                                  04062000 T  00361001713
                            IF BOOLEAN(GT4) THEN BEGIN CTR←CTR+1;EMITO(DUP)END;          04063000 T  00361001811
                            EMITO(MUL);                                                  04064000 T  00361002112
                            GT4 ← GT4 DIV 2 END;                                         04065000 T  00361002210
                        IF GT4 =0 THEN BEGIN EMITO(DEL);EMITL(1) END                     04066000 T  00361002313
                         ELSE WHILE CTR ← CTR-1 ≠ 0 DO EMITO(MUL);                       04067000 T  00361002611
                        IF BACKUP = 2                                                    04068000 T  00361003011
                         THEN BEGIN                                                      04069000 T  00361003011
                            EMITL(1);                                                    04070000 T  00361003113
                            EMITO(XCH);                                                  04071000 T  00361003211
                            EMITO(128) END END                                           04072000 T  00361003312
                        ELSE BEGIN                                                       04073000 T  00361003410
```

```
F:              L ← L+BACKUP;                                                    04074000 T  00361003411
                EMIT0(MKS);                                                      04075000 T  00361003610
                EMITPAIR(GNAT(LOGI),LOD);                                        04076000 T  00361003712
                EMITPAIR(GNAT(EXPI),LOD);                                        04077000 T  00361003811
                EMITV(GNAT(XTOTHEI));                                            04078000 T  00361004010
                STACKCT ← 0;                                          %A         04078500 T  00361004112
                EMIT0(DEL) END END EMITUP;                                       04079000 T  00361004210
                                                                      36 IS   45 LONG, NEXT SEG    3


        COMMENT ADJUST ADJUST L TO THE BEGINING OF A WORD AND FILLS IN THE       04080000 T  00031036113
                INERVENING SPACE WITH NOPS. IT CHECKS STREAMTOG TO DECIDE        04081000 T  00031036113
                WHICH SORT OF NOP TO USE;                                        04082000 T  00031036113
        PROCEDURE ADJUST;                                                        04083000 T  00031036113
PRT(674) = ADJUST
                BEGIN                                                            04084000 T  00031036113
                  DIALA ← DIALB ← 0;                                             04085000 T  00031036113
                  WHILE L.[46:2] ≠ 0 DO EMIT(IF STREAMTOG THEN 1 ELSE 45)        04086000 T  00031036312
                END ADJUST;                                                      04087000 T  00031036713


        COMMENT EMITLNG CHANGES A RELATIONAL FOLLOWED BY A NEGATE TO THE         04088000 T  00031036811
                NEGATED RELATIONAL.  IT ALSO CHANGES A NEGATE FOLLOWED           04089000 T  00031036811
                BY A NEGATE TO NOTHING.  CARE MUST BE EXERCIZED.  A LINK         04090000 T  00031036811
                (FOR CONSTANT TO BE EMITTED LATER) MIGHT LOOK LIKE AN LNG        04091000 T  00031036811
                OR A RELATIONAL OPERATOR.  THIS IS THE USE OF LINKTOG.           04092000 T  00031036811
                ALSO A CONSTRUCT AS  NOT ( IF B THEN X=Y ELSE Y=Z)               04093000 T  00031036811
                COULD GIVE TROUBLE.  THIS IS THE MEANING OF THE OBSCURE          04094000 T  00031036811
                EMITS FOLLOWED BY L ← L-1 FOUND IN IFEXP, BOOSEC, BOOCOMP,       04095000 T  00031036811
                AND RELATION - THAT CODE SERVES TO SET A FLAG FOR USE BY         04096000 T  00031036811
                EMITLNG;                                                         04097000 T  00031036811
        PROCEDURE EMITLNG;                                                       04098000 T  00031036811
PRT(675) = EMITLNG
                BEGIN LABEL E;                                                   04099000 T  00031036811
                                                                      START OF SEGMENT **********   37
                  IF NOT LINKTOG THEN GO TO E;                                   04100000 T  00371000010
        COMMENT   GO TO E IF LAST THING IS A LINK;                               04101000 T  00371000011
                  IF GET(L) ≠ 0 THEN GO TO E;                                    04102000 T  00371000011
        COMMENT   EITHER LAST EXPRESSION WAS CONDITIONAL OR THERE IS NO          04103000 T  00371000211
                  LNG OR RELATIONAL OPERATOR;                                    04104000 T  00371000211
                  IF GT1 ← GET(L-1) = 77 THEN L ← L-1                            04105000 T  00371000211
        COMMENT LAST THING WAS AN LNG - SO CANCEL IT;                            04106000 T  00371000513
             ELSE IF GT1.[42:6]=21 AND GT1.[37:2]=0 THEN % AHA                   04107000 T  00371000513
        COMMENT LAST THING WAS A RELATIONAL;                                     04108000 T  00371000913
                    BEGIN L←L-1; EMIT0(REAL(BOOLEAN(GT1.[36:10]) EQV             04109000 T  00371000913
                      BOOLEAN(IF GT1.[40:2] = 0 THEN 511 ELSE 463)))             04110000 T  00371001211
        COMMENT NEGATE THE RELATIONAL; END ELSE                                  04111000 T  00371001513
              E:  EMIT0(LNG) END EMITLNG;                                        04112000 T  00371001513
                                                                      37 IS   18 LONG, NEXT SEG    3


        COMMENT   EMITB EMITS A BRANCH OPERATOR AND ITS ASSOCIATED NUMBER;       04113000 T  00031036811
```

```
                    PROCEDURE EMITB(BRANCH,FROM,TOWARDS);                                   04114000 T  0003!0368!1
                              VALUE BRANCH,FROM,TOWARDS;                                    04115000 T  0003!0368!1
                              INTEGER BRANCH,FROM,TOWARDS;                                  04116000 T  0003!0368!1
                              BEGIN                                                         04117000 T  0003!0368!1
                                INTEGER TL;                                                 04118000 T  0003!0368!1
                                                                                  START OF SEGMENT **********   38
STACK(F+2) = TL
                                TL ← L;                                                     04119000 T  0038!0000!0
                                L ← FROM-2;                                                 04120000 T  0038!0000!1
                                GT1 ← TOWARDS-FROM;                                         04120100 T  0038!0002!0
                                IF TOWARDS.[46:2] = 0                                       04120200 T  0038!0003!2
                                  THEN BEGIN                                                04120300 T  0038!0004!0
                                    BRANCH ← BRANCH&1[39:47:1];                             04120400 T  0038!0005!2
                                    GT1 ← TOWARDS DIV 4 - (FROM-1) DIV 4 END;               04120500 T  0038!0006!1
                                EMITNUM(ABS(GT1));                                          04121000 T  0038!0009!3
                                EMITO(BRANCH&(REAL(GT1≥ 0)+1)[42:46:2]);                    04122000 T  0038!0010!1
                                IF BOOLEAN(BRANCH.[38:1]) THEN DIALA ← DIALB ← 0;           04123000 T  0038!0013!2
                                L ← TL                                                      04124000 T  0038!0015!3
                              END EMITB;                                                    04125000 T  0038!0015!3
                                                                                    38 IS    19 LONG, NEXT SEG    3


                        COMMENT   DEBUGWORD FORMATS TWO FIELDS FOR DEBUGGING OUTPUT IN      04126000 T  0003!0368!1
                                  OCTAL, NAMELY :                                           04127000 T  0003!0368!1
                                  1. 4 CHARACTERS FOR THE L REGISTER,                       04128000 T  0003!0368!1
                                  2.16 CHARACTERS FOR THE WORD BEING EMITTED,    ;          04129000 T  0003!0368!1
                    STREAM  PROCEDURE DEBUGWORD( SEQ,CODE,FEIL); VALUE SEQ,CODE ;           04130000 T  0003!0368!1
PRT(676) = DEBUGWORD
                                  BEGIN                                                     04131000 T  0003!0368!1
                                    DI←FEIL; SI← LOC SEQ; SI← SI+4; DS ← 4 CHR;             04132000 T  0003!0369!2
                                    DS ← 2 LIT"  ";                                         04133000 T  0003!0370!0
                                    SI ← LOC CODE ;                                         04134000 T  0003!0370!1
                                    16(  DS ← 3 RESET; 3( IF SB THEN DS←SET ELSE            04135000 T  0003!0370!1
                                         DS ← RESET ; SKIP 1 SB));                          04136000 T  0003!0372!1
                                 49(DS ← 2 LIT "  ");                                       04137000 T  0003!0373!3
                                  END ;                                                     04138000 T  0003!0374!1


                          COMMENT   EMITWORD PLACES THE PARAMETER,"WORD",INTO EDOC, IF      04139000 T  0003!0374!1
                                    DEBUGGING IS REQUIRED, "L" AND "WORD" ARE OUTPUT ON     04140000 T  0003!0374!1
                                    THE  PRINTER FILE IN OCTAL FORMAT.     ;                04141000 T  0003!0374!1
                      PROCEDURE  EMITWORD (WORD); VALUE WORD; REAL WORD;                    04142000 T  0003!0374!1
PRT(677) = EMITWORD
                                  BEGIN                                                     04143000 T  0003!0374!1
                                  ADJUST;                                                   04144000 T  0003!0374!1
                                  IF L≥4092 THEN ERR(200)                                   04145000 T  0003!0375!3
                                       ELSE BEGIN                                           04146000 T  0003!0377!2
                                       MOVE(1,WORD,EDOC[L.[36:3],L.[39:7]]);                04147000 T  0003!0379!2
                                       IF DEBUGTOG THEN                                     04148000 T  0003!0382!1
                                       BEGIN DEBUGWORD(B2D(L),WORD,LIN);                    04149000 T  0003!0383!2
                                            WRITELINE  END;                                 04150000 T  0003!0385!3
                                       L←L+4    END                                        04151000 T  0003!0396!0
                                  END  EMITWORD;                                            04152000 T  0003!0397!2
```

```
                    COMMENT CONSTANTCLEAN IS CALLED AFTER AN UNCONDITIONAL BRANCH HAS          04153000 T  0003!039713
                            BEEN EMITTED. IF ANY CONSTANTS HAVE BEEN ACCUMULATED BY           04154000 T  0003!039713
                            EMITNUM IN INFO[0,*], CONSTANTCLEAN WILL FIX  THE CHAIN           04155000 T  0003!039713
                            OF  C-RELATIVE  OPDC S LEFT BY EMITNUM, IF  C-RELATIVE            04156000 T  0003!039713
                            ADDRESSING IS IMPOSSIBLE (I.E. THE ADDRESS                        04157000 T  0003!039713
                            IF GREATER THAN 127 WORDS) THEN THE CONSTANT ALONG WITH           04158000 T  0003!039713
                            THE  1ST  LINK OF THE OPDC CHAIN IS ENTERED IN INFO.              04159000 T  0003!039713
                            AT  PURGE  TIME THE REMAINING OPDC S ARE EMITTED WITH             04160000 T  0003!039713
                            F -RELATIVE ADDRESSING AND CODE EMITTED TO STORE THE              04161000 T  0003!039713
                            CONSTANTS INTO THE PROPER F-RELATIVE CELLS.  ;                    04162000 T  0003!039713
                  PROCEDURE CONSTANTCLEAN ;                                                   04163000 T  0003!039713
                            IF MRCLEAN THEN                                                   04164000 T  0003!039713
                              BEGIN                                                           04165000 T  0003!039810
                                INTEGER J,TEMPL,D,LINK;                                       04166000 T  0003!039811
PRT(700) = *SEGMENT DESCRIPTOR*

                                                                                  START OF SEGMENT **********   39
STACK(F+2) = J
STACK(F+3) = TEMPL
STACK(F+4) = D
STACK(F+5) = LINK
                                BOOLEAN CREL;                                             04167000 T  0039!000010
STACK(F+6) = CREL
                                LABEL  ALLTHU ;                                           04168000 T  0039!000010
                                DIALA ← DIALB ← 0;                                        04169000 T  0039!000010
                                FOR J ← 1 STEP 2 UNTIL LASTENTRY DO                       04170000 T  0039!000112
                                 BEGIN                                                    04171000 T  0039!000210
                                ADJUST; TEMPL←L; L←INFO[0,255-J+1];                       04172000 T  0039!000210
                                   CREL ← FALSE;                                          04173000 T  0039!000610
                                   DO BEGIN                                               04174000 T  0039!000611
                                        IF D←(TEMPL-L+3)DIV 4≥128 THEN                    04175000 T  0039!000712
                                          BEGIN                                           04176000 T  0039!000913
                          NCII←NCII+1;                                                    04177000 T  0039!001010
                  PUTNBUMP(L&NONLITNO[2:41:7]&(NEXTINFO-LASTINFO)[27:40:8]);              04178000 T  0039!001113
                  PUTNBUMP(TAKE(255-J)); LASTINFO←NEXTINFO-2;                             04179000 T  0039!001411
                                          GO TO ALLTHU;                                   04180000 T  0039!001713
                                        END;                                              04181000 T  0039!001810
                                       LINK←GET(L);                                       04182000 T  0039!001810
                                       CREL ← TRUE;                                       04183000 T  0039!001913
                                       EMITV(D + 768);                                    04184000 T  0039!002010
                                     END  UNTIL L←   LINK   = 4095  ;                     04185000 T  0039!002113
                        ALLTHU:      L ← TEMPL;                                           04186000 T  0039!002312
                                   IF  CREL  THEN  EMITWORD( INFO[0,255-J ]);             04187000 T  0039!002411
                                 END;                                                     04188000 T  0039!002713
                               LASTENTRY ← 0;                                             04189000 T  0039!003010
                             END ;                                                        04190000 T  0039!003011
PRT(701) = *SEGMENT DESCRIPTOR*

                                                                        39 IS   33 LONG, NEXT SEG    3


                    COMMENT   EMITNUM HANDLES THE EMISSION OF CODE FOR CONSTANTS,BOTH         04191000 T  0003!040312
                              EXPLICIT AND IMPLICIT, IN EVERY CASE,EMITNUM WILL               04192000 T  0003!040312
```

```
                        PRODUCE CODE TO GET THE DESIRED CONSTANT ON TOP OF      04193000 T  0003!040!2
                        THE STACK. IF THE NUMBER IS A LITERAL A SIMPLE LITC     04194000 T  0003!040!2
                        SYLLABLE IS PRODUCED, HOWEVER,NON-LITERALS ARE KEPT     04195000 T  0003!040!2
                        IN THE ZERO-TH ROW OF INFO WITH THE SYLLABLE            04196000 T  0003!040!2
                        POSITION,L, THE FIRST EMITNUM ON A PARTICULAR           04197000 T  0003!040!2
                        CONSTANT CAUSES THE VALUES OF L AND THE CONSTANT        04198000 T  0003!040!2
                        TO BE STORED IN INFO[0,*] (NOTE:ITEMS ARE STORED        04199000 T  0003!040!2
                        IN REVERSE STARTING WITH INFO[0,255],ETC.), THEN        04200000 T  0003!040!2
                        ITS THE JOB OF CONSTANTCLEAN TO EMIT THE ACTUAL         04201000 T  0003!040!2
                        OPDC (SEE CONSTANTCLEAN PROCEDURE FOR DETAILS) ;        04202000 T  0003!040!2
            PROCEDURE EMITNUM( C ); VALUE C; REAL C;                            04203000 T  0003!040!2
                        BEGIN LABEL FINISHED,FOUND ; REAL N;                    04204000 T  0003!040!2
                                                                          START OF SEGMENT **********   40
STACK(F+2) = N          \
                        IF C,[1:37]=0 THEN EMITL(C)                             04205000 T  0040!000!0
                        ELSE                                                    04206000 T  0040!000!1
                             BEGIN                                              04207000 T  0040!000!2
                             FOR  N ← 1 STEP 2 UNTIL LASTENTRY DO               04208000 T  0040!000!3
                          IF INFO[0,255-N] = C THEN  GO TO FOUND ;              04209000 T  0040!000!2
                             INFO[0,255 -LASTENTRY] ← L;                        04210000 T  0040!001!0
                             INFO[0,255 -LASTENTRY-1]← C ;                      04211000 T  0040!001!2
                             EMITN(1023);                                       04212000 T  0040!001!3
                             LINKTOG←FALSE;                                     04213000 T  0040!001!0
                             IF LASTENTRY ← LASTENTRY+2 ≥ 128  THEN             04214000 T  0040!001!2
                                  BEGIN                                         04215000 T  0040!001!1
                                  C ← BUMPL;                                    04216000 T  0040!001!2
                                  CONSTANTCLEAN;                                04217000 T  0040!002!2
                                  EMITB(BFW,C,L);                               04218000 T  0040!002!3
                                  END;                                          04219000 T  0040!002!1
                             GO TO FINISHED;                                    04220000 T  0040!002!1
                   FOUND:    EMIT(INFO[0,255 -N+1]);                           04221000 T  0040!002!2
                             LINKTOG←FALSE;                                     04222000 T  0040!002!1
                             INFO[0,255-N+1]← L-1;                              04223000 T  0040!002!3
                             END;                                               04224000 T  0040!003!2
                  FINISHED:END  EMITNUM  ;                                      04225000 T  0040!003!2
                                                                        40 IS   34 LONG, NEXT SEG    3
```

```
                   COMMENT   SEARCH PERFORMS A BINARY SEARCH ON THE COP AND MOP      04226000 T  0003!040!2
                             ARRAYS, GIVEN THE OPERATOR BITS SEARCH  YIELDS THE BCD  04227000 T  0003!040!2
                             MNEUMONIC FOR THAT OPERATOR, IF THE OPERATOR CANNOT    04228000 T  0003!040!2
                             BE FOUND SEARCH YIELDS BLANKS.                         04229000 T  0003!040!2
                             NOTE: DIA,DIB,TRB ARE RETURNED AS BLANKS.      ;       04230000 T  0003!040!2
                ALPHA PROCEDURE  SEARCH (Q,KEY); VALUE KEY; ARRAY Q[0]; REAL KEY ;  04231000 T  0003!040!2
PRT(702) = SEARCH
                             BEGIN  LABEL L;                                        04232000 T  0003!040!2
                                                                          START OF SEGMENT **********   41
                   COMMENT   GT1 AND GT2 ARE INITIALIZED ASSUMMING THAT Q IS ORDERED 04233000 T  0041!000!0
                             BY PAIRS (ARGUMENT,FUNCTION,ARGUMENT,FUNCTION,ETC.)    04234000 T  0041!000!0
                             AND THAT THE FIRST ARGUMENT IS IN Q[4].  FURTHERMORE   04235000 T  0041!000!0
                             THE LENGTH OF Q IS 128.                        ;       04236000 T  0041!000!0
                             INTEGER N,I ;                                          04237000 T  0041!000!0
STACK(F+3) = N
STACK(F+4) = I
                             N ← 64 ;                                               04238000 T  0041!000!0
```

```
                        FOR I ← 66 STEP IF Q[I]<KEY THEN N ELSE - N                                  04239000 T  0041!0000!1
                                WHILE N←N DIV 2 ≥ 1 DO                                               04240000 T  0041!0005!2
                        IF Q[ I ]= KEY THEN GO TO L ;                                                04241000 T  0041!0009!2
                         I ←0;   COMMENT ARGUMENT NOT FOUND,SEARCH=Q[1] ;                            04242000 T  0041!0011!2
                    L.: SEARCH←Q[  I +1] ;                                                           04243000 T  0041!0011!3
                        END      SEARCH  ;                                                           04244000 T  0041!0013!3
                                                                                                  41 IS   16 LONG, NEXT SEG    3


                COMMENT   B2D  CONVERTS THE FOUR LOW ORDER OCTAL DIGITS TO BCD                       04245000 T  0003!0403!2
                          CODE ;                                                                     04246000 T  0003!0403!2
            ALPHA   PROCEDURE   B2D(B); VALUE B; REAL B;                                             04247000 T  0003!0403!2
                          B2D←0&B[45:45:3]&B[39:42:3]&B[33:39:3]&B[27:36:3] ;                        04248000 T  0003!0403!2


                COMMENT   PACK IS A STREAM PROCEDURE WHICH INSERTS THE SYLLABLE                      04249000 T  0003!0411!2
                          INTO THE EDOC ARRAY,   THE SPECIFIC ELEMENT OF EDOC                        04265000 T  0003!0411!2
                          IS  PRECISILY = EDOC[(L DIV 4) DIV 128,(< DIV 4)MOD 128]                   04266000 T  0003!0411!2
                          SYLLABLE  POSITION=(L MOD 4 ), WHERE L IS THE SYLLABLE                     04267000 T  0003!0411!2
                          NUMBER RELATIVE TO THE BEGINNING OF THE SEGMENT;                           04268000 T  0003!0411!2
            STREAM PROCEDURE PACK(WORD,POSITION,SYLLABLE);                                           04269000 T  0003!0411!2
PRT(703) = PACK                                                                                      04270000 T  0003!0411!2
                          VALUE POSITION,SYLLABLE;                                                   04271000 T  0003!0411!2
                          BEGIN                                                                      04272000 T  0003!0411!2
                          DI←WORD ; DI ← DI+POSITION ; DI ← DI+POSITION;                             04273000 T  0003!0412!0
                          SI←LOC SYLLABLE ; SI←SI+6;                                                 04274000 T  0003!0413!2
                          DS←2 CHR ;                                                                 04275000 T  0003!0413!3
                          END  PACK  ;                                                               04276000 T  0003!0414!0


                    PROCEDURE   DEBUG(S);                                                            04277000 T  0003!0414!0
PRT(704) = DEBUG
                          VALUE S;  REAL  S   ;                                                      04278000 T  0003!0414!0
                          IF  STREAMTOG THEN                                                         04279000 T  0003!0414!0
                         IF SINGLTOG THEN                                                            04279100 T  0003!0415!2
                         WRITE(LINE,BUG,B2D(L),COP[S,[42:6]],B2D(S,[36:6]),B2D(S))                   04279200 T  0003!0416!1
PRT(705) = *LIST, LABEL, OR SEGMENT DESCRIPTOR*
                          ELSE                                                                       04279300 T  0003!0429!2
                          WRITE(LINE[DBL],BUG,B2D(L),COP[S,[42:6]],B2D(S,[36:6]),                    04280000 T  0003!0433!2
PRT(706) = *LIST, LABEL, OR SEGMENT DESCRIPTOR*
                                          B2D(S))                                                    04281000 T  0003!0445!2
                                                                                                     04282000 T  0003!0446!0
                          ELSE                                                                       04283000 T  0003!0446!0
                                                                                                     04284000 T  0003!0450!0
                         IF SINGLTOG THEN                                                            04284100 T  0003!0450!0
                         WRITE(LINE,BUG,B2D(L),IF T1←S,[46:2]=1 THEN                                 04284200 T  0003!0451!3
PRT(707) = *LIST, LABEL, OR SEGMENT DESCRIPTOR*
                              BUGGER(S,[36:10]) ELSE WOP[T1],IF T1=1 THEN WOP[1]                     04284300 T  0003!0460!1
                              ELSE B2D(S,[36:10]),B2D(S))                                            04284400 T  0003!0466!1
                         ELSE                                                                        04284500 T  0003!0470!1
                          WRITE(LINE[DBL],BUG,B2D(L),IF T1←S,[46:2]=1 THEN                           04285000 T  0003!0475!2
```

```
PRT(710) = *LIST, LABEL, OR SEGMENT DESCRIPTOR*
                        BUGGER(S,[36:10]) ELSE WOP[T1],IF T1=1 THEN WOP[1]                    04286000 T  0003:048:1
                        ELSE  B2D(S,[36:10]),B2D(S));                                         04287000 T  0003:049:1



            COMMENT     EMIT PLACES SYLLABLES INTO EDOC, CALLS DEBUG FOR                       04288000 T  0003:049:3
                        DEBUGGING OUTPUT ON THE PRINTER, AND CHECKS FOR SEGMENTS               04289000 T  0003:049:3
                        GREATER THAN 4093 SYLLABLES.          ;                               04290000 T  0003:049:3
          PROCEDURE EMIT ( S ) ; VALUE S; REAL   S  ;                                          04291000 T  0003:049:3
                        BEGIN                                                                  04292000 T  0003:049:3
                        IF L  <4092 THEN                                                       04293000 T  0003:049:3
                        BEGIN                                                                  04294000 T  0003:050:1
                        LINKTOG ← TRUE;                                                        04295000 T  0003:050:2
                        PACK( EDOC[L,[36:3 ],L,[39:7]],L,[46:2],S);                            04296000 T  0003:050:0
                        IF DEBUGTOG THEN DEBUG(S);                                             04297000 T  0003:050:3
                        L←L+1;                                                                 04298000 T  0003:050:3
                        END  ELSE                                                              04299000 T  0003:050:2
                        ERR(200)                                                               04300000 T  0003:050:2
            COMMENT     200 EMIT - SEGMENT GREATER THAN 4093 SYLLABLES          *;             04301000 T  0003:051:2
                        END  EMIT ;                                                            04302000 T  0003:051:3

            COMMENT     THE PRINCIPLE FUNCTION OF DEBUG IS TO COMPUTE THE PROPER               04303000 T  0003:051:0
                        PARAMETERS FOR STREAM PROCEDURE BUG;                                   04304000 T  0003:051:0
            COMMENT     EMITD EMITS THE DIA,DIB,TRB SEQUENCE OF CODE, THE                      04305000 T  0003:051:0
                        PREVIOUS SETTING OF THE G-H AND K-V REGISTERS IS COMPARED               04306000 T  0003:051:0
                        THE CURRENT . IF THE G-H,K-V OR BOTH ARE ALREADY SET THEN              04307000 T  0003:051:0
                        THE APPROIATE SYLLABLE(S) ARE OMITTED                                  04308000 T  0003:051:0
                        IF 0 BITS ARE TO BE TRANSFERED THEN NO SYLLABLES ARE                   04309000 T  0003:051:0
                        EMITTED     ;                                                          04310000 T  0003:051:0
          PROCEDURE     EMITD(A,B,T); VALUE A,B,T ; INTEGER A,B,T;                             04311000 T  0003:051:0
                BEGIN LABEL EXIT,NORMAL;                                                       04311010 T  0003:051:0
                                                                                  START OF SEGMENT **********   42
                        REAL Q;                                                                04311020 T  0042:000:0
STACK(F+2) = Q

                        IF T = 15 THEN                                                         04311030 T  0042:000:0
                        BEGIN                                                                  04311040 T  0042:000:1
                        IF A = 33 THEN Q ← 512                                                 04311050 T  0042:000:2
                        ELSE IF A ≠ 18 THEN GO TO NORMAL;                                      04311060 T  0042:000:1
                        IF B = 18 THEN Q ← Q+256                                               04311070 T  0042:000:2
                        ELSE IF B ≠ 33 THEN GO TO NORMAL;                                      04311080 T  0042:000:1
                        EMITO(Q+197);  COMMENT -- THIS GETS OUT FIXED FIELD;                   04311090 T  0042:000:2
                        GO TO EXIT;                                                            04311100 T  0042:001:1
                        END;                                                                   04311110 T  0042:001:2
            NORMAL;                                                                            04311120 T  0042:001:2
                        IF  T≠ 0 THEN                                                          04312000 T  0042:001:2
                        BEGIN                                                                  04313000 T  0042:001:3
                        IF DIALA≠ A THEN                                                       04314000 T  0042:001:0
                         EMIT(((DIALA←A) DIV 6)×512 + ( A← A MOD 6)× 64 + DIA);                04315000 T  0042:001:2
                        IF DIALB≠B  THEN                                                       04316000 T  0042:001:0
                         EMIT(((DIALB←B) DIV 6)×512 + ( B← B MOD 6)× 64 + DIB);                04317000 T  0042:001:2
                         EMIT(TRB+64×T);                                                       04318000 T  0042:002:0
                        DIALA←DIALB←0;                                                         04319000 T  0042:002:0
```

```
                    COMMENT    THE PRECEEDING STATEMENT CAN BE REMOVED FOR OPTIMIZING          04320000 T  00421002712
                               G-H AND R-V REGISTERS,OTHERWISE NO OPTIMIZING OCCURS;           04321000 T  00421002712
                               END       EMITD  ;                                             04322000 T  00421002712
                    EXIT:      END;                                                           04322100 T  00421002712
                                                                                        42 IS   31 LONG, NEXT SEG    3



               PROCEDURE EMITI(E,A,B); VALUE E,A,B; REAL E,A,B;                               04500000 T  00031051210
PRT(711) = EMITI
                         BEGIN LABEL EXIT,IS;                                               04501000 T  00031051210
                                                                                      START OF SEGMENT **********   43
                              INTEGER S,T1,T2;                                              04502000 T  00431000010
STACK(F+2) = S
STACK(F+3) = T1
STACK(F+4) = T2

                              PROCEDURE EMIT21(E,B); VALUE E,B;                             04503000 T  00431000010
PRT(712) = EMIT21
                                   REAL E;                                                  04504000 T  00431000010
                                   BOOLEAN B;                                               04505000 T  00431000010
                                   BEGIN IF E = 0 THEN                                      04506000 T  00431000010
                                             BEGIN IF B THEN EMITO(XCH); END                04507000 T  00431000011
                                        ELSE BEGIN GT1 ← E.ADDRESS;                         04508000 T  00431000211
                                                   IF E ← E.CLASS ≤ INTID THEN              04509000 T  00431000411
                                                        EMITV(GT1)                          04521000 T  00431000610
                                                   ELSE IF E ≤ INTARRAYID THEN              04522000 T  00431000712
                                                        EMITPAIR(GT1,LOD) ELSE              04523000 T  00431000811
                                                   EMITN(GT1)                               04524000 T  00431001010
                                             END                                            04525000 T  00431001112
                                   END;                                                     04526000 T  00431001113



                    IF B = 0 THEN                                                           04526100 T  00431001113
                         BEGIN EMIT21(E,FALSE); GO TO EXIT END;                             04526200 T  00431001211
                    IF STACKCT ≠ 0 THEN GO TO IS;                                           04527000 T  00431001411
                    IF B = 15 THEN                                                          04528000 T  00431001610
                         BEGIN IF A = 33 THEN                                               04529000 T  00431001611
                                   BEGIN EMIT21(E,FALSE);                                   04530000 T  00431001810
                                        EMIT(0); EMITO(INX);                                04531000 T  00431001913
                                        GO TO EXIT;                                         04532000 T  00431002112
                                   END;                                                     04533000 T  00431002113
                              IF A = 18 THEN                                                04534000 T  00431002113
                                   BEGIN EMIT(0);                                           04535000 T  00431002210
                                        EMIT21(E,TRUE);                                     04536000 T  00431002313
                                        EMITO(197);                                         04537000 T  00431002411
                                        GO TO EXIT;                                         04538000 T  00431002512
                                   END;                                                     04539000 T  00431002513
                              GO TO IS;                                                     04540000 T  00431002513
                         END;                                                               04541000 T  00431002610
                    IF B ≤ 10 AND A+B = 48 THEN                                             04542000 T  00431002610
                         BEGIN EMIT21(E,FALSE);                                             04543000 T  00431002811
                              EMITL(2*B-1);                                                 04544000 T  00431003010

PRT(713) = LN
PRT(714) = EXP
PRT(715) = X TO THE I
```

```
                              EMITO(LND);                                                    04545000 T  0043:0033:2
                              GO TO EXIT;                                                    04546000 T  0043:0034:0
                         END;                                                                04547000 T  0043:0034:1
                    IS:  IF (S ← (48-A-B) MOD 6)+B ≤ 39 THEN                                 04548000 T  0043:0034:1
                         BEGIN EMIT21(E,FALSE);                                              04549000 T  0043:0038:0
                              EMIT(T2←(T1←A DIV 6)×512+(A MOD 6)×64+D!A);                    04550000 T  0043:0039:3
                              EMIT(((A+B-1) DIV 6 -T1+1)×512+64×S+37);                       04551000 T  0043:0044:1
                              GO TO EXIT;                                                    04552000 T  0043:0049:3
                         END;                                                                04553000 T  0043:0050:0
                    EMIT(0);                                                                 04554000 T  0043:0050:0
                    EMIT21(E,TRUE);                                                          04555000 T  0043:0051:2
                    EMITD(A,48-B,B);                                                         04556000 T  0043:0052:0
                                                                                             04557000 T  0043:0053:3
               EXIT: END;                                                                    04558000 T  0043:0053:3
                                                                                  43 IS   57 LONG, NEXT SEG    3


          ALPHA PROCEDURE BUGGER(OP); VALUE OP; INTEGER OP;                                  04600000 T  0003:0512:0
            BEGIN INTEGER Q;                                                                 04601000 T  0003:0512:0
                                                                                  START OF SEGMENT **********   44
STACK(F+3) = Q
                    WOP[1]←"    ";                                                           04602000 T  0044:0000:0
                    IF BUGGER←SEARCH(WOP,OP)="    " THEN                                     04603000 T  0044:0001:2
                    IF Q←OP.[44:4]≥9 THEN                                                    04604000 T  0044:0003:3
                   BEGIN BUGGER←POP[IF Q≠10 THEN Q-5 ELSE OP.[42:2]];                        04605000 T  0044:0006:0
                         WOP[1]←(IF Q=10 THEN OP.[39:3]&OP[41:38:1] ELSE                     04606000 T  0044:0010:1
                                   OP.[41:3]&OP[39:38:3])&"  "[24:36:12];                    04607000 T  0044:0014:1
                    END; END BUGGER;                                                         04608000 T  0044:0017:3
                                                                                  44 IS   23 LONG, NEXT SEG    3


               PROCEDURE CHECKDISJOINT(A); VALUE A; INTEGER A;                               04609000 T  0003:0512:0
PRT(716) = CHECKDISJOINT
               BEGIN                                                                         04610000 T  0003:0512:0
                    IF LEVEL > SUBLEVEL+1 THEN                                               04611000 T  0003:0512:0
                    BEGIN                                                                    04612000 T  0003:0513:2
                         EMIT(0);                                                            04613000 T  0003:0513:3
                         EMITPAIR(A, STD);                                                   04614000 T  0003:0514:1
                    END;                                                                     04615000 T  0003:0515:3
                    EMITN(A);                                                                04616000 T  0003:0515:3
               END CHECKDISJOINT;                                                            04617000 T  0003:0516:0


                COMMENT THIS SECTION CONTAINS MISCELLANEOUS SERVICE ROUTINES;                05000000 T  0003:0516:1
                COMMENT STEPI AND STEPIT ARE SHORT CALLS ON TABLE;                           05001000 T  0003:0516:1
                PROCEDURE STEPIT; ELCLASS ← TABLE(I←I+1);                                    05002000 T  0003:0516:1


                INTEGER PROCEDURE STEPI; STEPI←ELCLASS←TABLE(I←I+1);                         05003000 T  0003:0519:3
```

```
PRT(717) = STEP1


                COMMENT TAKE FETCHS A WORD FROM INFO;                                   05004000 T  00031052:2
                REAL PROCEDURE TAKE(INDEX); VALUE INDEX; INTEGER INDEX;                 05005000 T  00031052:2
                        TAKE ← INFO[INDEX.LINKR,INDEX.LINKC];                           05006000 T  00031052:2



                COMMENT PUT PLACES A WORD INTO INFO;                                    05007000 T  00031053:2
                PROCEDURE PUT(WORD,INDEX); VALUE WORD,INDEX; REAL WORD,INDEX;           05008000 T  00031053:2
                        INFO[INDEX.LINKR,INDEX.LINKC] ← WORD;                           05009000 T  00031053:2



                COMMENT FLAG FLAGS ERROR MESSAGES, COUNTS THEM AND SUPPRESS FUTURE      05010000 T  00031053:2
                        ERROR MESSAGES UNTIL THE COMPILER THINKS IT HAS RECOVERED;       05011000 T  00031053:2
                PROCEDURE FLAG(ERRNUM); VALUE ERRNUM; INTEGER ERRNUM;                   05012000 T  00031053:2
                        BEGIN                                                           05013000 T  00031053:2
                   COMMENT WRITERROR IS THE STREAM PROCEDURE WHICH ACTUALLY PRODUCES     05014000 T  00031053:2
                        THE ERROR MESSAGE ON THE PRINTER;                               05015000 T  00031053:2
                STREAM PROCEDURE WRITERROR(RMT,ERRNUM,ACCUM,LINE,COUNT,LSTSEQ);         05016000 T  00031053:2
                                                                        START OF SEGMENT **********   45
PRT(720) = WRITERROR
                        VALUE ERRNUM,COUNT;                                             05017000 T  00451000:0
                        BEGIN                                                           05018000 T  00451000:0
                             DI:=LINE; 11(DS:=8 LIT "        "); %BLANK LINE        05019000 T  00451000:0
                             SI ←LSTSEQ; SI ← SI-8; DS ←WDS;                            05020000 T  00451000:2
                        DS:=24 LIT "     <<<<<<<<<<<<<<<<<<<<<<<"; %SET FLAG            05021000 T  00451000:1
                        SI ← LSTSEQ; DI ← LSTSEQ; DI ← DI-8; DS ← WDS;                  05023000 T  00451000:0
                        DI←LINE; SI←RMT; SI←SI+7;                                       05024000 T  00451000:2
                        IF SC="1" THEN                                                  05024100 T  00451000:3
                        BEGIN SI←LSTSEQ; DS←10 LIT "NEAR LINE ";                        05024200 T  00451000:0
                             7(IF SC>"0" THEN JUMP OUT;                                 05024300 T  00451001:1
                                  SI←SI+1; TALLY←TALLY+1);                              05024400 T  00451001:3
                             RMT←TALLY; DS←8 CHR; DI←DI-RMT;                            05024500 T  00451001:2
                        END ELSE DI←DI+7;                                               05024600 T  00451001:0
                        DS←14 LIT " ERROR NUMBER ";                                     05025000 T  00451001:1
                        SI ← LOC ERRNUM; DS ← 3 DEC; COMMENT CONVERT ERRNUM;            05026000 T  00451001:1
                        DS ← 4 LIT " -- ";                                              05027000 T  00451001:2
                        SI ← ACCUM; SI ← SI+3; DS ← COUNT CHR;                          05028000 T  00451001:3
                           COMMENT PLACE ALPHA IN BUFFER;                               05029000 T  00451001:1
                        DS ← LIT ":"                                                    05030000 T  00451001:1
                END WRITERROR;                                                          05031000 T  00451001:2



                IF ERRORTOG THEN % DO NOTHING IF WE SUPPRESS MSSGS.                     05032000 T  00451001:3
                     BEGIN                                                              05033000 T  00451002:0
                     SPECTOG := FALSE;                                                  05034000 T  00451002:1
                     ERRORCOUNT := ERRORCOUNT+1; COMMENT COUNT ERRORS;                  05035000 T  00451002:3
                     IF NOT(LISTER OR REMOTOG) THEN                                     05036000 T  00451002:1
                          BEGIN                                                         05037000 T  00451002:3
```

```
                                EDITLINE(LIN,FCR,"    ",0,0,MEDIUM,0);      %114-          05038000 P  0045:0024:0
                               MOVE(1,INFO[LASTSEQROW,LASTSEQUENCE],LIN[12]);               05039500 T  0045:0026:1
                               IF NOHEADING THEN DATIME; WRITELINE;                         05039600 T  0045:0029:3
                               END;                                                         05041000 T  0045:0042:0
                    COMMENT    PRINT CARDIMAGE IF WE ARE NOT LISTING;                       05042000 T  0045:0042:0
                               ACCUM[1] := Q;    COMMENT RESTORE ACCUMULATOR;               05043000 T  0045:0042:0
                               WRITERROR(REMOTOG,ERRNUM,ACCUM[1],LIN,Q,[12:6],            05044000 T  0045:0043:3
                                    INFO[LASTSEQROW,LASTSEQUENCE]);                         05045000 T  0045:0046:0
                         IF REMOTOG THEN WRITE(REMOTE,10,LIN[*]);                           05045900 T  0045:0047:3
                         IF NOT NOHEADING THEN BEGIN WRITE (LINE); WRITELINE; END;          05046000 T  0045:0052:1
                               ERRORTOG := FALSE; COMMENT INHIBIT MESSAGES;                 05047000 T  0045:0067:3
               IF PUNCHTOG THEN                                                             05048000 T  0045:0068:1
                    BEGIN                                                                   05049000 T  0045:0069:2
                         STREAM PROCEDURE PUNCH(FL,ST);                                     05050000 T  0045:0069:3
PRT(721) = *SEGMENT DESCRIPTOR*

                                                                  START OF SEGMENT **********   46

PRT(722) = PUNCH
                              VALUE ST;                                                     05051000 T  0046:0000:0
                                BEGIN                                                       05052000 T  0046:0000:0
                                DI := FL;                                                   05053000 T  0046:0000:0
                                SI := ST;                                                   05054000 T  0046:0000:0
                                DS := 9 WDS                                                 05055000 T  0046:0000:1
                                END PUNCH;                                                  05056000 T  0046:0000:1




                         PUNCH(PNCH(0),FCR);                                                05057000 T  0046:0001:2
                         MOVE(1,INFO[LASTSEQROW,LASTSEQUENCE], PNCH(9));                    05058000 T  0046:0004:1
                              WRITE(PNCH)                                                   05059000 T  0046:0010:0
                         END                                                                05060000 T  0046:0011:2
                                   END END FLAG;                                            05101000 T  0046:0014:0
PRT(723) = *SEGMENT DESCRIPTOR*

                                                                  46 IS    15 LONG, NEXT SEG    45
                                                                  45 IS    72 LONG, NEXT SEG     3




                              LABEL ENDOFITALL;                                             05101100 T  0003:0535:2
                    COMMENT ERR, IS THE SAME AS FLAG EXCEPT THAT IT MAKES AN ATTEMPT TO     05102000 T  0003:0535:2
                              RECOVER FROM ERROR SITUATIONS BY SEARCHING FOR A              05103000 T  0003:0535:2
                              SEMICOLON, END, OR BEGIN;                                     05104000 T  0003:0535:2
                    PROCEDURE ERR(ERRNUM); VALUE ERRNUM; INTEGER ERRNUM;                    05105000 T  0003:0535:2
                              BEGIN FLAG(ERRNUM);                                           05106000 T  0003:0535:2
                                   I := I-1;                                                05107000 T  0003:0536:1
                                   IF ERRNUM=200 THEN GO TO ENDOFITALL;                     05107100 T  0003:0538:0
PRT(724) = ENDOFITALL
                                   IF ERRNUM=611 THEN GO TO ENDOFITALL;%ERRMAX EXCEEDED,    05107200 T  0003:0541:3
                                   DO IF STEPI = BEGINV THEN STMT UNTIL                     05108000 T  0003:0545:2
                                      ELCLASS = ENDV OR ELCLASS = SEMICOLON END ERR;        05109000 T  0003:0547:2




                    DEFINE ERROR = ERR#;  COMMENT ERROR IS A SYNONM FOR ERR;                05110000 T  0003:0549:3
                    COMMENT CHECKER IS A SMALL PROCEDURE THAT CHECKS TO SEE THAT THE        05111000 T  0003:0549:3
```

```
                        UPLEVEL ADDRESSING CONVENTIONS ARE OBEYED;                                  05112000 T  0003:054913
                    PROCEDURE CHECKER(ELBATWORD); VALUE ELBATWORD; REAL ELBATWORD;                  05113000 T  0003:054913
PRT(725) = CHECKER
                        BEGIN                                                                       05114000 T  0003:054913
                             IF MODE ≥ 2 THEN                                                       05115000 T  0003:054913
                             IF GT1 ← ELBATWORD.LVL ≥ FRSTLEVEL THEN                                05116000 T  0003:055011
                             IF GT1 < SUBLEVEL THEN                                                 05117000 T  0003:055312
                        IF ELBATWORD.[9:2] ≠ 1                                                      05118000 T  0003:055410
                              THEN BEGIN FLAG(101); ERRORTOG ← TRUE END                             05119000 T  0003:055513
                        END CHECKER;                                                                05120000 T  0003:055810



                    COMMENT GIT IS USED TO OBTAIN THE INDEX TO ADDITIONAL INFORMATION               05121000 T  0003:055810
                            GIVEN THE LINK TO THE ELBAT WORD;                                       05122000 T  0003:055810
                    INTEGER PROCEDURE GIT(L); VALUE L; REAL L;                                      05123000 T  0003:055810
                            GIT ← TAKE(L).INCR+L.LINK;                                              05124000 T  0003:055810



                    COMMENT GNAT IS USED TO OBTAIN THE PRT ADDRESS OF A GIVEN DESCRIPTOR,           05125000 T  0003:056410
                            IF THE ADDRESS HAS NOT BEEN ASSIGNED, THEN IT USES                      05126000 T  0003:056410
                            GETSPACE TO OBTAIN THE PRT ADDRESS;                                     05127000 T  0003:056410
                    INTEGER PROCEDURE GNAT(L); VALUE L; REAL L;                                     05128000 T  0003:056410
                            BEGIN                                                                   05129000 T  0003:056410
                            REAL A;                                                                 05130000 T  0003:056410
                                                                                          START OF SEGMENT **********   47
STACK(F+3) = A
                            IF GNAT ←(A←TAKE(L)).ADDRESS=0                                          05131000 T  0047:000010
                            THEN PUT(A&(GNAT:=GETSPACE(TRUE,L.LINK+1))[16:37:11],L)                 05132000 T  0047:000210
                            END GNAT;                                                               05133000 T  0047:000712
                                                                                            47 IS   10 LONG, NEXT SEG    3



                    COMMENT PASSFILE COMPILES CODE THAT BRINGS TO TOP OF STACK A DESCRIPTOR         05134000 T  0003:056410
                            POINTING AT THE I/O DESCRIPTOR (ON TOP).  IT HANDLES                    05135000 T  0003:056410
                            SUPERFILES AS WELL AS ORDINARY FILES;                                   05136000 T  0003:056410
                    PROCEDURE PASSFILE;                                                             05137000 T  0003:056410
PRT(726) = PASSFILE
                        BEGIN INTEGER ADDRES;                                                       05138000 T  0003:056410
                                                                                          START OF SEGMENT **********   48
STACK(F+2) = ADDRES
                            CHECKER(ELBAT[I]);                                                      05139000 T  0048:000010
                            ADDRES ← ELBAT[I].ADDRESS;                                              05140000 T  0048:000112
                            IF ELCLASS = SUPERFILEID                                                05141000 T  0048:000211
                              THEN BEGIN                                                            05142000 T  0048:000211
                                   BANA; EMITN(ADDRES); EMITO(LOD) END                              05143000 T  0048:000313
                              ELSE BEGIN                                                            05144000 T  0048:000513
                                   IF NOT BOOLEAN(ELBAT[I],FORMAL) THEN EMITL(5);                   05145000 T  0048:000610
                                   STEPIT;                                                          05146000 T  0048:000811
                                   EMITN(ADDRES) END END PASSFILE;                                  05147000 T  0048:000912
                                                                                            48 IS   13 LONG, NEXT SEG    3
```

```
                PROCEDURE PASSMONFILE(ADDRESS);                                          05148000 T  0003!056!0
PRT(727) = PASSMONFILE
                    VALUE ADDRESS ;                                                      05149000 T  0003!056!0
                     REAL ADDRESS ;                                                      05150000 T  0003!056!0
                BEGIN COMMENT PASSMONFILE GENERATES CODE TO PASS THE MONITOR          05151000 T  0003!056!0
                     FILE TO PRINT;                                                      05152000 T  0003!056!0
                     IF ADDRESS < 768 OR ADDRESS > 1023                                  05153000 T  0003!056!0
                     THEN EMITL(5);                                                      05154000 T  0003!056!0
                     EMITN(ADDRESS);                                                     05155000 T  0003!056!0
                END PASSMONFILE;                                                         05156000 T  0003!056!1


                PROCEDURE PASFILE;                                                       05157000 T  0003!056!2
PRT(730) = PASFILE
                BEGIN COMMENT PASFILE PASSES THE LAST THREE PARAMETERS TO KEN          05158000 T  0003!056!2
                     MEYERS FOR THE LOCK, CLOSE, AND REWIND STATEMENTS;                  05159000 T  0003!056!2
                     DEFINE ELBATWORD = RR1#;  COMMENT ELBATWORD CONTAINS THE            05160000 T  0003!056!2
                                                                          START OF SEGMENT **********  49
                                       ELBATWORD FOR THE FILE BEING                      05161000 T  0049!000!0
                                       OPERATED ON;                                      05162000 T  0049!000!0
                     DEFINE LTEMP = RR2#;  COMMENT LTEMP IS USED TO HOLD THE L          05163000 T  0049!000!0
                                   REGISTER SETTING FOR THE SAVE OR                      05164000 T  0049!000!0
                                   RELEASE LITERAL THAT GETS PASSED TO                   05165000 T  0049!000!0
                                   KEN MEYERS;                                           05166000 T  0049!000!0
                     EMITO(MKS);        L←(LTEMP←L)+1;     EMITL(0);                  05167000 T  0049!000!0
                     EMITL(2);          CHECKER(ELBATWORD←ELBAT[I]);                    05168000 T  0049!000!2
                     IF RRB1←(RRB2← ELCLASS   = SUPERFILEID)OR                           05169000 T  0049!000!3
                     BOOLEAN(ELBATWORD,FORMAL)                                           05170000 T  0049!000!1
                     THEN EMITO(LNG);                                                    05171000 T  0049!000!1
                     IF RRB2                                                             05172000 T  0049!000!3
                     THEN BANA                                                           05173000 T  0049!000!3
                     ELSE STEPIT;                                                        05174000 T  0049!001!0
                     EMITN(ELBATWORD,ADDRESS);                                           05175000 T  0049!001!3
                     IF RRB2                                                             05176000 T  0049!001!2
                     THEN EMITO(LOD);                                                    05177000 T  0049!001!2
                     IF RRB1                                                             05178000 T  0049!001!1
                     THEN EMITO(INX);                                                    05179000 T  0049!001!1
                     EMITL(4); EMITV(14);                                                05180000 T  0049!001!0
                END PASFILE;                                                             05181000 T  0049!001!3
                                                                          49 IS   18 LONG, NEXT SEG    3


          COMMENT  CHECKPRESENCE CAUSES THE CORRECT CODE TO BE GENERATED TO CAUSE        05182000 T  0003!056!2
                PRESENCE BIT INTERRUPTS ON I/O DESCRIPTORS;                              05183000 T  0003!056!2
          PROCEDURE CHECKPRESENCE;                                                       05184000 T  0003!056!2
PRT(731) = CHECKPRESENCE
                BEGIN                                                                    05185000 T  0003!056!2
                     EMITO(DUP); EMITO(LOD); EMITL(0); EMITO(CDC); EMITO(DEL);           05186000 T  0003!056!2
                END CHECKPRESENCE;                                                       05187000 T  0003!057!1


          COMMENT  PROCEDURE PASSLIST WILL BRING THE LIST PROGRAM DESCRIPTOR             05187500 T  0003!057!2
```

```
                    TO TOP OF STACK FOR A SUBSCRIPTED LIST ID OR SIMPLE ID;                05187510 T  0003:057:2
               PROCEDURE PASSLIST;                                                         05187520 T  0003:057:2
PRT(732) = PASSLIST
                    BEGIN    .                                                             05187530 T  0003:057:2
                    INTEGER  LISTADDRESS;                                                  05187540 T  0003:057:2
                                                                                 START OF SEGMENT **********   50
STACK(F+2) = LISTADDRESS
               COMMENT  PASSLIST ASSUMES T POINTING AT LIST ID;                            05187550 T  0050:000:0
                    CHECKER(ELBAT[T]);                                                     05187560 T  0050:000:0
                    LISTADDRESS:=ELBAT[T],ADDRESS;                                         05187570 T  0050:001:2
                    IF ELCLASS = SUPERLISTID THEN % SUBSCRIPTED LIST ID.                   05187580 T  0050:002:1
                         BEGIN                                                             05187590 T  0050:003:2
                         BANA;  EMITN(LISTADDRESS);  EMITO(LOD);                           05187600 T  0050:003:3
                         END                                                               05187610 T  0050:005:3
                    ELSE BEGIN EMITL(LISTADDRESS); STEPIT END;                             05187620 T  0050:005:3
                    END OF PASSLIST;                                                       05187630 T  0050:007:3
                                                                                  50 IS   10 LONG, NEXT SEG    3


               REAL PROCEDURE TAKEFRST;                                                    05188000 T  0003:057:2
PRT(733) = TAKEFRST
                         TAKEFRST ← TAKE(ELBAT[T].LINK+ELBAT[T].INCR);                     05189000 T  0003:057:2


               COMMENT STUFFF DIALS THE F-REGISTER INTO THE F-REGISTER FIELD OF A          05196000 T  0003:057:0
                         DESCRIPTOR,  THE DESCRIPTOR REMAINS ON THE TOP OF THE             05197000 T  0003:057:0
                         STACK;                                                            05198000 T  0003:057:0
               PROCEDURE STUFFF(ADDRESS); VALUE ADDRESS; INTEGER ADDRESS;                  05199000 T  0003:057:0
PRT(734) = STUFFF
                         BEGIN                                                             05200000 T  0003:057:0
                              EMITPAIR(ADDRESS,LOD);                                       05201000 T  0003:057:0
                                   EMITN(512);                                             05202000 T  0003:058:0
                                   EMITD(33,18,15) END STUFFF;                             05203000 T  0003:058:1


               COMMENT    LOCAL IS USED TO SEE WHETHER OR NOT A LABEL IS LOCAL TO OUR     05204000 T  0003:058:0
                          PRESENT CODE;                                                   05205000 T  0003:058:0
               BOOLEAN PROCEDURE    LOCAL(ELBATWORD);                                  05206000 T  0003:058:0
PRT(735) = LOCAL
                         VALUE ELBATWORD; REAL ELBATWORD;                                  05207000 T  0003:058:0
                         BEGIN IF ELBATWORD.LVL = LEVEL AND                                05208000 T  0003:058:0
                               NOT BOOLEAN(ELBATWORD.FORMAL) THEN                          05209000 T  0003:058:0
                                   LOCAL ← TRUE END    LOCAL;                            05210000 T  0003:058:3


               COMMENT PASSFORMAT COMPILES CODE THAT PASSES A FORMAT,  TWO ITEMS ARE       05211000 T  0003:058:2
                         PASSED - THE ARRAY REFERENCING FORMAT TABLE AND THE               05212000 T  0003:058:2
                         STARTING INDEX.  THE ROUTINE HANDLES SUPERFORMATS ALSO;           05213000 T  0003:058:2
               PROCEDURE PASSFORMAT;                                                       05214000 T  0003:058:2
PRT(736) = PASSFORMAT
```

```
                        BEGIN INTEGER ADRES;                                                                    05215000 T  0003:058912
                                                                                                              START OF SEGMENT **********   51
STACK(F+2) = ADRES

                            CHECKER(ELBAT[I]);                                                                  05216000 T  0051:000010
                            ADRES ← ELBAT[I].ADDRESS;                                                           05217000 T  0051:000112
                            IF BOOLEAN(ELBAT[I],FORMAL)                                                         05218000 T  0051:000211
                               THEN BEGIN EMITV(ADRES); ADRES ← ADRES-1 END                                     05219000 T  0051:000211
                               ELSE BEGIN                                                                       05220000 T  0051:000610
                                 IF TABLE(I) = SUPERFRMTID                                                      05221000 T  0051:000611
                                   THEN EMITL(TAKEFRST) ELSE EMITL(ELBAT[I].INCR)                               05222000 T  0051:000712
                               END;                                                                             05223000 T  0051:001010
                            IF TABLE(I) = SUPERFRMTID                                                           05224000 T  0051:001112
                               THEN BEGIN  BANA; I ← I-1;                                                       05225000 T  0051:001210
                                 EMITO(SSP); EMITO(ADD); EMITV(ADRES) END;                                      05226000 T  0051:001411
                            EMITPAIR(ADRES,LOD) END PASSFORMAT;                                                 05227000 T  0051:001712
                                                                                                          51 IS   21 LONG, NEXT SEG    3


                    COMMENT  STREAMWORDS EITHER RESERVES OR UNRESERVES STREAM RESERVED                          05228000 T  0003:058912
                             WORDS - IT COMPLEMENTS THEIR STATE;                                                05229000 T  0003:058912
                    PROCEDURE STREAMWORDS;                                                                      05230000 T  0003:058912
PRT(737) = STREAMWORDS
                            BEGIN GT1 ← 0;                                                                      05231000 T  0003:058912
                             DO BEGIN                                                                           05232000 T  0003:059011
                              INFO[1,GT1].LINK←STACKHEAD[GT2←(T←INFO[1,GT1]).ADDRESS];                          05233000 T  0003:059112
                              STACKHEAD[GT2] ← T.LINK;                                                          05234000 T  0003:059513
                              GT1 ← GT1+2;                                                                      05235000 T  0003:059811
                             END UNTIL BOOLEAN(T,FORMAL) END STREAMWORDS;                                       05236000 T  0003:060010
```

```
                    STREAM PROCEDURE DEBUGDESC(LIN,PRT,TYP,RELAD,SGNO);                                         05237000 T  0003:060113
PRT(740) = DEBUGDESC
                                   VALUE PRT,TYP,RELAD,SGNO;                                                    05237500 T  0003:060113
                        BEGIN  LOCAL COUNT;                                                                     05238000 T  0003:060113
                        DI:=LIN; DS:=6 LIT"  PRT("; SI:=LOC PRT; SI:=SI+4; TALLY:=4;                            05238500 T  0003:060210
                        3(IF SC="0" THEN % DONT PRINT LEADING ZEROES,                                           05239000 T  0003:060410
                             BEGIN SI:=SI+1; TALLY:=TALLY+63  END ELSE JUMP OUT);                               05239500 T  0003:060411
                        COUNT:=TALLY; DS:=COUNT CHR;                                                            05240000 T  0003:060712
                        DS:= 31 LIT") = SEGMENT DESCRIPTOR, TYPE = ";                                           05240500 T  0003:060713
                        SI:=LOC TYP; SI:=SI+7; DS:=CHR; % TYPE,                                                 05241000 T  0003:061210
                        DS:=21 LIT", RELATIVE ADDRESS = ";                                                      05241500 T  0003:061211
                        SI:=LOC RELAD; SI:=SI+4; DS:=4 CHR; % REL. ADDR,                                        05242000 T  0003:061513
                        DS:=19 LIT", SEGMENT NUMBER = ";                                                        05242500 T  0003:061611
                        SI:=LOC SGNO; SI:=SI+4; DS:=4 CHR; DS:=LIT".";                                          05243000 T  0003:061912
                        END DEBUGDESC;                                                                          05243500 T  0003:062011


                    REAL PROCEDURE PROGDESCBLDR(TYPE,RELAD,SPAC);                                               05245000 T  0003:062011
                      COMMENT  THIS PROCEDURE BUILDS PDPRT AS DESCRIBED ABOVE.IT IS                             05246000 T  0003:062011
                               CONCERNED WITH TYPE 1 ENTRIES.THE INFORMATION FURNISHED                          05247000 T  0003:062011
                               BY PDPRT ALLOWS A DRUM DESCRIPTOR TO BE BUILT FOR EACH                           05248000 T  0003:062011
```

```
SEGMENT AND A PSEUDO PROGRAM DESCRIPTOR TO BE BUILT INTO                05249000 T  0003:0620:1
THE OBJECT TIME PRT. THE 3 PARAMETERS FUNCTION AS FOLLOWS:              05250000 T  0003:0620:1
        TYPE     ---  THIS 2 BIT QUANTITY FURNISHES THE MODE           05251000 T  0003:0620:1
                      AND ARGUMENT BIT FOR THE PROGRAM                  05252000 T  0003:0620:1
                      DESCRIPTOR TO BE BUILT.                           05253000 T  0003:0620:1
        RELAD    ---  RELATIVE WORD ADDRESS WITHIN SEGMENT              05254000 T  0003:0620:1
        SPAC     ---  IF=0 THEN A SPACE MUST BE OBTAINED                05255000 T  0003:0620:1
                      IF≠0 THEN SPACE IS ALREADY GOTTEN                 05256000 T  0003:0620:1
ALL PROGRAM DESCRIPTORS REQUIRE A PERMANENT SPACE IN PRT.               05257000 T  0003:0620:1
PDINX IS THE INDEX FOR PDPRT.IT IS GLOBAL AND 0 INITIALLY;              05258000 T  0003:0620:1
  VALUE TYPE,RELAD,SPAC;REAL TYPE,RELAD,SPAC;                           05259000 T  0003:0620:1
BEGIN IF SPAC=0 THEN SPAC:=GETSPACE(TRUE,-2);% DESCR.                   05260000 T  0003:0620:1
     PDPRT[PDINX,[37:5],PDINX,[42:6]]←0&RELAD[18:36:10]                 05261000 T  0003:0624:0
           &SGNO[28:38:10]&TYPE[4:46:2]&SPAC[8:38:10];                  05262000 T  0003:0627:3
    IF DEBUGTOG THEN                                                    05263000 T  0003:0631:2
         BEGIN                                                          05263500 T  0003:0631:3
         BLANKET(14,LIN);                                               05264000 T  0003:0632:0
         DEBUGDESC(LIN,B2D(SPAC),TYPE,B2D(RELAD),B2D(SGNO));            05264500 T  0003:0633:3
         IF NOHEADING THEN DATIME; WRITELINE;                           05265000 T  0003:0637:2
         END;                                                           05265100 T  0003:0648:1
     PDINX←PDINX+1;PROGDESCBLDR.SPAC END PROGDESCBLDR;                  05266000 T  0003:0648:1


COMMENT  DOTSYNTAX ANALYSES THE SYNTAX OF A PARTIAL WORD DESIGNATOR.    05267000 T  0003:0653:2
         IT REPORTS IF AN ERROR IS FOUND.  IT RETURNS WITH THE          05268000 T  0003:0653:2
         LITERALS INVOLVED;                                             05269000 T  0003:0653:2
    BOOLEAN PROCEDURE DOTSYNTAX(FIRST,SECOND);                          05270000 T  0003:0653:2
PRT(741) = DOTSYNTAX
         INTEGER FIRST,SECOND;                                          05271000 T  0003:0653:2
         BEGIN                                                          05272000 T  0003:0653:2
           LABEL EXIT;                                                  05273000 T  0003:0653:2
                                                          START OF SEGMENT **********   52
            IF STEPI = FIELDID THEN % GET INFO FROM INFO        %117-   05273100 C  0052:0000:0
              BEGIN                                             %117-   05273200 C  0052:0001:2
                FIRST := ELBAT[I],SBITF;                        %117-   05273300 C  0052:0001:3
                SECOND := ELBAT[I],NBITF;                       %117-   05273400 C  0052:0003:2
                GO TO EXIT;                                     %117-   05273500 C  0052:0005:2
              END                                               %117-   05273600 C  0052:0005:3
            ELSE                                                %117-   05273700 C  0052:0005:3
              IF ELCLASS = LFTBRKET THEN                        %117-   05273800 C  0052:0005:3
                IF STEPI = FIELDID THEN                         %117-   05273900 C  0052:0006:1
                  BEGIN                                         %117-   05274000 P  0052:0008:0
                    FIRST := ELBAT[I],SBITF;                    %117-   05274100 C  0052:0008:1
                    SECOND := ELBAT[I],NBITF;                   %117-   05274200 C  0052:0010:1
                    IF STEPI = RTBRKET THEN                     %117-   05274300 C  0052:0012:0
                      GO TO EXIT;                               %117-   05274400 C  0052:0013:2
                  END                                           %117-   05274500 C  0052:0013:3
                ELSE                                            %117-   05274600 C  0052:0013:3
                  IF ELCLASS = LITNO THEN                       %117-   05275000 P  0052:0013:3
              IF STEPI = COLON THEN                                     05276000 T  0052:0015:2
              IF STEPI = LITNO THEN                                     05277000 T  0052:0016:1
              IF STEPI = RTBRKET THEN                                   05278000 T  0052:0018:0
    COMMENT IF TESTS ARE PASSED THEN SYNTAX IS CORRECT;                 05279000 T  0052:0019:3
              IF (FIRST ≠ ELBAT[I-3].ADDRESS) ×                         05280000 T  0052:0019:3
                (SECOND ≠ ELBAT[I-1].ADDRESS)≠0 THEN                    05281000 T  0052:0022:0
```

```
                        IF FIRST + SECOND ≤ 48 THEN                                    05282000 T  0052:0025:2
                  COMMENT IF TESTS ARE PASSED THEN RANGES OF LITERALS ARE O.K.;        05283000 T  0052:0027:2
                          GO TO EXIT;                                                  05284000 T  0052:0027:2
                        ERR(114);  COMMENT ERROR IF SYNTAX OR RANGE FAILS;             05285000 T  0052:0027:3
                        DOTSYNTAX ← TRUE; EXIT; END DOTSYNTAX;                         05286000 T  0052:0028:0
                                                                                  52 IS   32 LONG, NEXT SEG    3



        BOOLEAN PROCEDURE CHECK(ELBATCLASS,ERRORNUMBER);                               05287000 T  0003:0653:2
PRT(742) = CHECK
                        VALUE ELBATCLASS,ERRORNUMBER;                                  05288000 T  0003:0653:2
                         REAL ELBATCLASS,ERRORNUMBER;                                  05289000 T  0003:0653:2
                    BEGIN COMMENT CHECK COMPARES ELBATCLASS WITH TABLE(I).  IF THEY     05290000 T  0003:0653:2
                        ARE NOT EQUAL, CHECK IS SET TRUE AND THE ERROR ROUTINE IS      05291000 T  0003:0653:2
                        CALLED PASSING ERRORNUMBER.  IF THEY ARE EQUAL  CHECK IS SET   05292000 T  0003:0653:2
                        FALSE;                                                         05293000 T  0003:0653:2
                        IF CHECK←(ELBATCLASS ≠ TABLE(I))                               05294000 T  0003:0653:2
                        THEN ERR(ERRORNUMBER);                                         05295000 T  0003:0654:1
                    END;                                                               05296000 T  0003:0657:2



        BOOLEAN PROCEDURE RANGE(LOWER,UPPER);                                          05297000 T  0003:0659:2
PRT(743) = RANGE
                        VALUE LOWER,UPPER;                                             05298000 T  0003:0659:2
                         REAL LOWER,UPPER;                                             05299000 T  0003:0659:2
                     COMMENT RANGE TESTS THE CLASS OF THE ITEM IN ELBAT[I] TO SEE IF   05300000 T  0003:0659:2
                    IT IS GREATER THAN OR EQUAL TO LOWER OR LESS THAN OR EQUAL TO      05301000 T  0003:0659:2
                    UPPER AND SETS RANGE TO TRUE OR FALSE ACCORDINGLY.  THE ITEMS      05302000 T  0003:0659:2
                    CLASS MUST BE IN ELCLASS;                                          05303000 T  0003:0659:2
                    RANGE←ELCLASS ≥ LOWER AND ELCLASS ≤ UPPER;                         05304000 T  0003:0659:2



    COMMENT GET OBTAINS A SYLLABLE FROM EDOC, THE ARRAY INTO WHICH CODE IS             05305000 T  0003:0664:0
            EMITTED;                                                                   05306000 T  0003:0664:0
    INTEGER PROCEDURE GET(L); VALUE L; REAL L;                                         05307000 T  0003:0664:0
            BEGIN                                                                      05308000 T  0003:0664:0
              INTEGER STREAM PROCEDURE GETSYL(W,S); VALUE S;                           05309000 T  0003:0664:0
                                                                              START OF SEGMENT **********   53
PRT(744) = GETSYL
                    BEGIN DI ← LOC GETSYL; DI ← DI+6;                                  05310000 T  0053:0000:0
                          SI ← W; SI ← SI+S; SI ← SI+S; DS ← 2 CHR END;                05311000 T  0053:0000:1



                    GET ← GETSYL(EDOC[L,[36:3],L,[39:7]],L,[46:2]) END GET;            05312000 T  0053:0003:2
                                                                                  53 IS   10 LONG, NEXT SEG    3



    COMMENT  CALL SWITCH PERFORMS THE FINAL MESS OF GETTING A PROPER DE-               05313000 T  0003:0664:0
```

```
                        SCRIPTOR TO THE TOP OF THE STACK;                                                    05314000 T  0003:066410
                PROCEDURE CALLSWITCH(H); VALUE H; REAL H;                                                    05315000 T  0003:066410
PRT(745) = CALLSWITCH
                    BEGIN EMITV(GNAT(H)); EMITO(PRTE); EMITO(LOD) END CALLSWITCH;                            05316000 T  0003:066410



            REAL STREAM PROCEDURE GETALPHA(INFO,INDEX,SIZE);                                                 05317000 T  0003:066810
PRT(746) = GETALPHA
                                    VALUE SIZE ;                                                             05318000 T  0003:066810
                BEGIN COMMENT GETALPHA PICKS ALPHA CHARACTERS OUT OF INFO AND                                05319000 T  0003:066810
                    FORMATS THE ID WORD THAT IS PASSED TO PRINT1.  THE FIRST                                 05320000 T  0003:066810
                    CHARACTER CONTAINS THE SIZE.   THE NEXT CHARACTER CONTAINS THE                           05321000 T  0003:066810
                    ALPHA LEFT JUSTIFIED WITH TRAILING ZEROS;                                                05322000 T  0003:066810
                    DI←LOC GETALPHA;     DS←8 LIT"0       ";    DI←DI-7;                                   05323000 T  0003:066810
                    SI←INFO,INDEX;       SI←SI+3;              DS←SIZE CHR;                               05324000 T  0003:066913
                END GETALPHA;                                                                                05325000 T  0003:067011



            PROCEDURE WRITEPRT(PORS,N,GS); VALUE PORS,N,GS; INTEGER PORS,N,GS;                               05325010 T  0003:067113
PRT(747) = WRITEPRT
                BEGIN                                                                                        05325020 T  0003:067113
                LABEL EXIT;                                                                                  05325030 T  0003:067113
                                                                                          START OF SEGMENT ********** 54
                STREAM PROCEDURE FILLIT(LIN,PORS,CELL,N,ID);                                                 05325040 T  0054:000010
PRT(750) = FILLIT
                                    VALUE PORS,CELL,N;                                                       05325050 T  0054:000010
                BEGIN                                                                                        05325060 T  0054:000010
                LOCAL COUNT;                                                                                 05325070 T  0054:000010
                LABEL M0,M1,M2,M3,M4,M5,M6,M7,XIT;                                                           05325080 T  0054:000010
                SI:=LOC PORS; SI:=SI+3; DI:=LIN; % "PRT" OR "STACK".                                         05325090 T  0054:000010
                IF SC="P" THEN                                                                               05325100 T  0054:000011
                    BEGIN  DS:=3 CHR; DS:=LIT"(";  END                                                       05325110 T  0054:000112
                ELSE BEGIN                                                                                   05325120 T  0054:000211
                    DS:=5 CHR; DS:=LIT"("; SI:=LOC CELL; SI:=SI+5;                                           05325130 T  0054:000211
                    IF SC≥"6" THEN DS:=2 LIT"F-" ELSE DS:=2 LIT"F+";                                         05325140 T  0054:000410
                    COUNT:=DI; DI:=LOC CELL; DI:=DI+4;                                                       05325150 T  0054:000513
                    DS:=11 RESET; DI:=COUNT;                                                                 05325160 T  0054:000611
                    END;                                                                                     05325170 T  0054:000712
                SI:=LOC CELL; SI:=SI+4; TALLY:=4; % LOCATION.                                                05325180 T  0054:000712
                3(IF SC="0" THEN % DONT PRINT LEADING ZEROES.                                                05325190 T  0054:000713
                    BEGIN SI:=SI+1; TALLY:=TALLY+63  END ELSE JUMP OUT);                                     05325200 T  0054:000811
                COUNT:=TALLY; DS:=COUNT CHR; TALLY:=0; COUNT:=TALLY;                                         05325210 T  0054:001112
                DS:=4 LIT") = "; CELL:=DI; % SAVE OUR PLACE.                                                 05325220 T  0054:001210
                CI:=CI+N;                                                                                    05325230 T  0054:001312
                GO M0;                                                                                       05325240 T  0054:001313
                GO M1;                                                                                       05325250 T  0054:001410
                GO M2;                                                                                       05325260 T  0054:001410
                GO M3;                                                                                       05325270 T  0054:001411
                GO M4;                                                                                       05325280 T  0054:001411
                GO M5;                                                                                       05325290 T  0054:001512
                GO M6;                                                                                       05325300 T  0054:001512
                GO M7;                                                                                       05325310 T  0054:001513
M0;             SI:=ID; SI:=SI+2; DI:=LOC COUNT;                                                             05325320 T  0054:001513
                DI:=DI+7; DS:=CHR; DI:=CELL;  DS:=COUNT CHR;                                                 05325330 T  0054:001611
```

```
                    GO XIT;
               M1:  DI:=CELL; DS:=19 LIT"*TEMPORARY STORAGE*"; GO XIT;
               M2:  DI:=CELL;
                      DS:=36 LIT"*LIST, LABEL, OR SEGMENT DESCRIPTOR*"; GO XIT;
               M3:  DI:=CELL; DS:=27 LIT"*CASE STATEMENT DESCRIPTOR*"; GO XIT;
               M4:  DI:=CELL; DS:=19 LIT"*FORMAT DESCRIPTOR*"; GO XIT;
PRT(751) = *LIST, LABEL, OR SEGMENT DESCRIPTOR*
               M5:  DI:=CELL; DS:=24 LIT"*OUTER BLOCK DESCRIPTOR*"; GO XIT;
PRT(752) = *LIST, LABEL, OR SEGMENT DESCRIPTOR*
               M6:  DI:=CELL; DS:=20 LIT"*SEGMENT DESCRIPTOR*"; GO XIT;
PRT(753) = *LIST, LABEL, OR SEGMENT DESCRIPTOR*
               M7:  DI:=CELL; DS:=18 LIT"*LABEL DESCRIPTOR*";
PRT(754) = *LIST, LABEL, OR SEGMENT DESCRIPTOR*
               XIT:
                    END FILLIT;
PRT(755) = *LIST, LABEL, OR SEGMENT DESCRIPTOR*
```

```
                BLANKFT(14,LIN);
                IF N=1 THEN FILLIT(LIN,PORS,GS,0,ACCUM[1])
           ELSE IF N>1 THEN FILLIT(LIN,PORS,GS,0,INFO[N,LINKR,N,LINKC])
           ELSE FILLIT(LIN,PORS,GS,ABS(N),N);
               IF NOHEADING THEN DATIME; WRITELINE;
               END WRITEPRT;
```

```
                                                                  54 IS   80 LONG, NEXT SEG    3
```

```
                COMMENT  GETSPACE MAKES ASSIGNMENTS TO VARIABLES AND DESCRIPTORS IN
                         THE STACK AND PRT.  PERMANENT TELLS WHETHER IT IS A
                         PERMANENTLY ASSIGNED CELL (ALWAYS IN PRT) OR NOT.  NON
                         PERMANENT CELLS ARE EITHER IN STACK OR PRT ACCORDING TO
                         MODE.  CARE IS TAKEN TO REUSE NON PERMANENT PRT CELLS;
           INTEGER PROCEDURE GETSPACE(PERMANENT,L); VALUE PERMANENT,L;
                                                    BOOLEAN PERMANENT; INTEGER L;
                   BEGIN LABEL L1,L2,EXIT;
                                                                  START OF SEGMENT **********   55
                     BOOLEAN STREAM PROCEDURE MASK(K); VALUE K;
PRT(756) = MASK
                        BEGIN DI←LOC MASK; DI←DI+2; SKIP K DB; DS←SET END MASK;
```

```
                     BOOLEAN M,Q;
STACK(F+3) = M
STACK(F+4) = Q
                     INTEGER ROW,COL,GS;
STACK(F+5) = ROW
STACK(F+6) = COL
STACK(F+7) = GS
                     IF PERMANENT
                       THEN BEGIN
                        IF PRTIMAX>1022 THEN FLAG(148);%
```

```
            SPRT[GS+PRTIMAX,[38:5]] ← MASK(PRTIMAX,[43:5]-35)                    05348000 T  0055:0005:3
                OR SPRT[GS];                                                     05349000 T  0055:0009:2
            PRTIMAX ← (GS ← PRTIMAX)+1 END                                       05350000 T  0055:0010:1
         ELSE IF MODE = 0 THEN BEGIN                                             05351000 T  0055:0012:1
            Q ← SPRT[ROW ← PRTI,[38:5]];                                         05352000 T  0055:0014:0
            M ← MASK(COL ← PRTI,[43:5]-35);                                      05353000 T  0055:0016:0
            COL ← COL+35;                                                        05354000 T  0055:0019:3
L1:         IF REAL(M AND Q) ≠ 0                                                 05355000 T  0055:0020:1
              THEN BEGIN                                                         05356000 T  0055:0021:3
                 IF REAL(BOOLEAN(GS←4294967296-REAL(M)) AND Q) = GS               05357000 T  0055:0022:1
                   THEN BEGIN                                                    05358000 T  0055:0024:1
                      COL ← 0; M ← TRUE;                                         05359000 T  0055:0025:3
                      IF ROW ← ROW+1 > 31                                        05360000 T  0055:0027:2
                         THEN BEGIN FLAG(148); GS ← PRTIMAX;                     05361000 T  0055:0028:0
                            GO TO L2 END;                                        05362000 T  0055:0030:1
                      Q ← SPRT[ROW];                                             05363000 T  0055:0033:2
                      GO TO L1 END;                                              05364000 T  0055:0034:0
                 COL ← COL+1; M ← BOOLEAN(REAL(M)+REAL(M));                      05365000 T  0055:0034:1
                 GO TO L1 END;                                                   05366000 T  0055:0037:2
            PRTI ← (GS ← 32×ROW+COL)+1;                                          05367000 T  0055:0037:3
            IF PRTI > PRTIMAX THEN PRTIMAX ← PRTI END                            05368000 T  0055:0040:0
         ELSE BEGIN                                                              05369000 T  0055:0042:0
            IF STACKCTR > 767 THEN FLAG(149);                                    05370000 T  0055:0042:1
            STACKCTR ← (GS ← STACKCTR)+1; Q ← FALSE;                             05371000 T  0055:0044:1
            GO TO EXIT END;                                                      05372000 T  0055:0047:2
L2:      IF GS ≥ 512 THEN GS ← GS+1024;                                          05373000 T  0055:0047:3
         Q ← TRUE;                                                               05374000 T  0055:0050:1
EXIT:    GETSPACE ← GS;                                                          05375000 T  0055:0051:2
         IF GS > 1023 THEN GS ← GS-1024;                                         05376000 T  0055:0052:1
IF PRTOG THEN WRITEPRT(IF Q THEN "PRT   " ELSE "STACK",L,B2D(GS));               05376100 T  0055:0055:2
     END GETSPACE;                                                               05378000 T  0055:0059:3
```

55 IS 67 LONG. NEXT SEG 3

```
            COMMENT ARRAYCHECK CHECKS A PARAMETER-INFO WORD FOR SORT/MERGE;       05379000 T  0003:0671:3
         BOOLEAN PROCEDURE ARRAYCHECK(AAW); VALUE AAW; REAL AAW;                 05380000 T  0003:0671:3
PRT(757) = ARRAYCHECK
            ARRAYCHECK←AAW.CLASS<BOOARRAYID OR AAW.CLASS>INTARRAYID              05381000 T  0003:0671:3
                                    OR AAW.INCR ≠1;                             05382000 T  0003:0674:0


            COMMENT COMMACHECK LOOKS FOR COMMAS AND STEPS AROUND THEM;          05383000 T  0003:0679:2
         BOOLEAN PROCEDURE COMMACHECK;                                          05384000 T  0003:0679:2
PRT(760) = COMMACHECK
            BEGIN IF NOT(COMMACHECK←(STEPI=COMMA)) THEN ERR(350);               05385000 T  0003:0679:2
                  STEPIT                                                        05386000 T  0003:0683:2
            END  COMMACHECK;                                                    05387000 T  0003:0683:2


            COMMENT   HVCHECK CHECKS VALIDITY OF HIVALU PROCEDURE FOR SORT;     05388000 T  0003:0685:2
         BOOLEAN PROCEDURE HVCHECK(ELBW); VALUE ELBW; REAL ELBW;                05389000 T  0003:0685:2
```

```
PRT(761) = HVCHECK
                    IF ELBW.CLASS≠PROCID THEN ERR(356) ELSE                          05390000 T  00031068:2
                    IF BOOLEAN(ELBW.FORMAL) THEN HVCHECK←TRUE  ELSE                  05390100 T  00031068:1
                    IF TAKE(GT1←GIT(ELBW))≠1 THEN ERR(357) ELSE                      05391000 T  00031069:2
                    IF ARRAYCHECK(TAKE(GT1←1)) THEN ERR(358) ELSE                    05392000 T  00031069:2
                         HVCHECK←TRUE;                                               05393000 T  00031069:1


                    COMMENT   OUTPROCHECK CHECKS SORT/MERGE OUTPUT PROCEDURE;        05394000 T  00031070:0
                  BOOLEAN PROCEDURE OUTPROCHECK(ELBW); VALUE ELBW; REAL ELBW;        05395000 T  00031070:0
PRT(762) = OUTPROCHECK
                    IF ELBW.CLASS≠PROCID THEN ERR(351) ELSE                          05396000 T  00031070:0
                    IF BOOLEAN(ELBW.FORMAL) THEN OUTPROCHECK←TRUE  ELSE              05396100 T  00031070:3
                    IF TAKE(GT1←GIT(ELBW))≠2 THEN ERR(352) ELSE                      05397000 T  00031070:0
                    IF TAKE(GT1+1).CLASS≠BOOID THEN ERR(353) ELSE                    05398000 T  00031071:0
                    IF ARRAYCHECK(TAKE(GT1←2)) THEN ERR(354) ELSE                    05399000 T  00031071:0
                         OUTPROCHECK←TRUE;                                           05400000 T  00031071:3


                    COMMENT   EQLESCHECK CHECKS THE COMPARE ROUTINE FOR SORT/MERGE;  05401000 T  00031072:2
                  BOOLEAN PROCEDURE EQLESCHECK(ELBW); VALUE ELBW; REAL ELBW;         05402000 T  00031072:2
PRT(763) = EQLESCHECK
                    IF ELBW.CLASS≠BOOPROCID THEN ERR(359) ELSE                       05403000 T  00031072:2
                    IF BOOLEAN (ELBW.FORMAL) THEN EQLESCHECK ← TRUE ELSE             05403100 T  00031072:1
                    IF TAKE(GT1←GIT(ELBW))≠2 THEN ERR(360) ELSE                      05404000 T  00031072:2
                    IF ARRAYCHECK(TAKE(GT1←1)) THEN ERR(361) ELSE                    05405000 T  00031073:2
                    IF ARRAYCHECK(TAKE(GT1←2)) THEN ERR(362) ELSE                    05406000 T  00031073:1
                         EQLESCHECK←TRUE;                                            05407000 T  00031074:0


               COMMENT      ROUTINES IN THIS SECTION COMPILE CODE FOR ALL EXPRESSIONS; 06000000 T  00031074:2
               COMMENT   AEXP IS THE ARITHMETIC EXPRESSION ROUTINE;                   06001000 T  00031074:2
               PROCEDURE AEXP;                                                        06002000 T  00031074:2
                         BEGIN                                                        06003000 T  00031074:2
                              IF ELCLASS = IFV                                        06004000 T  00031074:2
                                THEN BEGIN IF IFEXP ≠ ATYPE THEN ERR(102) END         06005000 T  00031074:0
                                ELSE BEGIN ARITHSEC; SIMPARITH END                    06006000 T  00031074:3
                         END AEXP;                                                    06007000 T  00031074:2


               COMMENT ARITHSEC COMPILES FIRST PRIMARY IN AN ARITHMETIC EXPRESSION.   06008000 T  00031074:2
                         IN PARTICULAR IT HANDLES P, +P, -P, AND -P*Q WHERE P         06009000 T  00031074:2
                         AND Q ARE PRIMARIES;                                         06010000 T  00031074:2
               PROCEDURE ARITHSEC;                                                    06011000 T  00031074:2
                         BEGIN                                                        06012000 T  00031074:2
                           IF ELCLASS = ADOP                                          06013000 T  00031074:2
                              THEN BEGIN                                              06014000 T  00031075:0
                                STEPIT;                                               06015000 T  00031075:2
                                IF ELBAT[I-1].ADDRESS = ADD THEN PRIMARY              06016000 T  00031075:3
```

```
                ELSE BEGIN                                                          06017000 T  0003:075:0
                  PRIMARY;                                                          06018000 T  0003:075:2
              WHILE ELCLASS = FACTOP DO                                      %WF    06019000 T  0003:075:3
                  BEGIN STEPIT;  PRIMARY;  EMITUP END;                       %WF    06020000 T  0003:075:2
                        ENDTOG ← LINKTOG; EMITO(CHS);                               06021000 T  0003:075:2
                        LINKTOG ← ENDTOG; ENDTOG ← FALSE END END                    06022000 T  0003:076:1
                  ELSE PRIMARY END ARITHSEC;                                        06023000 T  0003:076:0


COMMENT SIMPARITH COMPILES SIMPLE ARITHMETIC EXPRESSIONS ON THE                     06024000 T  0003:076:3
            ASSUMPTION THAT AN ARITHMETIC PRIMARY HAS ALREADY BEEN                  06025000 T  0003:076:3
            COMPILED.  IT ALSO HANDLES THE CASE OF A CONCATENATE                    06026000 T  0003:076:3
            WHERE ACTUALPARAPART CAUSED THE VARIABLE ROUTINE TO                     06027000 T  0003:076:3
            COMPILE ONLY PART OF A PRIMARY.  MOST OF THE WORK OF                    06028000 T  0003:076:3
            SIMPARITH IS DONE BY ARITHCOMP. AN ARTIFIAL ROUTINE                     06029000 T  0003:076:3
            WHICH DOES THE HIERARCHY ANALYSIS USING RECURSION.                      06030000 T  0003:076:3
            ARITHCOMP IS A SUBROUTINE ONLY TO GET THIS RECURSION;                   06031000 T  0003:076:3
PROCEDURE SIMPARITH;                                                                06032000 T  0003:076:3
            BEGIN                                                                   06033000 T  0003:076:3
              WHILE ELCLASS = AMPERSAND                                             06034000 T  0003:076:3
                 DO BEGIN STEPIT; PRIMARY; PARSE END;                               06035000 T  0003:076:4
              WHILE ELCLASS≥ADOP AND ELCLASS≤FACTOP DO ARITHCOMP END;               06036000 T  0003:076:2


                  COMMENT ARITHCOMP IS THE GUTS OF THE ARITHMETIC EXPRESSION ROUTINE    06037000 T  0003:077:3
                            ANALYSIS.  IT CALLS PRIMARY AT APPROPRIATE TIMES AND        06038000 T  0003:077:3
                            EMITS THE ARITHMETIC OPERATORS.  THE HIERARCHY ANALYSIS     06039000 T  0003:077:3
                            IS OBTAINED BY RECURSION;                                   06040000 T  0003:077:3
                  PROCEDURE ARITHCOMP;                                                  06041000 T  0003:077:3
                            BEGIN INTEGER OPERATOR, OPCLASS;                            06042000 T  0003:077:3
                                                                                  START OF SEGMENT **********   56
STACK(F+2) = OPERATOR
STACK(F+3) = OPCLASS
                             DO BEGIN                                                   06043000 T  0056:000:0
                              OPERATOR ← 1 & ELBAT[I] [36:17:10];                       06044000 T  0056:000:0
                     COMMENT THIS SETS UP THE OPERATOR WHICH WILL BE EMITTED.  THE HIGH  06045000 T  0056:000:2
                            ORDER TEN BITS OF THE OPERATOR ARE LOCATED IN [17:10]       06046000 T  0056:000:2
                            OF THE ELBAT WORD;                                          06047000 T  0056:000:2
                              OPCLASS ← ELCLASS;                                        06048000 T  0056:000:2
                              STEPIT; PRIMARY;                                          06049000 T  0056:000:1
                              IF OPCLASS = FACTOP THEN EMITUP                           06050000 T  0056:000:3
                                 ELSE BEGIN                                             06051000 T  0056:000:2
                                   WHILE OPCLASS < ELCLASS DO ARITHCOMP;                06052000 T  0056:000:6
                     COMMENT THE CLASSES ARE ARRANGED IN ORDER OF HIERARCHY;            06053000 T  0056:000:8
                                   STACKCT ← 1;                                 %A     06053500 T  0056:000:8
                                   EMIT(OPERATOR)  END                                  06054000 T  0056:000:9
                             END UNTIL OPCLASS ≠ ELCLASS END ARITHCOMP;                  06055000 T  0056:000:9
                                                                                  56 IS    14 LONG, NEXT SEG     3


                  COMMENT  IMPFUN HANDLES ALL OF THE SPECIAL FUNCTIONS;                 06056000 T  0003:077:3
                  PROCEDURE IMPFUN;                                                     06057000 T  0003:077:3
```

```
                    BEGIN                                                                    06058000 T  0003:0771:3
                         REAL T1,T2,T3 ;                                                     06059000 T  0003:0771:3
                                                                                  START OF SEGMENT **********   57
STACK(F+2) = T1
STACK(F+3) = T2
STACK(F+4) = T3
                         BOOLEAN B ;                                                         06059050 T  0057:0000:0
STACK(F+5) = B
                         DEFINE ERRX(ERRX1)=BEGIN T1←ERRX1; GO ERROR END #;                  06059100 T  0057:0000:0
                         LABEL ABS, SIGN, ENTIER, TIME, STATUS,%                             06060000 T  0057:0000:0
                              MAXANDMIN, DELAY, OTHERS, EXIT;%                               06060100 T  0057:0000:0
                         LABEL ERROR,L1,L2,L3 ;                                              06060110 T  0057:0000:0
                         SWITCH S ← OTHERS, ABS, SIGN, ENTIER, TIME, STATUS,%                06061000 T  0057:0000:0
                                   MAXANDMIN, MAXANDMIN, DELAY;%                             06061100 T  0057:0002:1
                         DEFINE MAXV = 6#;%                                                  06061200 T  0057:0007:3
                         IF T2←(T1←ELBAT[I]).[27:6]<9 THEN GO S[T2+1] ;                      06062000 T  0057:0007:3
                         IF T2≠25 THEN EMITO(MKS) ;                                          06062110 T  0057:0013:3
                         IF STEPI≠LEFTPAREN THEN ERRX(105); STEPIT ;                         06062120 T  0057:0015:3
                         IF T2<24 THEN                                                       06062125 T  0057:0018:1
                              BEGIN                                                          06062130 T  0057:0019:3
L3:                           IF TABLE(I+1)=COMMA THEN                                      06062135 T  0057:0020:0
                                   IF ELCLASS>BOOID AND ELCLASS<BOOARRAYID THEN           06062140 T  0057:0021:3
                                        BEGIN                                            06062145 T  0057:0024:0
                                        CHECKER(T3←ELBAT[I]); STEPIT; STEPIT ;           06062150 T  0057:0024:1
                                        AEXP; EMITV(T3,ADDRESS); GO L1 ;                 06062155 T  0057:0027:2
                                        END ;                                            06062160 T  0057:0029:2
L2:                           AEXP; IF ELCLASS≠COMMA THEN ERRX(80); STEPIT ;             06062165 T  0057:0029:2
                              AEXP; IF T2<24 THEN EMITO(XCH) ;                           06062170 T  0057:0033:3
                              END                                                        06062175 T  0057:0036:0
                         ELSE BEGIN                                                      06062180 T  0057:0036:0
                              IF T2=24 THEN GO L2; EMITL(0); AEXP ;                      06062185 T  0057:0036:1
                              IF ELCLASS≠COMMA THEN ERRX(80); EMITPAIR(9,SND) ;           06062190 T  0057:0039:2
                              EMITD(1,1,8); STEPIT; AEXP ;                               06062195 T  0057:0042:1
                              END ;                                                      06062200 T  0057:0044:1
L1:                      IF T2<23 THEN BEGIN IF ELCLASS≠COMMA THEN ERRX(80) END          06062210 T  0057:0044:1
                         ELSE IF ELCLASS≠RTPAREN THEN ERRX(104) ;                        06062220 T  0057:0048:1
                         STEPIT ;                                                        06062230 T  0057:0051:3
                         IF T2<21 OR B THEN                                              06062280 T  0057:0052:0
                              BEGIN EMITV(GNAT(T1)) ;                                    06062285 T  0057:0053:3
                              B←ELCLASS=INTID OR ELCLASS=INTARRAYID                      06062290 T  0057:0055:2
                                             OR ELCLASS=INTPROCID ;                      06062295 T  0057:0056:0
                              IF ELCLASS≥REALID AND ELCLASS≤INTID THEN                   06062340 T  0057:0058:1
                                   BEGIN EMITN(ELBAT[I],ADDRESS); STEPIT END             06062350 T  0057:0060:0
                              ELSE IF ELCLASS<BOOID AND ELCLASS>BOOPROCID THEN           06062370 T  0057:0062:1
                                        IF ELBAT[I].LINK≠PROINFO.LINK THEN FLAG(211)     06062380 T  0057:0065:2
                                        ELSE BEGIN EMITL(514); STEPIT END                06062385 T  0057:0068:0
                                   ELSE IF ELCLASS≤LABELID AND ELCLASS>BOOARRAYID       06062390 T  0057:0070:1
                                             THEN VARIABLE(FL)                           06062400 T  0057:0072:0
                                        ELSE ERRX(185) ;                                 06062420 T  0057:0073:3
                              IF ELCLASS≠RTPAREN THEN ERRX(104); STEPIT ;                 06062430 T  0057:0075:3
                              EMITO(IF B THEN ISD ELSE STD); EMITV(17); GO EXIT ;         06062435 T  0057:0078:1
                              END ;                                                      06062440 T  0057:0082:0
                         IF T2<23 THEN                                                   06062470 T  0057:0082:0
                              BEGIN % DMOD, DARCTAN2                                     06062480 T  0057:0083:2
                              B←TRUE; GO L3 ;                                            06062500 T  0057:0083:3
                              END ;                                                      06062535 T  0057:0084:1
                         IF T2<25 THEN BEGIN EMITV(GNAT(T1)); GO EXIT END ;              06062540 T  0057:0084:1
```

```
                EMITD(9,47,1); EMITV(9); EMITO(ADD); GO EXIT ;                    06062560 T  0057:008713
ERROR:          ERR(T1); GO EXIT ;                                                06062565 T  0057:009112
OTHERS:         EMITO(MKS) ;                                                      06064000 T  0057:009210
                PANA;                                                             06065000 T  0057:009313
                EMITV(GNAT(T1)); GO TO EXIT;                                      06066000 T  0057:009410
ABS:            PANA; EMITO(SSP); GO TO EXIT;                                     06067000 T  0057:009610
SIGN:           PANA;                                                             06068000 T  0057:009713
                EMITO(DUP); EMITL(0); EMITO(NEQ); EMITO(XCH);                     06069000 T  0057:009811
                EMITD(1,1,1); GO TO EXIT;                                         06070000 T  0057:010113
ENTIER:         PANA; EMITNUM(.5); EMITO(SUB);                                    06071000 T  0057:010312
                EMITPAIR(JUNK,ISN); GO TO EXIT;                                   06072000 T  0057:010610
MAXANDMIN:%                                                                       06072010 T  0057:010912
                IF STEPI≠LEFTPAREN THEN ERR(105) ELSE%                            06072030 T  0057:010912
                BEGIN STEPIT; AEXP;%                                              06072040 T  0057:011112
                     WHILE ELCLASS=COMMA DO%                                      06072050 T  0057:011211
                     BEGIN STEPIT; EMITO(DUP); AEXP;%                             06072060 T  0057:011410
                          EMITPAIR(JUNK, SND);%                                   06072070 T  0057:011610
                          IF T2=MAXV THEN EMITO(LSS) ELSE EMITO(GTR) ;            06072080 T  0057:011712
                          EMITPAIR(2, BFC); EMITO(DEL); EMITV(JUNK);              06072090 T  0057:012010
                     END;%                                                        06072100 T  0057:012211
                     IF ELCLASS≠RTPAREN THEN ERR(104) ELSE STEPIT;%               06072110 T  0057:012312
                END;%                                                             06072120 T  0057:012610
                GO TO EXIT;%                                                      06072130 T  0057:012610
DELAY:          IF STEPI≠LEFTPAREN THEN%                                          06072200 T  0057:012611
                BEGIN ERR(105); GO TO EXIT END;%                                  06072210 T  0057:012810
                STEPIT; AEXP; IF ELCLASS≠COMMA THEN%                              06072220 T  0057:012913
                BEGIN ERR(165); GO TO EXIT END;%                                  06072230 T  0057:013113
                STEPIT; AEXP; IF ELCLASS≠COMMA THEN%                              06072240 T  0057:013312
                BEGIN ERR(165); GO TO EXIT END;%                                  06072250 T  0057:013512
                STEPIT; AEXP; IF ELCLASS≠RTPAREN THEN%                            06072260 T  0057:013611
                 BEGIN ERR(104); GO TO EXIT END ELSE STEPIT;%                     06072270 T  0057:013811
                EMITPAIR(31, COM); EMITO(DEL); EMITO(DEL);%                       06072280 T  0057:014112
                GO TO EXIT;%                                                      06072290 T  0057:014313
TIME:           PANA; EMITL(1); EMITO(COM);                                       06073000 T  0057:014410
                GO TO EXIT;                                                       06073100 T  0057:014712
STATUS:         IF STEPI≠LEFTPAREN THEN BEGIN ERR(105); GO TO EXIT END;           06073200 T  0057:014713
                IF STEPI=SUPERFILEID OR ELCLASS=FILEID THEN                       06073250 T  0057:015011
                BEGIN EMIT(16); EMIT(0); EMIT(0); PASSFILE;                       06073300 T  0057:015211
                     EMITPAIR(32, COM); T1←3;                                     06073350 T  0057:015610
                END ELSE BEGIN EMIT(4); EMIT(0); T1←0;                            06073400 T  0057:015713
                     IF ELCLASS≥BOOARRAYID AND ELCLASS≤INTARRAYID THEN            06073450 T  0057:016011
                       BEGIN T1←FI; VARIABLE(T1); END ELSE AEXP;                  06073500 T  0057:016210
                     IF T1=FI THEN                                                06073550 T  0057:016512
                     BEGIN EMITPAIR(0, XCH); EMITPAIR(32, COM); T1←3              06073600 T  0057:016610
                     END ELSE BEGIN IF ELCLASS=RTPAREN THEN                       06073650 T  0057:016811
                          BEGIN EMIT(0); EMITPAIR(32, COM); T1←3 END              06073700 T  0057:017011
                          ELSE BEGIN EMITO(XCH); EMITO(DEL);                      06073750 T  0057:017313
                               EMITO(XCH); EMITO(DEL);                            06073800 T  0057:017513
                               IF ELCLASS≠COMMA THEN                              06073810 T  0057:017712
                               BEGIN ERR(129); GO TO EXIT END;                    06073820 T  0057:017713
                               STEPIT; AEXP; EMITPAIR(28,COM); T1←1;              06073830 T  0057:017913
                END; END; END;                                                    06073840 T  0057:018210
                GTI1←0;                                                           06073845 T  0057:018210
                DO EMITO(DEL) UNTIL GTI1←GTI1+1=T1;%                              06073850 T  0057:018312
                IF ELCLASS≠RTPAREN THEN ERR(104) ELSE STEPIT;                     06073860 T  0057:018610
                GO TO EXIT;                                                       06073870 T  0057:018912
EXIT; END IMPFUN;                                                                 06074000 T  0057:018913
```

57 IS 193 LONG, NEXT SEG 3

```
COMMENT PRIMARY COMPILES ARITHMETIC PRIMARIES.  IT HANDLES MOST CASES
        OF THE CONCATENATE AND SOME CASES OF THE PARTIAL WORD
        DESIGNATORS, ALTHOUGH VARIABLE HANDLES THE MORE COMMON
        CASES;
PROCEDURE PRIMARY;
        BEGIN
        LABEL
                                                                    START OF SEGMENT **********   58
                L11, L12, L13, L14, L15, L16, L17, L18, L19,
            L20, L21, L22, L23, L24, L25, L26, L27, L28, L29,
            L30, L31, L32, L33, L34, L35;
        SWITCH S :=
                L11, L12, L13, L14, L15, L16, L17, L18, L19,
            L20, L21, L22, L23, L24, L25, L26, L27, L28, L29,
            L30, L31, L32, L33, L34, L35;
  COMMENT  LN IS THE LABEL FOR THE CLASS N;
        LABEL EXIT,RP,LDOT,LAMPER;
        GO TO S[ELCLASS-PROCID];  COMMENT  GO TO PROPER SYNTAXER;
        IF ELCLASS = UNKNOWNID THEN ERR(100);
        IF ELCLASS=FILEID OR ELCLASS=SUPERFILEID THEN
            BEGIN
            IF FILEATTRIBUTEHANDLER(FP)#ATYPE THEN FLAG(294) ;
            GO TO LAMPER ;
            END ;
L12: L13: L17: L21: L25: L29: L30:
 COMMENT NO PRIMARY MAY BEGIN WITH QUANTITIES WITH THESE CLASSES;
        IF REL AND ELCLASS = BOOARRAYID THEN GO L22;
        ERR(103); GO TO EXIT;
L11:
 COMMENT INTRINSIC FUNCTIONS;
        IMPFUN; STACKCT := STACKCT-1; GO TO LDOT;                   %A
L14: L15: L16:
 COMMENT STREAM PROCEDURE FUNCTION DESIGNATORS;
 IF ARRAYFLAG THEN CHECKBOUNDLVL;
        STRMPROCSTMT; GO TO LDOT;
L18: L19: L20:
 COMMENT ORDINARY FUNCTION DESIGNATORS;
 IF ARRAYFLAG THEN CHECKBOUNDLVL;
        PROCSTMT(FALSE); GO TO LDOT;
L22: L23: L24: L26: L27: L28:
 COMMENT VARIABLES, SIMPLE AND SUBSCRIPTED;
 IF ARRAYFLAG THEN CHECKBOUNDLVL;
        VARIABLE(FP); GO TO LAMPER;
L32:
 COMMENT LITERALS - I.E. INTEGERS BETWEEN 0 AND 1023;
        EMIT(0&ELBAT[I] [36:17:10]); STEPIT;GO TO LAMPER;
L31: L33:
 COMMENT STRINGS AND NONLITERALS;
        EMITNUM(C); STEPIT; GO TO LAMPER;
L35:
 COMMENT   COULD BE REAL TRANSFER FUNCTION,  IF IT IS COMPILE BOOLEAN
        EXPRESSION - OTHERWISE AN ERROR;
        IF ELBAT[I].ADDRESS = REALV THEN BEGIN
```

```
06075000 T  0003:077:3
06076000 T  0003:077:3
06077000 T  0003:077:3
06078000 T  0003:077:3
06079000 T  0003:077:3
06080000 T  0003:077:3
06081000 T  0003:077:3
06082000 T  0058:000:0
06083000 T  0058:000:0
06084000 T  0058:000:0
06085000 T  0058:000:0
06086000 T  0058:002:1
06087000 T  0058:002:1
06088000 T  0058:002:1
06089000 T  0058:001:3
06090000 T  0058:001:3
06091000 T  0058:001:3
06092000 T  0058:001:1
06092005 T  0058:002:1
06092010 T  0058:002:0
06092015 T  0058:002:1
06092020 T  0058:002:2
06092025 T  0058:002:3
06093000 T  0058:002:3
06094000 T  0058:002:0
06094950 T  0058:002:0
06095000 T  0058:002:0
06096000 T  0058:002:2
06097000 T  0058:003:0
06098000 T  0058:003:0
06099000 T  0058:003:0
06100000 T  0058:003:2
06100100 T  0058:003:2
06101000 T  0058:003:0
06102000 T  0058:003:2
06103000 T  0058:003:0
06103100 T  0058:003:0
06104000 T  0058:003:2
06105000 T  0058:003:1
06106000 T  0058:003:2
06106100 T  0058:003:2
06107000 T  0058:004:0
06108000 T  0058:004:3
06109000 T  0058:004:2
06110000 T  0058:004:2
06111000 T  0058:004:2
06112000 T  0058:004:2
06113000 T  0058:004:2
06114000 T  0058:004:1
06115000 T  0058:004:2
06116000 T  0058:004:2
06117000 T  0058:004:2
```

```
                    IF STEPI ≠ LEFTPAREN                                                    06118000 T  0058:0049:2
                    THEN BEGIN ERR(105); GO TO EXIT END;                                    06119000 T  0058:0049:3
                  STEPIT; BEXP; GO TO RP END;                                               06120000 T  0058:0051:3
                IF ELBAT[I].ADDRESS = INTV THEN                                             06120100 T  0058:0053:2
                   BEGIN PANA; EMITPAIR(JUNK,ISN); GO TO LDOT END;                          06120200 T  0058:0054:1
                ERR(106); GO TO EXIT;                                                       06121000 T  0058:0057:2
L34:                                                                                        06122000 T  0058:0058:1
 COMMENT (;                                                                                 06123000 T  0058:0059:2
                STEPIT; AEXP;                                                               06124000 T  0058:0059:2
                    STACKCT ← STACKCT-1;                                          %A        06124500 T  0058:0060:0
RP:             IF ELCLASS ≠ RTPAREN THEN BEGIN ERR(104); GO EXIT END;                      06125000 T  0058:0061:2
                STEPIT;                                                                     06126000 T  0058:0064:1
LDOT:           DOT;  COMMENT THIS CHECKS FOR PARTIAL WORDS;                                06127000 T  0058:0065:2
LAMPER:         STACKCT ← STACKCT+1;                                          %A            06128000 T  0058:0066:1
                WHILE ELCLASS = AMPERSAND                                     %A            06128500 T  0058:0068:0
                  DO BEGIN STEPIT; PRIMARY; PARSE END;                                      06129000 T  0058:0069:2
 COMMENT THIS CODE HANDLES CANCATENATES;                                                    06130000 T  0058:0072:0
EXIT:           END PRIMARY;                                                                06131000 T  0058:0072:0
                                                                                  58 IS   74 LONG, NEXT SEG    3



COMMENT BEXP IS THE BOOLEAN EXPRESSION ROUTINE;                                             06132000 T  0003:0771:3
PROCEDURE BEXP; IF EXPRSS ≠ BTYPE THEN ERR(107);                                            06133000 T  0003:0771:3

                COMMENT EXPRSS IS A GENERAL EXPRESSION ROUTINE CAPABLE OF COMPILING         06134000 T  0003:0774:1
                        ANY GIVEN TYPE OF EXPRESSION.  IT REPORTS ON ITS ACTION             06135000 T  0003:0774:1
                        BY GIVING AS A RESULT EITHER ATYPE,BTYPE, OR DTYPE                  06136000 T  0003:0774:1
                        DEPENDING ON WHETHER IT COMPILED AN ARITHMETIC, BOOLEAN,            06137000 T  0003:0774:1
                        OR DESIGNATIONAL EXPRESSION;                                        06138000 T  0003:0774:1
                INTEGER PROCEDURE EXPRSS;                                                   06139000 T  0003:0774:1
                        BEGIN                                                               06140000 T  0003:0774:1
                          IF ELCLASS = IFV                                                  06141000 T  0003:0774:1
                            THEN BEGIN                                                      06142000 T  0003:0775:2
                              IF EXPRSS ← IFEXP = ATYPE                                     06143000 T  0003:0776:0
                                THEN IF ELCLASS = RELOP THEN ERR(108) END                   06144000 T  0003:0777:2
                            ELSE IF EXPRSS ← BOOSEC = BTYPE THEN SIMPBOO                    06145000 T  0003:0780:0
                        END EXPRSS;                                                         06146000 T  0003:0782:1



                COMMENT BOOSEC COMPILES EITHER A BOOLEAN SECONDARY OR AN ARITHMETIC         06147000 T  0003:0785:2
                        EXPRESSION OR A DESIGNATIONAL EXPRESSION.  IT REPORTS               06148000 T  0003:0785:2
                        AS EXPRSS REPORTS;                                                  06149000 T  0003:0785:2
                INTEGER PROCEDURE BOOSEC;                                                   06150000 T  0003:0785:2
                        BEGIN BOOLEAN N;                                                    06151000 T  0003:0785:2
                                                                                 START OF SEGMENT **********   59
STACK(F+3) ← N
                          IF N ← ELCLASS = NOTOP THEN STEPIT;                               06152000 T  0059:0000:0
                          GT4 ← BOOSEC ← BOOPRIM;                                           06153000 T  0059:0002:0
                          IF N THEN BEGIN EMITLNG; EMIT(0); L ← L-1;                        06154000 T  0059:0003:3
                   COMMENT  THE LAST LINE IS PREPARATORY.  LATER ROUTINES USE THE           06155000 T  0059:0007:2
```

```
                RESULTS HERE TO ELIMINATE PAIRS OF LNGS;                                  06156000 T  0059:0007:2
                IF GT4 ≠ BTYPE THEN ERR(109)                                              06157000 T  0059:0007:2
      COMMENT  AN ARITHMETIC OR DESIGNATIONAL EXPRESSION MAY NOT BE                       06158000 T  0059:0008:1
               LOGICALLY NEGATED;                                                         06159000 T  0059:0008:1
                END END BOOSEC;                                                           06160000 T  0059:0008:1
                                                                                59 IS    12 LONG, NEXT SEG    3


COMMENT SIMPBOO COMPILES SIMPLE BOOLEAN EXPRESSIONS ON THE ASSUMPTION             06161000 T  0003:0785:2
               THAT A BOOLEAN PRIMARY HAS ALREADY BEEN COMPILED.  IT              06162000 T  0003:0785:2
               ALSO HANDLES THE CASE OF A CONCATENATE WHERE ACTUALPARA-            06163000 T  0003:0785:2
               PART CAUSED THE VARIABLE ROUTINE TO COMPILE ONLY PART OF            06164000 T  0003:0785:2
               A PRIMARY.  MOST OF THE WORK OF SIMPBOO IS DONE BY BOO-             06165000 T  0003:0785:2
               COMP, AN ARTIFIAL ROUTINE WHICH DOES THE HIERARCHY ANA-             06166000 T  0003:0785:2
               LYSIS USING RECURSION;                                             06167000 T  0003:0785:2
PROCEDURE SIMPBOO;                                                                06168000 T  0003:0785:2
               BEGIN                                                              06169000 T  0003:0785:2
                 WHILE ELCLASS = AMPERSAND                                        06170000 T  0003:0785:2
                    DO BEGIN                                                      06171000 T  0003:0786:0
                      STEPIT;                                                     06172000 T  0003:0787:2
                      IF BOOPRIM≠ BTYPE THEN ERR(109);                            06173000 T  0003:0787:3
                      PARSE END;                                                  06174000 T  0003:0790:0
                 WHILE ELCLASS ≥ EQVOP AND ELCLASS ≤ ANDOP DO BOOCOMP              06175000 T  0003:0791:2
               END BOOCOMP;                                                       06176000 T  0003:0793:2

                    COMMENT BOOCOMP IS THE GUTS OF THE BOOLEAN EXPRESSION ROUTINE ANALYSIS,   06177000 T  0003:0794:1
                            IT CALLS BOOSEC AT APPROPRIATE TIMES AND EMITS THE BOOLEAN        06178000 T  0003:0794:1
                            OPERATORS.  THE HIERARCHY ANALYSIS IS OBTAINED BY RECUR-          06179000 T  0003:0794:1
                            SION;                                                             06180000 T  0003:0794:1
                    PROCEDURE BOOCOMP;                                                        06181000 T  0003:0794:1
                            BEGIN INTEGER OPCLASS, OPERATOR; LABEL EXIT;                      06182000 T  0003:0794:1
                                                                                  START OF SEGMENT **********   60
STACK(F+2) = OPCLASS
STACK(F+3) = OPERATOR
                              DO BEGIN                                                        06183000 T  0060:0000:0
                                OPERATOR ← 1 & ELBAT[I] [36:17:10];                           06184000 T  0060:0000:0
                         COMMENT THIS SETS UP THE OPERATOR WHICH WILL BE EMITTED.  THE HIGH    06185000 T  0060:0002:0
                                 ORDER TEN BITS OF THE OPERATOR ARE LOCATED IN [17:10]         06186000 T  0060:0002:0
                                 OF THE ELBAT WORD;                                           06187000 T  0060:0002:0
                                     OPCLASS ← ELCLASS;                                       06188000 T  0060:0002:0
                                     STEPIT;                                                  06189000 T  0060:0002:1
                                     IF BOOSEC ≠ BTYPE                                        06190000 T  0060:0003:2
                                       THEN BEGIN ERR(109); GO TO EXIT END;                   06191000 T  0060:0003:3
                                     WHILE OPCLASS < ELCLASS                                  06192000 T  0060:0006:0
                                       DO IF ELCLASS ≤ ANDOP THEN BOOCOMP                     06193000 T  0060:0006:0
                                          ELSE BEGIN ERR(110); GO TO EXIT END;                06194000 T  0060:0008:1
                         COMMENT THE CLASSES ARE ARRANGED IN ORDER OF HIERARCHY;              06195000 T  0060:0011:2
                                 STACKCT ← 1;                                          %A     06195500 T  0060:0011:2
                                 IF OPCLASS = IMPOP                                           06196000 T  0060:0012:0
                                   THEN BEGIN                                                 06197000 T  0060:0012:0
                         COMMENT SINCE IMP IS NOT IN THE MACHINE REPETOIRE WE MUST CONSTRUCT   06198000 T  0060:0013:2
                                 ONE.  NOTICE THAT WE USE EMITLNG IN ONE SPOT TO OBTAIN        06199000 T  0060:0013:2
```

```
                    THE CANCELING OF POSSIBLE MULTIBLE LNGS.  ALSO THE 0          06200000 T  0060:0013:2
                    EMITTED PROVIDES THE POSSIBILITY OF DOING THIS IN THE         06201000 T  0060:0013:2
                    FUTURE. (SEE CODE FOR EMITLNG);                               06202000 T  0060:0013:2
                            EMITLNG;                                              06203000 T  0060:0013:2
                            EMITO(LND);                                           06204000 T  0060:0013:3
                            EMITO(LNG);                                           06205000 T  0060:0014:1
                            EMIT(0);                                              06206000 T  0060:0015:2
                            L ← L-1 END                                           06207000 T  0060:0016:0
                        ELSE EMIT(OPERATOR)                                       06208000 T  0060:0017:2
                    END UNTIL OPCLASS ≠ ELCLASS;                                  06209000 T  0060:0018:0
EXIT:               END BOOCOMP;                                                  06210000 T  0060:0019:3
                                                                    60 IS   23 LONG. NEXT SEG    3


                COMMENT BOOPRIM COMPILES BOOLEAN PRIMARIES, AND ARITHMETIC OR     06211000 T  0003:0794:1
                        DESIGNATIONAL EXPRESSIONS.  IT REPORTS AS EXPRSS REPORTS; 06212000 T  0003:0794:1
                INTEGER PROCEDURE BOOPRIM;                                        06213000 T  0003:0794:1
                        BEGIN INTEGER TYPE;                                       06214000 T  0003:0794:1
                                                                  START OF SEGMENT **********   61
STACK(F+3) = TYPE
                        LABEL  L9,                                                06215000 T  0061:0000:0
                               L10, L11, L12, L13, L14, L15, L16, L17, L18, L19,  06216000 T  0061:0000:0
                               L20, L21, L22, L23, L24, L25, L26, L27, L28, L29,  06217000 T  0061:0000:0
                               L30, L31, L32, L33, L34, L35;                      06218000 T  0061:0000:0
                        SWITCH S ← L9,                                            06219000 T  0061:0000:0
                               L10, L11, L12, L13, L14, L15, L16, L17, L18, L19,  06220000 T  0061:0002:1
                               L20, L21, L22, L23, L24, L25, L26, L27, L28, L29,  06221000 T  0061:0002:1
                               L30, L31, L32, L33, L34, L35;                      06222000 T  0061:0002:1
 COMMENT LN IS THE LABEL FOR THE CLASS N;                                         06223000 T  0061:0016:1
                LABEL EXIT,LE,D,TD,T;                                             06224000 T  0061:0016:1
                LABEL FAH ;                                                       06224500 T  0061:0016:1
                GO TO S[ELCLASS-SUPERFILEID];                                     06225000 T  0061:0016:1
                IF ELCLASS = ADOP THEN GO TO L11;                                 06226000 T  0061:0019:3
                IF ELCLASS = UNKNOWNID THEN ERR(100);                             06227000 T  0061:0020:1
                IF ELCLASS=FILEID OR ELCLASS=SUPERFILEID THEN                     06227500 T  0061:0022:1
                     BEGIN                                                        06227510 T  0061:0024:1
                     BOOPRIM←TYPE←FILEATTRIBUTEHANDLER(FP) ;                      06227520 T  0061:0025:2
                     GO FAH ;                                                     06227530 T  0061:0026:1
                     END ;                                                        06227540 T  0061:0027:2
LE: L10: L12:                                                                     06228000 T  0061:0027:2
 COMMENT NO BOOLEAN PRIMARY, ARITHMETIC EXPRESSION, OR DESIGNATIONAL              06229000 T  0061:0028:0
                EXPRESSION MAY BEGIN WITH QUANTITIES WITH THESE CLASSES;          06230000 T  0061:0028:0
                ERR(111); GO TO EXIT;                                             06231000 T  0061:0028:0
L35:            IF GT1 ← ELBAT[T].ADDRESS = BOOV                                  06232000 T  0061:0029:2
                  THEN BEGIN PANA; GO TO TD END;                                  06233000 T  0061:0031:3
                IF GT1 ≠ REALV THEN BEGIN ERR(112); GO TO EXIT END;               06234000 T  0061:0033:3
L11: L14: L15: L16: L18: L19: L20: L22: L23: L24: L26: L27: L28:                  06235000 T  0061:0036:0
     L31: L32: L33:                                                               06236000 T  0061:0036:0
 COMMENT ARITHMETIC TYPE STUFF;                                                   06237000 T  0061:0036:0
                AEXP;                                                             06238000 T  0061:0036:0
D:              IF ELCLASS ≠ RELOP THEN BEGIN BOOPRIM ← ATYPE;GO EXIT END;        06239000 T  0061:0036:1
                RELATION;                                                         06240000 T  0061:0039:3
                BOOPRIM ← BTYPE; GO TO EXIT;                                      06241000 T  0061:0040:0
L13:                                                                              06242000 T  0061:0041:2
 COMMENT BOOLEAN STREAM PROCEDURE DESIGNATOR;                                     06243000 T  0061:0042:0
 IF ARRAYFLAG THEN CHECKBOUNDLVL;                                                 06243100 T  0061:0042:0
```

```
            STRMPROCSTMT; GO TO TD;                                                  06244000 T  0061:0043:2
L17:                                                                                 06245000 T  0061:0044:0
 COMMENT BOOLEAN PROCEDURE DESIGNATOR;                                               06246000 T  0061:0045:2
      IF ARRAYFLAG THEN CHECKBOUNDLVL;                                               06246100 T  0061:0045:2
            PROCSTMT(FALSE); GO TO TD;                                               06247000 T  0061:0046:0
L21: L25:                                                                            06248000 T  0061:0047:3
 COMMENT BOOLEAN VARIABLES;                                                          06249000 T  0061:0048:0
 IF ARRAYFLAG THEN CHECKBOUNDLVL;                                                    06249100 T  0061:0048:0
            VARIABLE(FP); GO TO T;                                                   06250000 T  0061:0049:2
L9: L29:                                                                             06251000 T  0061:0050:1
 COMMENT LABELS AND SWITCHES;                                                        06252000 T  0061:0051:2
            DEXP; BOOPRIM ← DTYPE; GO TO EXIT;                                       06253000 T  0061:0051:2
L30:                                                                                 06254000 T  0061:0052:1
 COMMENT TRUE OR FALSE;                                                              06255000 T  0061:0053:2
            EMIT(0&ELBAT[1][45:26:1]); STEPIT; GO TO T;                              06256000 T  0061:0053:2
L34:                                                                                 06257000 T  0061:0056:0
 COMMENT (;                                                                          06258000 T  0061:0056:0
            STEPIT; TYPE ← BOOPRIM ← EXPRSS;                                         06259000 T  0061:0056:0
 COMMENT COMPILE THE EXPRESSION, WHATEVER IT IS;                                     06260000 T  0061:0058:0
            STACKCT ← STACKCT-1;                                          %A         06260500 T  0061:0058:0
            IF ELCLASS ≠ RTPAREN THEN BEGIN ERR(104); GO TO EXIT END;                06261000 T  0061:0059:2
            STEPIT;                                                                  06262000 T  0061:0061:3
FAH:                                                                                 06262500 T  0061:0062:0
            IF TYPE = DTYPE THEN GO TO EXIT;                                         06263000 T  0061:0063:2
 COMMENT FINISHED IF EXPRESSION COMPILED WAS DESIGNATIONAL;                          06264000 T  0061:0064:0
            IF TYPE = BTYPE THEN BEGIN                                               06265000 T  0061:0064:0
TD:         DOT; COMMENT HANDLES PARTIAL WORDS;                                      06266000 T  0061:0065:3
T:          STACKCT ← STACKCT+1;                                              %A     06267000 T  0061:0067:3
            WHILE ELCLASS = AMPERSAND DO                                      %A     06267500 T  0061:0069:2
               COMMENT HANDLES CONCATENATE;                                          06268000 T  0061:0071:2
                  BEGIN                                                              06269000 T  0061:0071:2
                     STEPIT;                                                         06270000 T  0061:0071:2
                     IF BOOPRIM ≠ BTYPE                                              06271000 T  0061:0071:3
                          THEN BEGIN ERR(109); GO TO EXIT END;                       06272000 T  0061:0072:0
                     PARSE END;                                                      06273000 T  0061:0074:1
                  BOOPRIM ← BTYPE; GO TO EXIT END;                                   06274000 T  0061:0075:3
            COMMENT IF NOT BOOLEAN OR DESIGNATIONAL, MUST COMPLETE ARITHMETIC         06275000 T  0061:0076:1
                  EXPRESSION;                                                        06276000 T  0061:0076:1
                  DOT; SIMPARITH; GO TO D;                                           06277000 T  0061:0076:1
 EXIT: END BOOPRIM;                                                                  06278000 T  0061:0079:3
                                                                                61 IS    83 LONG, NEXT SEG    3


            COMMENT RELATION COMPILES RELATIONS.  IT ASSUMES THAT THE LEFTHAND       06279000 T  0003:0794:1
                  EXPRESSION HAS ALREADY BEEN COMPILED;                              06280000 T  0003:0794:1
            PROCEDURE RELATION;                                                      06281000 T  0003:0794:1
               BEGIN                                                                 06282000 T  0003:0794:1
               INTEGER OPERATOR;                                                     06282200 T  0003:0794:1
                                                                          START OF SEGMENT **********   62
STACK(F+2) = OPERATOR                                                                06282400 T  0062:0000:0
            REAL A;
STACK(F+3) = A                                                                       06282600 T  0062:0000:0
            BOOLEAN SIGNA,CONSTANA,SIMPLE,MANY,SIGN;
STACK(F+4) = SIGNA
STACK(F+5) = CONSTANA
```

```
STACK(F+6) = SIMPLE
STACK(F+7) = MANY
STACK(F+10) = SIGN
                    DEFINE FORMALNAME = [9:2]=2#;                                   06282800 T  0062:0000:0
                    PROCEDURE PLUG(C,A,S); VALUE C,A,S; BOOLEAN  C,S; REAL A;       06283000 T  0062:0000:0
PRT(764) = PLUG
                         BEGIN                                                      06283200 T  0062:0000:0
                         IF C THEN EMITNUM(A)                                       06283400 T  0062:0000:0
                    ELSE      BEGIN CHECKER(A);  EMITV(A,ADDRESS) END;              06283600 T  0062:0001:2
                         IF S THEN EMITO(CHS);                                      06283800 T  0062:0004:0
                         END PLUG;                                                  06284000 T  0062:0005:3



                    DO   BEGIN                                                      06284200 T  0062:0005:3
                         OPERATOR:=1&ELBAT[I][36:17:10];                            06284400 T  0062:0006:0
                COMMENT  SET UP CODE FOR RELATIONAL OPERATOR TO BE                  06284600 T  0062:0008:0
                         EMITTED LATER (AFTER PROCESSING SECOND HALF),              06284800 T  0062:0008:0
                         THE HIGH-ORDER BITS OF THE BINARY OPERATOR                 06285000 T  0062:0008:0
                         ARE TAKEN FROM THE [17:10] FIELD OF THE                    06285200 T  0062:0008:0
                         ELBAT WORD FOR THE RELATIONAL SYMBOL;                      06285400 T  0062:0008:0
                         IF MANY THEN                                               06285600 T  0062:0008:0
                           IF SIMPLE THEN PLUG(CONSTANA,A,SIGNA) ELSE EMITV(JUNK);  06285800 T  0062:0008:0
                         SIGNA:=FALSE;                                              06286000 T  0062:0012:0
                         IF STEPI=ADOP THEN SIGNA:=ELBAT[I].ADDRESS=SUB;            06286200 T  0062:0012:1
                         IF SIGN:=ELCLASS=ADOP THEN STEPIT;                         06286400 T  0062:0016:0
                         CONSTANA:=ELCLASS≥NONLITNO AND ELCLASS≤STRNGCON;           06286600 T  0062:0018:1
                         A:=REAL(ELCLASS≥REALID AND ELCLASS≤INTID                   06286800 T  0062:0020:1
                              AND NOT ELBAT[I].FORMALNAME);                         06287000 T  0062:0021:3
                         SIMPLE:=(CONSTANA OR BOOLEAN(A)) AND STEPI=RELOP;          06287200 T  0062:0024:1
                         IF SIMPLE THEN                                             06287400 T  0062:0027:2
                              BEGIN                                                 06287600 T  0062:0027:3
                              IF CONSTANA THEN A:=C ELSE A:=ELBAT[I-1];             06287800 T  0062:0028:0
                              PLUG(CONSTANA,A,SIGNA)                                06288000 T  0062:0031:3
                              END                                                   06288200 T  0062:0032:0
                         ELSE BEGIN                                                 06288400 T  0062:0032:1
                              I:=I-REAL(SIGN)-2; STEPIT; AEXP;                      06288600 T  0062:0033:2
                              IF ELCLASS=RELOP THEN EMITPAIR(JUNK,SND);             06288800 T  0062:0036:0
                              END;                                                  06289000 T  0062:0038:0
                         STACKCT:=1;  EMIT(OPERATOR);                               06289200 T  0062:0038:0
                         IF MANY THEN EMITO(LND)                                    06289400 T  0062:0039:3
                    ELSE      BEGIN EMIT(0);  L:=L-1 END;                           06289600 T  0062:0040:1
                         MANY:=TRUE;                                                06289800 T  0062:0043:3
                         END UNTIL ELCLASS≠RELOP                                    06290000 T  0062:0044:1
                    END RELATION;                                                   06290200 T  0062:0044:1
                                                                              62 IS   49 LONG, NEXT SEG    3



                COMMENT IFEXP COMPILES CONDITIONAL EXPRESSIONS.  IT REPORTS THE TYPE  06292000 T  0003:0794:1
                         OF EXPRESSIONS AS EXPRSS REPORTS;                          06293000 T  0003:0794:1
                INTEGER PROCEDURE IFEXP;                                            06294000 T  0003:0794:1
                         BEGIN INTEGER TYPE,THENBRANCH,ELSEBRANCH;                  06295000 T  0003:0794:1
                                                                        START OF SEGMENT **********   63
STACK(F+3) = TYPE
STACK(F+4) = THENBRANCH
```

```
STACK(F+5) = ELSEBRANCH
                        IFCLAUSE;                                                    06296000 T  0063:000:0
                        STACKCT ← 0;                                          %A     06296500 T  0063:000:1
                        THENBRANCH ← BUMPL;                                          06297000 T  0063:001:2
                COMMENT SAVE L FOR LATER FIXUP;                                      06298000 T  0063:000:2
                        IFEXP ← TYPE ← EXPRSS; COMMENT COMPILE 1ST EXPRSS;           06299000 T  0063:000:2
                        STACKCT ← 0;                                          %A     06299500 T  0063:000:1
                        ELSEBRANCH ← BUMPL;                                          06300000 T  0063:000:2
                        EMITB(BFC,THENBRANCH,L);                                     06301000 T  0063:000:7
                        IF ELCLASS ≠ ELSEV THEN ERR(155) ELSE BEGIN                  06302000 T  0063:000:8
                        STEPI;                                                       06303000 T  0063:001:1
                        IF TYPE = ATYPE THEN AEXP ELSE                               06304000 T  0063:001:2
                        IF TYPE = DTYPE THEN DEXP ELSE BEXP;                         06305000 T  0063:001:2
                        STACKCT ← 1;                                          %A     06305500 T  0063:001:0
                        COMMENT THIS COMPILES PROPER TYPE SECOND EXPRSS;              06306000 T  0063:001:2
                        EMITB(BFW,ELSEBRANCH,L);                                     06307000 T  0063:001:2
                        EMIT(1); L ← L-1;                                            06308000 T  0063:001:0
                COMMENT THIS IS USED BY EMITLNG TO CLEANUP CODE,  COMPARE WITH       06309000 T  0063:002:0
                        BOOSEC, BOOCOMP, AND RELATION;                               06310000 T  0063:002:0
                        END END IFEXP;                                               06311000 T  0063:002:0
                                                                                  63 IS   23 LONG, NEXT SEG    3


        PROCEDURE PARSE ;%COMPILES CODE FOR THE CONCATENATE ;                        06312000 T  0003:079:1
        BEGIN   INTEGER FIRST,SECOND,THIRD;                                          06312500 T  0003:079:1
                                                                                START OF SEGMENT **********   64
STACK(F+2) = FIRST
STACK(F+3) = SECOND
STACK(F+4) = THIRD
                BOOLEAN P1,P2,P3;                                                    06313000 T  0064:000:0

STACK(F+5) = P1
STACK(F+6) = P2
STACK(F+7) = P3

                LABEL L1,L2,L3,SKIP1,SKIP2,EXIT;                                     06313500 T  0064:000:0
                IF ELCLASS = FIELDID THEN                             %117-          06313550 C  0064:000:0
                  BEGIN                                               %117-          06313600 C  0064:000:1
                    FIRST := ELBAT[I].SBITF;                          %117-          06313650 C  0064:000:2
                    SECOND := 48 - (THIRD := ELBAT[I].NBITF);         %117-          06313700 C  0064:000:2
                    GO TO SKIP1;                                      %117-          06313750 C  0064:000:2
                  END                                                 %117-          06313800 C  0064:000:3
                ELSE                                                  %117-          06313850 C  0064:000:3
                IF ELCLASS ≠ LFTBRKET THEN BEGIN ERR(90);GO TO EXIT END;             06314000 T  0064:000:3
                IF STEPI = FIELDID THEN                               %117-          06314050 C  0064:000:8
                  BEGIN                                               %117-          06314100 C  0064:000:9
                    FIRST := ELBAT[I].SBITF;                          %117-          06314150 C  0064:001:0
                    SECOND := 48 - (THIRD := ELBAT[I].NBITF);         %117-          06314200 C  0064:001:3
                    IF STEPI ≠ RTBRKET THEN                           %117-          06314250 C  0064:001:0
                      BEGIN                                           %117-          06314300 C  0064:001:2
                        ERR(94);                                      %117-          06314350 C  0064:001:3
                        GO TO EXIT;                                   %117-          06314400 C  0064:001:1
                      END;                                            %117-          06314450 C  0064:001:2
                    GO TO SKIP1;                                      %117-          06314500 P  0064:001:2
                  END                                                 %117-          06314550 C  0064:001:3
                ELSE                                                  %117-          06314600 C  0064:001:3
                  IF ELCLASS ≠ LITNO THEN % PREPARE FOR DYNAMIC DIAL  %117-          06314650 C  0064:001:3
```

```
          GO TO L1;                                                                    %117-
     FIRST ← C;
     IF TABLE(I+1) = COLON THEN
BEGIN
     STEPIT;
     IF FIRST ≤0 THEN FLAG(92);
END ELSE
BEGIN
L1:  EMITO(MKS);
     AEXP;
     P1 ← TRUE;
     IF ELCLASS ≠ COLON THEN BEGIN ERR(91);GO TO EXIT END;
END;
     IF STEPI ≠ LITNO THEN GO TO L2;

     SECOND ← C ;
     IF GT1 ← TABLE(I+1) = COLON THEN
BEGIN
     STEPIT;
     IF SECOND ≤0 THEN FLAG(092);
END ELSE
BEGIN
     IF GT1 = RTBRKET THEN
BEGIN
     STEPIT;
     SECOND ← 48 - (THIRD ← SECOND);
     GO TO SKIP2;
END;
L2:  IF NOT P1 THEN BEGIN EMITO(MKS);EMITL(FIRST) END;
     AEXP;
     P1 ← P2 ← TRUE;
     IF ELCLASS = COLON THEN


     ELSE
     IF ELCLASS = RTBRKET THEN
BEGIN
     EMITO(DUP);
     EMITL(48) ;EMITO(SUB);
     EMITO(CHS);EMITO(XCH);
     P3 ← TRUE;
     GO TO SKIP1;
END ELSE BEGIN ERR(91);GO TO EXIT END;
END;
     IF STEPI ≠ LITNO THEN GO L3 ;
     THIRD ← C;
     IF TABLE(I+1) = RTBRKET THEN
BEGIN
     STEPIT;
SKIP2: IF THIRD ≤0 OR THIRD > 47 THEN FLAG(95);
END ELSE
BEGIN
L3:  IF NOT P2 THEN
BEGIN
     IF NOT P1 THEN BEGIN EMITO(MKS);EMITL(FIRST) END;
     EMITL(SECOND);
END;
     AEXP;
```

```
06314700 C  00641001811
06315000 T  00641001912
06315500 T  00641002010
06316000 T  00641002113
06316500 T  00641002210
06317000 T  00641002211
06317500 T  00641002411
06318000 T  00641002411
06318500 T  00641002512
06319000 T  00641002611
06319500 T  00641002712
06320000 T  00641002810
06320500 T  00641003011
06321000 T  00641003011
06321100 T  00641003210
06321500 T  00641003210
06322000 T  00641003211
06322500 T  00641003512
06323000 T  00641003513
06323500 T  00641003610
06324000 T  00641003810
06324500 T  00641003810
06325000 T  00641003811
06325500 T  00641003912
06326000 T  00641003913
06326500 T  00641004010
06327000 T  00641004210
06327500 T  00641004211
06328000 T  00641004211
06328500 T  00641004513
06329000 T  00641004610
06329100 T  00641004712
06329200 T  00641004810
06329300 T  00641004810
06329350 T  00641004810
06329400 T  00641004811
06329450 T  00641004913
06329500 T  00641005010
06329550 T  00641005112
06329600 T  00641005211
06329700 T  00641005410
06329800 T  00641005411
06329900 T  00641005512
06330000 T  00641005712
06330500 T  00641005712
06330600 T  00641005811
06330700 T  00641005912
06330800 T  00641006112
06331000 T  00641006113
06331100 T  00641006210
06331200 T  00641006512
06331300 T  00641006512
06331500 T  00641006513
06331600 T  00641006611
06331700 T  00641006712
06331800 T  00641006913
06332000 T  00641007010
06332100 T  00641007010
```

```
          P1← P2←P3 ←TRUE;                                                                    06332200 T  0064:0070:1
          IF ELCLASS ≠ RTBRKET THEN BEGIN ERR(94);GO TO EXIT END;                             06332300 T  0064:0072:1
     END;                                                                                     06332400 T  0064:0075:2
     SKIP1: IF P1 THEN                                                                        06332500 T  0064:0075:2
     BEGIN                                                                                    06333000 T  0064:0075:2
          IF NOT P2 THEN EMITL(SECOND);                                                       06333500 T  0064:0075:3
          IF NOT P3 THEN                                                                      06334000 T  0064:0077:3
     BEGIN                                                                                    06334100 T  0064:0078:0
          EMITL(THIRD);EMITL(1);                                                              06334200 T  0064:0078:1
          EMITV(GNAT(DIALER));                                                                06334500 T  0064:0080:0
          EMIT(TRB & THIRD[36:42:6]);                                                         06334600 T  0064:0081:2
     END ELSE                                                                                 06334700 T  0064:0083:2
     BEGIN                                                                                    06335000 T  0064:0083:2
          EMITL(0);                                                                           06335100 T  0064:0083:3
          EMITV(GNAT(DIALER));                                                                06335200 T  0064:0084:0
          EMITO(DEL);                                                                         06335500 T  0064:0085:3
     END;                                                                                     06335700 T  0064:0086:0
     END ELSE                                                                                 06336000 T  0064:0086:0
     BEGIN                                                                                    06336100 T  0064:0086:0
          IF FIRST + THIRD > 48 OR SECOND + THIRD > 48 THEN FLAG(095);                        06336200 T  0064:0086:1
          EMITD(SECOND,FIRST,THIRD);                                                          06336300 T  0064:0090:1
     END;                                                                                     06336400 T  0064:0092:0
          STEPIT;                                                                             06336500 T  0064:0092:0
     EXIT: STACKCT ← 1;                                                                       06336600 T  0064:0092:1
END PARSE;                                                                                    06336700 T  0064:0093:3
                                                                                        64 IS   97 LONG, NEXT SEG    3
```

```
          COMMENT  DOT COMPILES CODE FOR PARTIAL WORD DESIGNATORS,EXCEPT FOR                  06337000 T  0003:0794:1
                   THOSE CASES HANDLED BY THE VARIABLE ROUTINE ;                              06337100 T  0003:0794:1
          PROCEDURE DOTIT;                                                                    06338000 T  0003:0794:1
                   BEGIN INTEGER FIRST,SECOND; LABEL EXIT;                                    06339000 T  0003:0794:1
                                                                                        START OF SEGMENT **********   65
STACK(F+2) = FIRST
STACK(F+3) = SECOND
                                                                                              06340000 T  0065:0000:0
                        IF DOTSYNTAX(FIRST,SECOND) THEN GO TO EXIT;                           06341000 T  0065:0000:0
                                                                                    %A        06342000 T  0065:0001:3
                        EMITI(0,FIRST,SECOND);                                      %A        06343000 T  0065:0001:3
                                                                                    %A        06344000 T  0065:0003:2
                        STEPIT;                                                               06345000 T  0065:0003:2
          EXIT: END DOTIT;                                                                    06346000 T  0065:0003:3
                                                                                        65 IS    7 LONG, NEXT SEG    3
```

```
          COMMENT GENGO CONSTRUCTS THE CALL ON AN INTRINSIC PROCEDURE WHICH                   06347000 T  0003:0794:1
                  PREPARES A LABEL DESCRIPTOR FOR THE MCP.  THE MCP EXPECTS                   06348000 T  0003:0794:1
                  THE F-REGISTER AND THE BLOCKCTR TO BE IN THIS DESCRIPTOR,                   06349000 T  0003:0794:1
                  SO THAT STORAGE CAN BE PROPERLY RETURNED.  THE BLOCKCTR                     06350000 T  0003:0794:1
                  IS AN OBJECT TIME COUNTER IN A FIXED CELL IN THE PRT.  IT                   06351000 T  0003:0794:1
                  IS INCREMENTED AND DECREMENTED AT ENTRY AND EXIT FROM                       06352000 T  0003:0794:1
                  BLOCKS,IF NECESSARY.  THE CODE TO DO THIS IS COMPILED BY                    06353000 T  0003:0794:1
                  THE BLOCK ROUTINE.  IN A PROCEDURE, THE BLOCKCTR AT ENTRY                   06354000 T  0003:0794:1
```

```
                    IS ALSO STORED IN F+1;
                PROCEDURE GENGO(ELBATWORD); VALUE ELBATWORD; REAL ELBATWORD;
                    BEGIN INTEGER TLEVEL;

STACK(F+2) = TLEVEL
                      EMITO(MKS);
                      IF TLEVEL + ELBATWORD.LVL > JUMPCTR THEN
                        JUMPCTR + TLEVEL;
                  COMMENT JUMPCTR IS USED BY THE BLOCK ROUTINE TO THINK ABOUT
                      INCREMENTING AND DECREMENTING THE BLOCKCTR.  HERE WE TELL
                      BLOCK ROUTINE ABOUT THE LEVEL TO WHICH OUR BAD GO TO IS
                      JUMPING;
                        IF TLEVEL < FRSTLEVEL OR MODE = 0
                          THEN BEGIN
                  COMMENT OUR BAD GO TO IS JUMPING OUTSIDE OF ALL PROCEDURES;
                            EMIT(0);
                            EMITL(TLEVEL) END
                          ELSE BEGIN
                            EMITN(512);
                            EMITV(513); COMMENT PICK UP BLOCKCTR AT ENTRY
                                               FROM F+1;
                            IF TLEVEL + TLEVEL - SUBLEVEL -1 ≠ 0
                              THEN BEGIN
                                EMITL(TLEVEL);
                                EMITO(ADD)  COMMENT IF JUMP IS NOT TO SAME LEVEL
                                            AS AT ENTRY TIME, FUDGE THE COUNTER;
                              END END;
                        EMITV(GNAT(GOTOSOLVER)) COMMENT CALL THE INTRINSIC;
                    END GENGO;

                  COMMENT DEXP COMPILES DESIGNATIONAL EXPRESSIONS.  FOR THE MOST PART
                      IT ASSUMES THAT A COMMUNICATE IS GOING TO BE USED AGAINST
                      THE LABEL DESCRIPTOR IN ORDER TO OBTAIN GO TO ACTION,
                      STORAGE RETURN, AND STACK CUT BACK.  HOWEVER IF IT NEVER
                      SETS GOTOG TO TRUE THEN THE LABELS ARE ALL LOCAL AND NO
                      COMMUNICATE WILL BE DONE;
                PROCEDURE DEXP;
                      BEGIN
                        LABEL EXIT;

                        BOOLEAN S,F;
STACK(F+2) = S
STACK(F+3) = F

                        REAL ELBW;
STACK(F+4) = ELBW
                          IF (S + ELCLASS = SWITCHID) OR ELCLASS = LABELID
                            THEN BEGIN
                              CHECKER(ELBW + ELBAT[I]);
                              SCATTERELBAT;
                              IF LEVEL ≠ LEVELF OR F + FORMALF THEN GOTOG + TRUE;
                              IF FAULTOG THEN
                                   IF S OR F THEN FAULTLEVEL+1 ELSE
                                   IF FAULTLEVEL>LEVELF THEN FAULTLEVEL+LEVELF;
```

```
                                                                  06355000 T  0003:079#:1
                                                                  06356000 T  0003:079#:1
                                                                  06357000 T  0003:079#:1
                                                          START OF SEGMENT **********   66
                                                                  06358000 T  0066:0000:0
                                                                  06359000 T  0066:0000:1
                                                                  06360000 T  0066:0002:1
                                                                  06361000 T  0066:0003:3
                                                                  06362000 T  0066:0003:3
                                                                  06363000 T  0066:0003:3
                                                                  06364000 T  0066:0003:3
                                                                  06365000 T  0066:0003:3
                                                                  06366000 T  0066:0004:1
                                                                  06367000 T  0066:0006:0
                                                                  06368000 T  0066:0006:0
                                                                  06369000 T  0066:0006:1
                                                                  06370000 T  0066:0007:3
                                                                  06371000 T  0066:0008:0
                                                                  06372000 T  0066:0008:1
                                                                  06373000 T  0066:0009:3
                                                                  06374000 T  0066:0009:3
                                                                  06375000 T  0066:0011:2
                                                                  06376000 T  0066:0012:0
                                                                  06377000 T  0066:0013:2
                                                                  06378000 T  0066:0013:3
                                                                  06379000 T  0066:0013:3
                                                                  06380000 T  0066:0013:3
                                                                  06381000 T  0066:0014:1
                                                          66 IS   18 LONG, NEXT SEG    3
                                                                  06382000 T  0003:079#:1
                                                                  06383000 T  0003:079#:1
                                                                  06384000 T  0003:079#:1
                                                                  06385000 T  0003:079#:1
                                                                  06386000 T  0003:079#:1
                                                                  06387000 T  0003:079#:1
                                                                  06388000 T  0003:079#:1
                                                                  06389000 T  0003:079#:1
                                                                  06390000 T  0003:079#:1
                                                          START OF SEGMENT **********   67
                                                                  06391000 T  0067:0000:0

                                                                  06392000 T  0067:0000:0

                                                                  06393000 T  0067:0000:0
                                                                  06394000 T  0067:0001:3
                                                                  06395000 T  0067:0002:1
                                                                  06396000 T  0067:0004:0
                                                                  06397000 T  0067:0004:1
                                                                  06397100 T  0067:0007:3
                                                                  06397200 T  0067:0008:0
                                                                  06397300 T  0067:0010:1
```

```
                IF S THEN BEGIN                                                  06398000 T  0067100131:2
                  BANA; EMITPAIR(JUNK,ISD);                                      06399000 T  0067100133:3
                  EMITV(GNAT(ELBW));                                             06400000 T  0067100152:2
                  IF F THEN GO TO EXIT; END                                      06401000 T  0067100161:1
                ELSE BEGIN                                                       06402000 T  0067100173:3
                  STEPIT;                                                        06403000 T  0067100180:0
                  IF F THEN BEGIN EMITV(ADDRSF); GO TO EXIT END;                 06404000 T  0067100181:1
                  EMITL(GNAT(ELBW)) END;                                         06405000 T  0067100201:1
                GENGO(ELBW);                                                     06406000 T  0067100213:3
              END ELSE IF EXPRSS # DTYPE THEN ERR(115);                          06407000 T  0067100221:1
EXIT;     END DEXP;                                                              06408000 T  0067100252:2
                                                                        67 IS   29 LONG, NEXT SEG    3


PROCEDURE IFCLAUSE;                                                              06409000 T  0003107941:1
          BEGIN STEPIT; STACKCT + 0; BEXP;                          %A           06410000 T  0003107941:1
            IF ELCLASS # THENV THEN ERR(116)ELSE STEPIT END IFCLAUS;             06411000 T  0003107961:1
COMMENT  PANA COMPILES THE CONSTRUCT: (<ARIT. EXP.>);                            06412000 T  0003107993:3


PROCEDURE PANA;                                                                  06413000 T  0003108001:0
          BEGIN                                                                  06414000 T  0003108001:0
           IF STEPI # LEFTPAREN THEN ERR(105)                                    06415000 T  0003108001:0
           ELSE BEGIN STEPIT; AEXP; IF ELCLASS # RTPAREN THEN                    06416000 T  0003108013:3
               ERR(104) ELSE STEPIT END END PANA;                                06417000 T  0003108041:1


          COMMENT  BANA COMPILES THE CONSTRUCT:  [<ARITH. EXP.>];                06418000 T  0003108072:2
          PROCEDURE BANA;                                                        06419000 T  0003108072:2
                    BEGIN                                                        06420000 T  0003108072:2
                     IF STEPI # LFTBRKET THEN ERR(117)                           06421000 T  0003108072:2
                         ELSE BEGIN STEPIT; AEXP; IF ELCLASS # RTBRKET THEN      06422000 T  0003108081:1
                           ERR(118) ELSE STEPIT END END BANA      ;              06423000 T  0003108113:3


          PROCEDURE MAKEALABEL;                                                  06500000 T  0003108141:0
PRT(765) = MAKEALABEL
                    BEGIN LABEL EXIT; REAL I;                                    06501000 T  0003108141:0
                                                                  START OF SEGMENT **********   68
STACK(F+2) = I
                    STREAMTOG+FALSE;                                             06502000 T  0068100001:0
                    EMITO(MKS); PASSFILE;                                        06503000 T  0068100001:1
                    IF ELCLASS#WITHV THEN                                        06504000 T  0068100021:0
                         BEGIN ERR(301); GO TO EXIT END;                         06505000 T  0068100021:1
                    FOR I+1 STEP 1 UNTIL 6 DO                                    06506000 T  0068100041:1
                         BEGIN IF STEPI=FACTOP THEN                              06507000 T  0068100061:0
                                   BEGIN EMIT(4); EMITO(CHS); STEPIT END         06508000 T  0068100072:2
                              ELSE AEXP;                                         06509000 T  0068100093:3
                              IF ELCLASS#COMMA THEN GO TO EXIT;                  06510000 T  0068100101:1
```

```
                        END;                                                                    06511000 T  0068:0011:3
                EXIT:  FOR I:=I STEP 1 UNTIL 5 DO                                               06512000 T  0068:0014:0
                BEGIN EMIT(4);EMITO(CHS) END;                                                   06512100 T  0068:0015:2
                EMITL(11);                                                                      06512200 T  0068:0018:1
                        EMITV(5);                                                               06513000 T  0068:0019:3
                    END;                                                                        06514000 T  0068:0020:0
                                                                                          68 IS   23 LONG, NEXT SEG    3


                COMMENT THIS SECTION CONTAINS THE STATEMENT ROUTINES;                           07000000 T  0003:0814:0
                COMMENT COMPOUNDTAIL COMPILES COMPOUNDTAILS.  IT ALSO ELIMINATES                07001000 T  0003:0814:0
                        COMMENTS FOLLOWING ENDS.  AFTER ANY ERROR, ERROR MESSAGES               07002000 T  0003:0814:0
                        ARE SUPPRESSED.  COMPOUNDTAIL IS PARTIALLY RESPONSIBLE                  07003000 T  0003:0814:0
                        FOR RESTORING THE ABILITY TO WRITE ERROR MESSAGES.  SOME                07004000 T  0003:0814:0
                        CARE IS ALSO TAKEN TO PREVENT READING BEYOND THE "END.";                07005000 T  0003:0814:0
            PROCEDURE COMPOUNDTAIL;                                                             07006000 T  0003:0814:0
PRT(766) = COMPOUNDTAIL
                        BEGIN LABEL ANOTHER;                                                    07007000 T  0003:0814:0
                                                                                          START OF SEGMENT **********   69
                          I ← I-1; BEGINCTR ← BEGINCTR+1;                                       07008000 T  0069:0000:0
                 ANOTHER:   ERRORTOG ← TRUE;   COMMENT  ALLOW ERROR MESSAGES;                   07009000 T  0069:0002:1
                          STEPIT;                                                               07010000 T  0069:0003:3
                          IF STREAMTOG THEN STREAMSTMT ELSE STMT;                               07011000 T  0069:0004:0
                          IF ELCLASS = SEMICOLON THEN GO TO ANOTHER;                            07012000 T  0069:0006:1
                          IF ELCLASS ≠ ENDV                                                     07013000 T  0069:0007:3
                            THEN BEGIN                                                          07014000 T  0069:0008:0
                              ERR(119); GO TO ANOTHER END;                                      07015000 T  0069:0009:2
                        ENDTOG←TRUE;                                                            07016000 T  0069:0010:0
                        DO STOPDEFINE←TRUE UNTIL                                                07017000 T  0069:0011:2
                             STEPI≤ENDV AND ELCLASS≥UNTILV                                      07018000 T  0069:0011:3
                             OR NOT ENDTOG;                                                     07019000 T  0069:0013:2
                        ENDTOG←FALSE;                                                           07020000 T  0069:0015:2
                             IF BEGINCTR ← BEGINCTR-1 ≠ 0 EQV ELCLASS = PERIOD                  07021000 T  0069:0015:3
                               THEN BEGIN                                                       07022000 T  0069:0017:3
                                  IF BEGINCTR = 0 THEN                                          07023000 T  0069:0019:2
                                   BEGIN FLAG(143); BEGINCTR ← 1; GO ANOTHER END;               07024000 T  0069:0019:3
FLAG (120);                                                                           %104-     07025000 P  0069:0022:0
FCR:= (LCR:=MKABS(CBUFF[9]))-9;                                                       %104-     07025010 C  0069:0023:2
IF LISTER THEN PRINTCARD;                                                             %104-     07025020 C  0069:0026:0
FCR:= (LCR:=MKABS(TBUFF[9]))-9  END;                                                  %104-     07025030 C  0069:0042:0
                        IF ELCLASS = PERIOD THEN                                            07026000 T  0069:0045:2
                          BEGIN                                                             07027000 T  0069:0046:0
                            GT5 ← "ND;END."&"E"[1:43:5];                                    07028000 T  0069:0046:1
                            MOVE(1,GT5,CBUFF[0]);                                           07029000 T  0069:0048:0
                            LASTUSED←4;                                                     07030000 T  0069:0050:0
                            ELBAT[I←I-2] ←SPECIAL[20];                                      07031000 T  0069:0050:1
                            ELCLASS ← SEMICOLON END                                         07032000 T  0069:0053:2
                    END COMPOUNDTAIL;                                                       07033000 T  0069:0054:0
                                                                                          69 IS   56 LONG, NEXT SEG    3


              COMMENT ACTUAPARAPART IS RESPONSIBLE FOR CONSTRUCTING ALL CALLS ON            07034000 T  0003:0814:0
                        PARAMETERS.  IT HANDLES THE ENTIRE PARAMETER LIST WITH           07035000 T  0003:0814:0
                        ONE CALL.  IT IS ALSO RESPONSIBLE FOR CHECKING FOR              07036000 T  0003:0814:0
```

```
                                NON-CORRESPONDENCE OF THE ACTUAL AND FORMAL PARAMETERS.          07037000 T  0003!081410
                                CONCERNING THE PARAMETERS;                                       07038000 T  0003!081410
                                     FBIT TELLS IF THE PROCEDURE BEING CALLED IS FORMAL          07039000 T  0003!081410
                                          OR NOT.                                                07040000 T  0003!081410
                                     SBIT TELLS IF THE PROCEDURE BEING CALLED IS A STREAM        07041000 T  0003!081410
                                          PROCEDURE OR NOT.                                      07042000 T  0003!081410
                                     INDEX IS THE INDEX INTO INFO OF THE ADDITIONAL              07043000 T  0003!081410
                                          INFORMATION;                                           07044000 T  0003!081410
                          PROCEDURE ACTUALPARAPART(FBIT,SBIT,INDEX);                             07045000 T  0003!081410
PRT(767) = ACTUALPARAPART
                                VALUE FBIT,SBIT,INDEX;                                           07046000 T  0003!081410
                                BOOLEAN FBIT,SBIT;                                               07047000 T  0003!081410
                                INTEGER INDEX;                                                   07048000 T  0003!081410
                                BEGIN                                                            07049000 T  0003!081410
                                  INTEGER PCTR,ACLASS,SCLASS;                                    07050000 T  0003!081410
                                                                                  START OF SEGMENT **********   70
STACK(F+2) = PCTR
STACK(F+3) = ACLASS
STACK(F+4) = SCLASS
                                     COMMENT                                                     07051000 T  0070!000010
                                          PCTR IS A COUNT OF THE NUMBER OF PARAMETERS            07052000 T  0070!000010
                                            COMPILED.                                            07053000 T  0070!000010
                                          ACLASS IS THE CLASS OF THE ACTUAL PARAMETER-           07054000 T  0070!000010
                                          SCLASS IS THE CLASS OF THE FORMAL PARAMETER.           07055000 T  0070!000010
                                            THEY ARE PUT IN A NORMALIZED FORM IN ORDER           07056000 T  0070!000010
                                            TO ALLOW INTEGER, REAL, AND ALPHA TO HAVE            07057000 T  0070!000010
                                            SIMILAR MEANINGS;                                    07058000 T  0070!000010
                                  REAL WHOLE;                                                    07059000 T  0070!000010
STACK(F+5) = WHOLE
                                     COMMENT WHOLE CONTAINS THE ELBAT WORD OF THE ACTUAL         07060000 T  0070!000010
                                          PARAMETERS;                                            07061000 T  0070!000010
                                  BOOLEAN VBIT;                                                  07062000 T  0070!000010
STACK(F+6) = VBIT
                                     COMMENT VBIT TELLS WHETHER OR NOT THE PARAMETER IS TO       07063000 T  0070!000010
                                          BE CALLED BY VALUE OR BY NAME;                         07064000 T  0070!000010
                                  LABEL ANOTHER,NORMAL,VE,STORE,LRTS,LOWBD,FINISHBOO,            07065000 T  0070!000010
                                        LODPOINT,NSBS,BS,COMMON,LP,GOBBLE,BSXX,BSX,EXIT,         07066000 T  0070!000010
                                        CERR,FGEN;                                               07067000 T  0070!000010
                                  LABEL                                                          07068000 T  0070!000010
                                     L4,L5,L6,L7,L8,L9,L10,L11,L12,L13,L14,L15,L16,L17,          07069000 T  0070!000010
                                     L18,L19,L20,L21,L22,L23,L24,L25,L26,L27,L28,L29,L30,        07070000 T  0070!000010
                                     L31,L32,L33;                                                07071000 T  0070!000010
                                  SWITCH S :=                                                    07072000 T  0070!000010
                                     L4,L5,L6,L7,L8,L9,L10,L11,L12,L13,L14,L15,L16,L17,          07073000 T  0070!000211
                                     L18,L19,L20,L21,L22,L23,L24,L25,L26,L27,L28,L29,L30,        07074000 T  0070!000211
                                     L31,L32,L33;                                                07075000 T  0070!000211
                                REAL T1,T2,T3,T4,T5,T6; COMMENT EXAMINE LATER WITH EYE           07076000 T  0070!001810
STACK(F+7) = T1
STACK(F+10) = T2
STACK(F+11) = T3
STACK(F+12) = T4
STACK(F+13) = T5
STACK(F+14) = T6
                                     TO REDUCING TOTAL NUMBER;                                   07077000 T  0070!001810
                                PCTR := 1;                                                       07078000 T  0070!001810
                      ANOTHER:  ACLASS := STEPI; WHOLE := ELBAT[I]; SCATTERELBAT;                07079000 T  0070!001913
                                     STACKCT := 0;                                      %A       07079500 T  0070!002211
```

```
COMMENT SETUP FIELDS OF AN ACTUAL PARAMETER;                                    07080000 T  0070:002:2
        IF FBIT THEN BEGIN VBIT ← FALSE; SCLASS ← LOCLID END                    07081000 T  0070:002:2
COMMENT   IF PROCEDURE IS FORMAL ALL CALLS ARE BY NAME AND NO CHECK             07082000 T  0070:002:3
        IS MADE FOR CORRESPONDENCE OF ACTUAL AND FORMAL PARA                    07083000 T  0070:002:3
        METERS  SETTING SCLASS TO LOCID HELPS TO COMPRESS CHECK                 07084000 T  0070:002:3
          ELSE BEGIN                                                            07085000 T  0070:002:3
             VBIT ← BOOLEAN(GT1← TAKE(INDEX+PCTR)),VO;                          07086000 T  0070:002:0
             IF SCLASS ←GT1,CLASS ≤ INTARRAYID AND                              07087000 T  0070:002:1
                                  SCLASS ≥ BOOSTRPROCID                         07088000 T  0070:003:1
               THEN IF GT1 ← (SCLASS - BOOSTRPROCID) MOD 4 ≠ 0                  07089000 T  0070:003:1
                      THEN SCLASS ← SCLASS-GT1+1                                07090000 T  0070:003:3
COMMENT IF PROCEDURE IS NOT FORMAL WE OBTAIN VBIT FROM THE ADDITION-            07091000 T  0070:003:3
        AL INFO FOR THE PROCEDURE.  WE ALSO GET SCLASS FROM THIS                07092000 T  0070:003:3
        SOURCE, HOWEVER SCLASS IS NORMALIZED TO REAL, IF NEEDED;                07093000 T  0070:003:3
          END;                                                                  07094000 T  0070:003:3
        IF T1 ← TABLE(I+1) ≠ COMMA THEN                                         07095000 T  0070:003:1
        IF T1 ≠ RTPAREN THEN                                                    07096000 T  0070:003:1
COMMENT   THE ACTUAL PARAMETER HAS MORE THAN ONE LOGICAL QUANTITY -             07097000 T  0070:004:0
        HENCE A DIFFERENT ANALYSIS IS REQUIRED;                                 07098000 T  0070:004:0
    BEGIN IF ACLASS ≤ IDMAX OR ACLASS = SUPERLISTID THEN                        07099000 T  0070:004:0
                CHECKFR(WHOLE);                                                 07099500 T  0070:004:2
              IF ACLASS < BOOARRAYID OR ACLASS > INTARRAYID                      07100000 T  0070:004:3
               THEN BEGIN                                                       07101000 T  0070:004:1
COMMENT   THE ACTUAL PARAMETER DOES NOT START WITH AN ARRAY NAME -              07102000 T  0070:004:3
        HENCE THE PARAMETER IS AN EXPRESSION, A SUPERFORMAT, A                  07103000 T  0070:004:3
        SUPERFILE, AN INDEXED FILE OR SUPERLIST;                                07104000 T  0070:004:3
              IF ACLASS = SUPERFRMTID THEN                                      07105000 T  0070:004:3
                BEGIN ACLASS ← FRMTID; GO TO FGEN END;                          07106000 T  0070:004:1
              IF ACLASS = SUPERFILEID OR ACLASS = FILEID                        07107000 T  0070:004:0
                THEN BEGIN                                                      07108000 T  0070:004:2
                   T4←L; EMITO(NOP) ;%MAY NEED FOR FILEATTRIBUTES,              07108500 T  0070:005:1
                   IF NOT VBIT THEN EMITO(NOP) ; % DITTO,                       07108505 T  0070:005:0
                   ACLASS ← FILEID;                                             07109000 T  0070:005:3
COMMENT  IT IS EITHER AN INDEXED FILE OR A SUPERFILE (OR BOTH);                 07110000 T  0070:005:1
                   PASSFILE;                                                    07111000 T  0070:005:1
                   IF ELCLASS=PERIOD THEN % THEN FILE ATTRIBUTE                 07111200 T  0070:005:2
                        BEGIN                                                   07111210 T  0070:005:3
                        IF VBIT THEN                                            07111220 T  0070:005:0
                             BEGIN                                              07111225 T  0070:005:1
                             T5←L; L←T4; EMITO(MKS); L←T5; T5←0 ;               07111230 T  0070:005:2
                             END ;                                              07111235 T  0070:006:1
                        ACLASS←IF FILEATTRIBUTEHANDLER(FA)=ATYPE                07111240 T  0070:006:1
                               THEN REALID ELSE BOOID ;                         07111250 T  0070:006:3
                        IF ELCLASS≠COMMA AND ELCLASS≠RTPAREN THEN               07111255 T  0070:006:0
                        IF ACLASS=BOOID THEN SIMPBOO ELSE                       07111260 T  0070:006:3
                             BEGIN                                              07111265 T  0070:006:0
                             SIMPARITH ;                                        07111270 T  0070:006:1
                             IF ELCLASS=RELOP THEN                              07111275 T  0070:006:2
                                  BEGIN                                         07111280 T  0070:006:3
                                  ACLASS←BOOID;  RELATION ;                     07111285 T  0070:007:0
                                  SIMPBOO ;                                     07111290 T  0070:007:3
                                  END ;                                         07111295 T  0070:007:0
                             END ;                                              07111300 T  0070:007:0
                        IF NOT VBIT THEN                                        07111303 T  0070:007:0
                             BEGIN                                              07111307 T  0070:007:1
                             EMITPAIR(JUNK,STD);  EMITN(JUNK) ;                 07111310 T  0070:007:2
                             EMITO(RTS);  ADJUST;  CONSTANTCLEAN ;              07111315 T  0070:007:1
                             EMITO(MKS);  EMITB(BBW,BUMPL,T4+2) ;               07111320 T  0070:007:1
```

```
                                    EMITB(BFW,T4+2,L) ;                                  07111325 T  0070:0080:0
                                    STUFFF(PROGDESCBLDR(0,L-3,0)) ;                      07111330 T  0070:0081:3
                                    END ;                                                07111335 T  0070:0084:0
                                GO BS ;                                                  07111340 T  0070:0084:0
                                END OF FILE ATTRIBUTE PARAMETER EXPRESSION;               07111345 T  0070:0084:1
                            IF ELCLASS ≠ LEFTPAREN THEN GO TO BS;                        07112000 T  0070:0084:1
                            I ← I-1;                                                     07113000 T  0070:0085:3
   COMMENT  IF WE ARE HERE IT IS INDEXED;                                                07114000 T  0070:0087:2
                             CHECKPRESENCE;                                              07115000 T  0070:0087:2
                            EMITO(LOD); PANA; EMITO(CDC);                                07116000 T  0070:0087:3
                            IF SCLASS = FILEID OR NOT SBIT OR VBIT                       07117000 T  0070:0089:3
                              THEN BEGIN ERR(121); GO TO CERR END                        07118000 T  0070:0091:2
   COMMENT   AN INDEXED FILE MAY BE PASSED BY NAME ONLY AND ONLY TO A                    07119000 T  0070:0093:2
        STREAM PROCEDURE THE STREAM PROCEDURE MAY NOT DO A                               07120000 T  0070:0093:2
        RELEASE ON THIS DESCRIPTOR                                                       07121000 T  0070:0093:2
                            ELSE GO TO COMMON END ;                                      07122000 T  0070:0093:2
        IF ACLASS = SUPERLISTID THEN BEGIN BANA;                                         07122500 T  0070:0093:2
             EMITV(WHOLE.ADDRESS);                                                       07122510 T  0070:0095:2
             IF WHOLE.ADDRESS>1023 THEN EMITO(PRTE);                                     07122520 T  0070:0096:0
                EMITO(LOD);                                                              07122530 T  0070:0098:1
          ACLASS←LISTID; GO TO BS END;                                                   07122540 T  0070:0099:3
   COMMENT   NORMAL IS REACHED ONLY IF THE PARAMETER IS AN EXPRESSION;                   07123000 T  0070:0100:1
NORMAL:     IF VBIT THEN                                                                07124000 T  0070:0100:1
VE:         T1 ← EXPRSS     COMMENT VALUE CALL EXPRESSION;                               07125000 T  0070:0101:2
            ELSE BEGIN      COMMENT NAME CALL EXPRESSION;                                07126000 T  0070:0102:0
              IF SBIT THEN BEGIN FLAG(122); GO TO CERR END;                              07127000 T  0070:0103:3
   COMMENT   STREAM PROCEDURES MAY NOT HAVE EXPRESSIONS PASSED BY NAME;                  07128000 T  0070:0105:3
              T2 ← BAE;                                                                  07129000 T  0070:0105:3
              T3 ← PROGDESCBLDR(0,L,0);                                                  07130000 T  0070:0106:1
   COMMENT   BUILD DESCRIPTOR FOR ACCIDENTAL ENTRY AND PREPARE JUMP                      07131000 T  0070:0108:0
        AROUND CODE FOR EXPRESSION;                                                      07132000 T  0070:0108:0
              T1 ← EXPRSS;  COMMENT  COMPILE EXPRESSION;                                 07133000 T  0070:0108:0
STORE:      EMITPAIR(JUNK,STD); EMITN(JUNK);                                            07134000 T  0070:0109:2
   COMMENT   THIS PROVIDES FOR PROTECTION IF ONE ATTEMPTS INSIDE OF A                    07135000 T  0070:0111:3
        PROCEDURE TO STORE INTO AN EXPRESSION - THE STORE GOES                           07136000 T  0070:0111:3
        INTO JUNK;                                                                       07137000 T  0070:0111:3
LRTS:       EMITO(RTS); CONSTANTCLEAN; EMITB(BFW,T2,L); STUFFF(T3)                       07138000 T  0070:0111:3
   COMMENT   LRTS IS RESPONSIBLE FOR THE CLEANUP ASSOCIATED WITH ALL                     07139000 T  0070:0114:1
        THE ACCIIDENTAL ENTRIES COMPILED BY ACTUALPARAPART.  IT                          07140000 T  0070:0115:2
        EMITS THE RETURN SPECIAL, DOES A CONSTANTCLEAN, FINISHES                         07141000 T  0070:0115:2
        THE BRANCH OPERATION AND PROVIDES FOR THE POSSIBILITY                            07142000 T  0070:0115:2
        OF STUFFING F INTO THE ACCIDENTAL ENTRY DESCRIPTOR;                              07143000 T  0070:0115:2
        END OF NAME CALL EXPRESSIONS;                                                    07144000 T  0070:0115:2
        ACLASS ← IF T1 = ATYPE THEN REALID ELSE IF T1 = BTYPE                            07145000 T  0070:0115:2
             THEN BOOID ELSE LABELID; GO TO BS;                                          07146000 T  0070:0117:3
          END OF EXPRESSION CALL CODE;                                                   07147000 T  0070:0120:1
   COMMENT IF WE REACH THIS POINT THE ACTUAL PARAMETER STARTS WITH AN                    07148000 T  0070:0120:1
        ARRAY NAME FOLLOWED BY SOMETHING ELSE;                                           07149000 T  0070:0120:1
        IF SCLASS ≥ BOOARRAYID THEN                                                      07150000 T  0070:0120:1
        IF SCLASS ≤ INTARRAYID THEN                                                      07151000 T  0070:0121:2
          BEGIN  T2 ← TAKE(INDEX+PCTR).INCR;                                             07152000 T  0070:0122:1
   COMMENT THE FORMAL PARAMETER CALLS FOR AN ARRAY AS ACTUAL PARAMETER,                  07153000 T  0070:0125:2
        THUS WE MUST HAVE A ROW DESIGNATOR;                                              07154000 T  0070:0125:2
             IF ACLASS ≠ BOOARRAYID THEN ACLASS ← REALARRAYID;                           07155000 T  0070:0125:2
   COMMENT NORMALISE ACLASS FOR LATER COMPARISON;                                        07156000 T  0070:0127:2
             VARIABLE(FA); IF TABLE(I-2) ≠ FACTOP                                       07157000 T  0070:0127:2
                THEN BEGIN ERR(123); GO TO EXIT END;                                     07158000 T  0070:0129:2
```

```
      COMMENT   IT MUST BE A ROW DESIGNATOR - OTHERWISE IT IS AN ERROR;
      COMMENT VARIABLE EMITS LOWER BOUNDS FOR EACH ASTERISK SUBSCRIPT.
                STLB IS THE NUMBER OF SUCH SUBSCRIPTS;
LOWBD:        IF T2 ≠ STLB THEN BEGIN FLAG(124); GO TO CERR END
                THE FORMAL PARAMETER MUST BE AN ARRAY OF ONE DIMENSION
                     ELSE GO TO BS END;
                IF VBIT THEN GO TO VE;
      COMMENT   IF THE FORMAL PARAMETER DOES NOT CALL FOR AN ARRAY AND
                  VBIT IS SET WE MUST HAVE A VALUE CALL EXPRESSION;
                IF SBIT
                  THEN BEGIN
                    T6 ← FL; VARIABLE(T6);
                    IF T6 ≠ 0 THEN GO TO BS;
                    FLAG(122);GO TO CERR END;
      COMMENT   IF PROCEDURE IS A STREAM PROCEDURE THEN WE COMPILE NAME
                  CALL EXPRESSION.  IT MUST BE SIMPLY A SUBSCRIPTED
                  VARIABLE OR A ROW DESIGNATOR.  IF VARIABLE DOES MORE
                  THAN THIS IT SETS T6 TO ZERO;
      COMMENT   IF THIS PLACE IS REACHED WE HAVE A NON-STREAM PROCEDURE.
                  WE HAVE NOT YET DECEIDED WHETHER WE HAVE
                     1) A ROW DESIGNATOR WITH FORMAL PROCEDURE,
                     2) A SUBSCRIPTED VARIABLE, OR
                     3) A GENUINE NAME CALL EXPRESSION;
                IF TABLE(I+2) = LITNO AND
                   ( GT1 ← TABLE(I+4) = COMMA OR GT1 = RTPAREN)
                  THEN BEGIN
      COMMENT  WE HAVE HERE A ONE DIMENSIONAL SUBCRIPTED VARIABLE WITH
                 CONSTANT LOWER BOUNDS.  WE MAKE A SPECIAL CASE TO AVOID
                   ACCIDENTAL ENTRY AND ADDITIONAL PRT CELL;
                     VARIABLE(FL);
                     ACLASS ← IF ACLASS = BOOARRAYID THEN BOOID ELSE
                         REALID;  GO TO BS END;
                   T2 ← BAE; T3 ← L;
        COMMENT WE PREPARE FOR ACCIDENTAL ENTRY EVEN THOUGH WE KNOW NOT YET
                   IF WE HAVE ROW DESIGNATOR;
                   T6 ← FA; VARIABLE(T6);
                   IF TABLE(I-2) = FACTOP
                     THEN BEGIN
        COMMENT WE HAVE A ROW DESIGNATOR AFTER ALL;
                       EMITB(BFW,T2,T3); T2 ← STLB; GO TO LOWBD END;
        COMMENT      WE NOW KNOW WE NEED ACCIDENTAL ENTRY;
                   T3 ← PROGDESCBLDR(0,T3,0);
                   T1 ← IF BOOARRAYID = ACLASS THEN BTYPE ELSE ATYPE;
                   IF ELCLASS = COMMA OR ELCLASS = RTPAREN THEN
        COMMENT    WE ARE AT END OF PARAMETER;
                      IF T6 = 0 THEN COMMENT MORE THAN SUBSCRIPTED VARIABLE;
                        GO TO STORE ELSE COMMENT SUBSCRIPTED VARIABLE;
                        GO TO LRTS;
                   IF T1 = BTYPE THEN GO TO FINISHBOO; SIMPARITH;
                   IF ELCLASS = RELOP THEN BEGIN T1 ← BTYPE; RELATION;
FINISHBOO:    SIMPBOO END; GO TO STORE END;
      COMMENT WHEN WE GET HERE WE HAVE THE CASE OF A SINGLE QUANTITY
                ACTUAL PARAMETER;
           IF ACLASS ≤ IDMAX OR ACLASS = SUPERLISTID THEN
                     CHECKER(WHOLE); STEPIT;
                GO TO S[ACLASS-3];
           IF ACLASS = 0 THEN FLAG(100) ELSE
```

```
07159000 T  0070:0131:3
07163000 T  0070:0131:3
07164000 T  0070:0131:3
07165000 T  0070:0131:3
07166000 T  0070:0134:1
07167000 T  0070:0134:1
07168000 T  0070:0134:1
07169000 T  0070:0135:3
07170000 T  0070:0135:3
07171000 T  0070:0135:3
07172000 T  0070:0135:3
07173000 T  0070:0136:0
07174000 T  0070:0137:3
07175000 T  0070:0139:2
07176000 T  0070:0140:0
07177000 T  0070:0140:0
07178000 T  0070:0140:0
07179000 T  0070:0140:0
07180000 T  0070:0140:0
07181000 T  0070:0140:0
07182000 T  0070:0140:0
07183000 T  0070:0140:0
07184000 T  0070:0140:0
07185000 T  0070:0140:0
07186000 T  0070:0142:0
07187000 T  0070:0144:1
07188000 T  0070:0146:0
07189000 T  0070:0146:0
07190000 T  0070:0146:0
07191000 T  0070:0146:0
07192000 T  0070:0146:1
07193000 T  0070:0148:1
07194000 T  0070:0150:0
07195000 T  0070:0151:3
07196000 T  0070:0151:3
07197000 T  0070:0151:3
07198000 T  0070:0153:2
07199000 T  0070:0154:1
07200000 T  0070:0155:3
07201000 T  0070:0155:3
07202000 T  0070:0158:0
07203000 T  0070:0158:0
07204000 T  0070:0159:3
07205000 T  0070:0162:1
07206000 T  0070:0164:0
07207000 T  0070:0164:0
07208000 T  0070:0165:3
07209000 T  0070:0165:3
07210000 T  0070:0166:1
07211000 T  0070:0168:0
07212000 T  0070:0170:1
07213000 T  0070:0172:0
07214000 T  0070:0172:0
07215000 T  0070:0172:0
07215500 T  0070:0173:3
07216000 T  0070:0175:3
07217000 T  0070:0178:0
```

```
                IF ACLASS= SUPERLISTID THEN                                          07217500 T  0070:0180:0
                    BEGIN EMITPAIR(ADDRSF,LOD); GO TO BS END;                        07217510 T  0070:0181:2
            FLAG(126);                                                               07217520 T  0070:0183:2
CERR:                                                                                07218000 T  0070:0184:0
L12:L13:L14:L15:L16:                                                                 07219000 T  0070:0184:0
   COMMENT STREAM PROCEDURES MAY NOT BE PASSED AS PARAMETERS;                        07220000 T  0070:0184:0
                FLAG(125); ERRORTOG ← TRUE; GO TO COMMON;                            07221000 T  0070:0184:0
LODPOINT:                                                                            07222000 T  0070:0186:0
L4:L8:                                                                               07223000 T  0070:0186:0
                COMMENT  LIST, SUPERLIST OR SUPERFILE;                               07224000 T  0070:0186:0
                EMITPAIR(ADDRSF,LOD);                                                07225000 T  0070:0186:0
NSBS:           IF SBIT THEN BEGIN FLAG(127); GO TO CERR END;                        07226000 T  0070:0187:2
   COMMENT      ITEMS WHICH FIND THEIR WAY HERE MAY NOT BE PASSED TO                 07227000 T  0070:0189:2
                STREAM PROCEDURES;                                                   07228000 T  0070:0189:2
BS:             IF SCLASS ≠ ACLASS THEN                                              07229000 T  0070:0189:2
                IF SCLASS ≠ LOCLID THEN                                              07230000 T  0070:0189:3
   COMMENT      IF WE ARRIVE HERE THE ACTUAL AND FORMAL PARAMETERS DO NOT            07231000 T  0070:0191:2
                AGREE;                                                               07232000 T  0070:0191:2
                    BEGIN FLAG(123); GO TO CERR END;                                 07233000 T  0070:0191:2
COMMON:                                                                              07234000 T  0070:0192:1
   COMMENT      ARRIVAL HERE CAUSES THE NEXT PARAMETER TO BE EXAMINED;               07235000 T  0070:0193:2
                PCTR ← PCTR+1;                                                       07236000 T  0070:0193:2
                IF ELCLASS = COMMA THEN GO TO ANOTHER;                               07237000 T  0070:0194:0
                IF ELCLASS ≠ RTPAREN                                                 07238000 T  0070:0195:3
                    THEN BEGIN ERROR(129); GO TO EXIT END;                           07239000 T  0070:0195:3
                IF NOT FBIT THEN                                                     07240000 T  0070:0198:0
                IF TAKE(INDEX).NODIMPART+1 ≠ PCTR                                    07241000 T  0070:0198:1
                  THEN BEGIN COMMENT WRONG NUMBER OF PARAMETERS;                     07242000 T  0070:0200:1
                    ERR(128); GO TO EXIT END;                                        07243000 T  0070:0201:3
                STEPIT; GO TO EXIT;                                                  07244000 T  0070:0203:2
L5:                                                                                  07245000 T  0070:0204:0
   COMMENT FORMATS;                                                                  07246000 T  0070:0204:0
                I←I-1;                                                               07247000 T  0070:0204:0
FGEN:           PASSFORMAT;                                                          07248000 T  0070:0205:2
                IF SBIT THEN BEGIN EMITO(XCH); EMITO(CDC) END;                       07249000 T  0070:0206:1
                I←I+1;                                                               07250000 T  0070:0208:1
                GO TO BS;                                                            07251000 T  0070:0210:0
L6:                                                                                  07252000 T  0070:0210:1
   COMMENT      SUPERFORMAT;                                                         07253000 T  0070:0211:2
                IF FBIT                                                              07254000 T  0070:0211:2
                  THEN BEGIN EMITV(ADDRSF); ADDRSF ← ADDRSF-1 END                    07255000 T  0070:0211:2
                  ELSE BEGIN I ← I -1; EMITL(TAKEFRST); I ← I+1 END;                 07256000 T  0070:0213:3
                GO TO LODPOINT;                                                      07257000 T  0070:0217:3
L7:                                                                                  07258000 T  0070:0218:0
   COMMENT      FILE;                                                                07259000 T  0070:0219:2
                I ← I-1; ELCLASS ← FILEID;                                           07260000 T  0070:0219:2
                PASSFILE; GO TO BS;                                                  07261000 T  0070:0221:2
L9:                                                                                  07262000 T  0070:0222:0
   COMMENT      SWITCH;                                                              07263000 T  0070:0222:0
                IF FORMALF THEN GO TO LODPOINT;                                      07264000 T  0070:0222:0
   COMMENT      OTHERWISE WE BUILD ACCIDENTAL ENTRY AND SET UP SO THAT               07265000 T  0070:0223:2
                MCP HANDLES LABEL PROPERLY.  SEE IN PARTICULAR OTHER                 07266000 T  0070:0223:2
                DISCUSSIONS OF GO TO PROBLEM.  IT SHOULD BE NOTED THAT               07267000 T  0070:0223:2
                ALL BUT VERY SIMPLE SWITCHES ARE MARKED FORMAL, WHETHER              07268000 T  0070:0223:2
                THEY ARE OR NOT;                                                     07269000 T  0070:0223:2
                T2 ← BAE; T3←PROGDESCBLDR(0,L,0); EMITV(GNAT(WHOLE));                07270000 T  0070:0223:2
                GENGO(WHOLE);                                                        07271000 T  0070:0227:2
```

```
                EMITO(RTS); EMITB(BFW,T2,L); STUFFF(T3); GO TO NSBS;                07272000 T  0070:022?3
L10:                                                                               07273000 T  0070:023!2
   COMMENT      PROCEDURE;                                                         07274000 T  0070:023!2
                TB1 ← TRUE; IF FORMALF THEN GO TO LODPOINT;                        07275000 T  0070:023!2
LP:             IF T1 ← TAKE(WHOLE ← GIT(WHOLE)).[40:8] = 0                        07276000 T  0070:023?!
                  THEN BEGIN                                                       07277000 T  0070:023?3
   COMMENT      THE PROCEDURE BEING PASSED HAS ZERO PARAMETERS;                    07278000 T  0070:023?!
                  IF TB1 THEN GO TO LODPOINT;                                      07279000 T  0070:023?!
   COMMENT      IF THE PROCEDURE IS NOT A FUNCTION, WE PASS THE PROCEDURE          07280000 T  0070:023?3
                DESCRIPTOR ITSELF (IN BOTH CASES THE PARAMETER PROCEDURE);         07281000 T  0070:023?3
                  IF NOT FBIT THEN                                                 07281900 T  0070:023?3
                  IF SCLASS ≤ INTPROCID THEN SCLASS ← SCLASS+4;                    07282000 T  0070:023?0
                  I ← I-2; STEPIT;                                                 07283000 T  0070:024!2
                  GO TO NORMAL;  COMMENT WE LET OUT NORMAL MECHANISM FOR            07284000 T  0070:024?2
                                         EXPRESSIONS HANDLE THIS CASE;             07285000 T  0070:024?3
                  END THE CASE OF ZERO PARAMETERS;                                 07286000 T  0070:024?3
                TB1 ← TRUE;                                                        07287000 T  0070:024?3
                FOR T2 ← 1 STEP 1 UNTIL T1                                         07288000 T  0070:024?0
                  DO BEGIN                                                         07289000 T  0070:024?0
                     IF BOOLEAN(T3←TAKE(WHOLE←T2)).VO                              07290000 T  0070:024?2
                       THEN                                                        07291000 T  0070:024?!2
                        IF T4 ← T3.CLASS < BOOARRAYID OR T4 > INTARRAYID            07292000 T  0070:024?!2
                         THEN BEGIN                                                07293000 T  0070:024?3
   COMMENT THE T2-TH PARAMETER TO THE PROCEDURE BEING PASSED IS VALUE;             07294000 T  0070:025!!2
                          IF TB1 THEN                                              07295000 T  0070:025!!2
                           BEGIN                                                   07296000 T  0070:025!!2
   COMMENT THIS IS THE FIRST VALUE PARAMETER,  IF ANY PARAMETERS ARE                07297000 T  0070:025!!3
                VALUE WE BUILD A THINK WHICH SEES THAT WHEN THIS                   07298000 T  0070:025!!3
                PROCEDURE IS CALLED FORMALLY, ITS PARAMETERS THAT ARE              07299000 T  0070:025!!3
                VALUE GET CALLED BY VALUE.  SINCE THIS IS FIRST VALUE              07300000 T  0070:025!!3
                PARAMETER WE CONSTRUCT THUNK HERE AND INHIBIT FUTURE THUNK         07301000 T  0070:025!!3
                CONSTRUCTIONS;                                                     07302000 T  0070:025!!3
GOBBLE:                                                                            07303000 T  0070:025!!3
                             TB1 ← FALSE; T5 ← BAE;                                07304000 T  0070:025?!0
                             T6 ← PROGDESCBLDR(1,L,0) END;                         07305000 T  0070:025?!3
                           EMITV(T4 ← T3.ADDRESS); EMITPAIR(T4,STD)END END;        07306000 T  0070:025?!3
   COMMENT THIS CALLS THE T2-TH PARAMETER BY VALUE;                                07307000 T  0070:025?!0
                IF NOT TB1                                                         07308000 T  0070:026!!1
                  THEN BEGIN                                                       07309000 T  0070:026!!1
   COMMENT      THERE WERE VALUE CALLS SO FINISH CONSTRUCTION OF THINK;            07310000 T  0070:026!!3
                     EMITPAIR(ADDRSF,LOD); EMITO(BFW);                             07311000 T  0070:026!!3
                     CONSTANTCLEAN; EMITB(BFW,T5,L); ADDRSF ← T6 END;              07312000 T  0070:026?!2
                GO TO LODPOINT;  COMMENT IN ANY CASE LOAD A DESCRIPTOR;            07313000 T  0070:026?!3
L11:                                                                               07314000 T  0070:026!!0
   COMMENT      INTRINSIC PROCEDURE;                                               07315000 T  0070:026?!2
                ADDRSF ← GNAT(WHOLE);                                              07316000 T  0070:026?!2
   COMMENT      GET PRT SPACE IF NOT ASSIGNED;                                     07317000 T  0070:026?!0
                ACLASS ← REALPROCID;                                               07318000 T  0070:026?!0
                T3.ADDRESS ← 897; T2←T1←1; GO TO GOBBLE;                           07319000 T  0070:026?!2
                COMMENT  THIS MAKES THE INTRINSICS LOOK LIKE ORDINARY              07320000 T  0070:027?!1
                         PROCEDURES;                                               07321000 T  0070:027?!1
L19:L20:                                                                           07322000 T  0070:027?!1
   COMMENT      ALFAPROC AND INTPROC;                                              07323000 T  0070:027?!2
                ACLASS ← REALPROCID;                                               07324000 T  0070:027?!2
L17:L18:                                                                           07325000 T  0070:027?!3
   COMMENT      BOOPROC AND REAL PROC;                                             07326000 T  0070:027?!0
                IF FORMALF                                                         07327000 T  0070:027?!0
```

```
                THEN BEGIN                                                          07328000 T  0070:027:0
   COMMENT THE PROCEDURE BEING PASSED IS ACTUALLY A FORMAL PARAMETER;               07329000 T  0070:027:1
                IF SCLASS > INTPROCID THEN ACLASS ← ACLASS+4;                       07330000 T  0070:027:1
   COMMENT CHANGE ACLASS SO THAT IT LOOKS LIKE WE ARE PASSING AN                    07331000 T  0070:027:2
             EXPRESSION.  THE FORMAL PARAMETER DOES NOT CALL FOR A                  07332000 T  0070:027:2
             PROCEDURE SO IT MUST CALL FOR AN EXPRESSION;                           07333000 T  0070:027:2
                IF VBIT                                                             07334000 T  0070:027:2
                  THEN BEGIN EMITV(ADDRSF); GO TO BS END                            07335000 T  0070:027:2
                  ELSE GO TO LODPOINT;                                              07336000 T  0070:027:2
   COMMENT    IF VBIT WE DO VALUE CALL. OTHERWISE WE PASS PROCEDURE                  07337000 T  0070:027:2
              DESCRIPTOR ALONG;                                                     07338000 T  0070:027:2
                END;                                                                07339000 T  0070:027:2
              TB1 ← FALSE; GO TO LP;                                                07340000 T  0070:027:2
L23:L24:                                                                            07341000 T  0070:028:1
   COMMENT    INTEGER AND ALPHA IDS;                                               07342000 T  0070:028:2
              ACLASS ← REALID;                                                      07343000 T  0070:028:2
L21:L22:                                                                            07344000 T  0070:028:3
   COMMENT    BOOLEAN AND REAL IDS;                                                07345000 T  0070:028:0
              IF VBIT THEN EMITV(ADDRSF)                                            07346000 T  0070:028:0
                ELSE IF NOT(SBIT OR VONF) AND FORMALF                               07347000 T  0070:028:2
                     THEN GO TO LODPOINT ELSE EMITN(ADDRSF);                        07348000 T  0070:028:2
   COMMENT    JUST PASS THE DESCRIPTOR ALONG IF PROCEDURE IS NOT STREAM            07349000 T  0070:028:2
              AND ACTUAL PARAMETER IS A NAME CALL FORMAL PARAMETER.  IF            07350000 T  0070:028:2
              THESE CONDITIONS ARE NOT MET DO DESCRIPTOR CALL;                      07351000 T  0070:028:2
              GO TO BS;                                                             07352000 T  0070:028:2
L27:L28:                                                                            07353000 T  0070:028:3
   COMMENT    INTEGER AND ALPHA ARRAYS;                                            07354000 T  0070:028:0
              ACLASS ← REALARRAYID;                                                 07355000 T  0070:028:0
L25:L26:                                                                            07356000 T  0070:028:1
   COMMENT    BOOLEAN AND REAL ARRAYS;                                             07357000 T  0070:028:2
              EMITPAIR(ADDRSF,LOD);                                                 07358000 T  0070:028:2
              IF SBIT THEN GO TO BS;                                                07359000 T  0070:029:0
   COMMENT    LOWER BOUNDS ARE NOT PASSED TO STREAM PROCEDURES;                     07360000 T  0070:029:2
              T1 ← TAKE(WHOLE ← GIT(WHOLE)).NODIMPART;                              07361000 T  0070:029:2
              FOR T2 ← 1 STEP 1 UNTIL T1                                            07362000 T  0070:029:3
                DO BEGIN                                                            07363000 T  0070:029:3
                IF T3 ← (STLB ← TAKE(WHOLE+T2)).[35:11] >1023                       07364000 T  0070:029:2
                   THEN EMITV(T3) ELSE EMIT(STLB);                                  07365000 T  0070:029:3
                   IF STLB.[23:10] = ADD THEN EMITO(CHS) END;                       07366000 T  0070:030:1
   COMMENT    THIS CODE EMITTED CALLS ON LOWER BOUNDS;                             07367000 T  0070:030:3
              IF FBIT THEN GO TO BS;                                                07368000 T  0070:030:3
              IF TAKE(INDEX+PCTR).INCR ≠ T1 THEN FLAG(124); GO TO BS;               07369000 T  0070:030:1
   COMMENT    ERROR IF ACTUAL AND FORMAL ARRAY DO NOT HAVE SAME NUMBER             07370000 T  0070:031:1
              OF DIMENSIONS;                                                        07371000 T  0070:031:1
L29:                                                                                07372000 T  0070:031:1
   COMMENT    LABEL;                                                               07373000 T  0070:031:2
              ELCLASS ← TABLE(I←I-1); DEXP; GO TO NSBS;                             07374000 T  0070:031:2
L30:                                                                                07375000 T  0070:031:4
   COMMENT    TRUTH VALUE;                                                         07376000 T  0070:031:5
              EMITL(ADDRSF); ACLASS ← BOOID; GO TO BSX;                             07377000 T  0070:031:5
L32:                                                                                07378000 T  0070:031:7
   COMMENT    LITERAL;                                                             07379000 T  0070:031:7
              EMITL(ADDRSF);                                                        07380000 T  0070:031:7
BSXX:         ACLASS ← REALID;                                                      07381000 T  0070:031:7
BSX:          IF SBIT AND NOT VBIT THEN FLAG(150); GO TO BS;                        07382000 T  0070:031:8
L31:L33:                                                                            07383000 T  0070:032:1
              EMITNUM(C); GO TO BSXX;                                               07384000 T  0070:032:2
```

```
                    EXIT;       STACKCT ← 0 END OF ACTUALPARAPART;                                %A          07385000 T  0070:032:2
                                                                                                        70 IS  329 LONG, NEXT SEG    3



                    COMMENT  PROCSTMT COMPILES CODE FOR ALL PROCEDURE STATEMENTS AND                      07386000 T  0003:081:0
                                 FUNCTION CALLS (EXCEPT FOR STREAM PROCEDURES).  THE                    07387000 T  0003:081:0
                                 PARAMETERS, FROM, TELLS WHO CALLED.  IF STMT CALLED FROM               07388000 T  0003:081:0
                                 IS TRUE.  PROCSTMT ALSO HANDLES FUNCTION NAME ASSIGNMENT               07389000 T  0003:081:0
                                 OPERATIONS;                                                            07390000 T  0003:081:0
                    PROCEDURE PROCSTMT(FROM); VALUE FROM; BOOLEAN FROM;                                 07391000 T  0003:081:0
                            BEGIN                                                                       07392000 T  0003:081:0
                                 REAL HOLE,ADDRESS;                                                     07393000 T  0003:081:0
                                                                                                  START OF SEGMENT **********   71
STACK(F+2) = HOLE
STACK(F+3) = ADDRESS
                                 LABEL EXIT;                                                            07394000 T  0071:000:0
                                 SCATTERELBAT;                                                          07395000 T  0071:000:0
                                 HOLE← ELBAT[1];                                                        07396000 T  0071:000:1
                                 ADDRESS ← ADDRSF;                                                      07397000 T  0071:000:3
                                 CHECKER(HOLE);                                                         07398000 T  0071:000:0
                                 IF ELCLASS ≠ PROCID THEN                                               07399000 T  0071:000:2
                                 IF NOT FORMALF THEN                                                    07400000 T  0071:000:3
                                 IF TABLE(I+1) = ASSIGNOP THEN                                          07401000 T  0071:000:1
                                      BEGIN VARIABLE(2-REAL(FROM)); GO TO EXIT END;                     07402000 T  0071:000:2
PRT(770) = *LIST, LABEL, OR SEGMENT DESCRIPTOR*
                              COMMENT  CALL VARIABLE TO HANDLE THIS ASSIGNMENT OPERATION;               07403000 T  0071:001:1
                                 IF ELCLASS ≠ PROCID EQV FROM                                           07404000 T  0071:001:1
                                      THEN BEGIN ERR(159); GO TO EXIT END;                              07405000 T  0071:001:2
                         COMMENT    IT IS PROCEDURE IF AND ONLY WE COME FROM STMT;                     07406000 T  0071:001:3
                                 STEPIT;                                                                07407000 T  0071:001:3
                                 EMITO(MKS);                                                            07408000 T  0071:001:0
                                 IF ELCLASS = LEFTPAREN                                                 07409000 T  0071:001:1
                                      THEN ACTUALPARAPART(FORMALF,FALSE,GIT(HOLE))                      07410000 T  0071:001:2
                                           ELSE IF FORMALF THEN                                         07411000 T  0071:001:2
                                                IF FROM THEN ELSE L←L-1                                 07411100 T  0071:002:1
                                        ELSE IF TAKE(GIT(HOLE)),NODIMPART≠0 THEN ERR(128);              07412000 T  0071:002:1
                                 EMITV(ADDRESS);                                                        07413000 T  0071:002:3
                         COMMENT    MONITOR CODE GOES HERE;                                            07414000 T  0071:002:0
                                 IF HOLE < 0                                                            07415000 T  0071:002:0
                                 THEN BEGIN COMMENT THIS IS A MONITORED FUNCTION DESIGNATOR              07416000 T  0071:002:1
                                           ;                                                            07417000 T  0071:002:3
                                           EMITL(JUNK);     EMITO(SND);     EMITO(MKS);                07418000 T  0071:002:3
                                           EMITV(JUNK);     EMITL(PASSTYPE(HOLE));                     07419000 T  0071:003:3
                                           EMITPAIR(GNAT(POWERSOFTEN),LOD);PASSALPHA(HOLE);           07420000 T  0071:003:3
                                           EMITPAIR(GNAT(CHARI ),LOD);     PASSMONFILE(TAKE           07421000 T  0071:003:3
                                          (GIT(HOLE)),FUNCMONFILE);                          %109-      07422000 P  0071:003:7
                                          EMITNUM(1&CARDNUMBER[1:4:44]);                     %109-      07422100 C  0071:003:9
                                           EMITV(GNAT(PRINTI));                                         07423000 T  0071:004:3
                                      END;                                                              07424000 T  0071:004:1
EXIT;    END  PROCSTMT;                                                                                 07425000 T  0071:004:1
                                                                                                  71 IS   46 LONG, NEXT SEG    3



COMMENT  STRMPROCSTMT COMPILES CODE FOR CALLS ON ALL STREAM PROCEDURES;                                 07426000 T  0003:081:0
```

```
                    PROCEDURE STRMPROCSTMT;                                                  07427000 T  0003:0814:0
                          BEGIN                                                              07428000 T  0003:0814:0
                               INTEGER ADDRS;                                                07429000 T  0003:0814:0
                                                                                  START OF SEGMENT **********   72
STACK(F+2) = ADDRS
                               IF ADDRS ← ELBAT[I].ADDRESS = 0                               07430000 T  0072:0000:0
                                 THEN BEGIN                                                  07431000 T  0072:0001:3
                                    UNKNOWNSTMT;                                           07432000 T  0072:0002:1
                               END                                                         07433000 T  0072:0003:2
                                                                                           07434000 T  0072:0003:2
                                                                                           07435000 T  0072:0003:2
                                                                                           07436000 T  0072:0003:2
                                                                                           07437000 T  0072:0003:2
                                                                                           07438000 T  0072:0003:2
                                                                                           07439000 T  0072:0003:2
                                 ELSE BEGIN                                                07440000 T  0072:0003:2
                               IF ELCLASS ≠ STRPROCID THEN EMIT(0); EMIT0(MKS); STEPIT;    07441000 T  0072:0003:3
                               GT1 ← (GT2 ← TAKE(GT3 ← GIT(ELBAT[I-1]))).[14:10];          07442000 T  0072:0006:1
                               GT4 ← GT1-GT2.[7:6];                                        07443000 T  0072:0010:1
                               FOR GT1 ← GT1-1 STEP -1 UNTIL GT4                           07444000 T  0072:0012:1
                                 DO EMITV(IF GT1 ≥ 512 THEN GT1+1024 ELSE GT1);            07445000 T  0072:0015:3
                          COMMENT  THIS CODE CALLS LABELS FROM PRT WHICH ARE NEEDED FOR LONG 07446000 T  0072:0022:0
                               JUMPS INSIDE OF STREAM PROCEDURES;                          07447000 T  0072:0022:0
                               GT4 ← GT2.[1:6];                                            07448000 T  0072:0022:0
                               FOR GT1 ← 1 STEP 1 UNTIL GT4 DO EMIT(0);                    07449000 T  0072:0023:2
                          COMMENT  THIS CODE CALLS ZERO LITS TO MAKE SPACE FOR LOCALS INSIDE 07450000 T  0072:0027:2
                               OF STREAM PROCEDURES;                                       07451000 T  0072:0027:2
                               IF ELCLASS ≠ LEFTPAREN THEN ERR(128)                        07452000 T  0072:0027:2
                                 ELSE BEGIN                                                07453000 T  0072:0028:1
                                   ACTUALPARAPART(FALSE,TRUE,GT3); EMITV(ADDRS) END;       07454000 T  0072:0029:3
                          END END STRMPROCSTMT;                                            07455000 T  0072:0031:3
                                                                                 72 IS   34 LONG, NEXT SEG    3


                     COMMENT  BAE BUILDS AN ACCIDENTAL ENTRY ( OR AT LEAST PREPARES FOR    07456000 T  0003:0814:0
                              ONE TO BE BUILT).  IT RETURNS VALUE OF L AT ENTRY;           07457000 T  0003:0814:0
                     INTEGER PROCEDURE BAE;                                                07458000 T  0003:0814:0
                              BEGIN BAE ← BUMPL; CONSTANTCLEAN; ADJUST END BAE;            07459000 T  0003:0814:0


                    COMMENT  RELSESTMT COMPILES THE RELEASE STATEMENT;                     07460000 T  0003:0819:2
                         RELEASE(<MEMORY DESIGNATOR>)          % AUXMEM RELEASE STATEMENT, 07460250 T  0003:0819:2
                         RELEASE(<TERMINAL BUFFER SPECIFIER>) % DATACOM RELEASE STATEMENT, 07460500 T  0003:0819:2
                         RELEASE(<FILE PART>)                  % FILE RELEASE STATEMENT,   07460750 T  0003:0819:2
                     ;                                                                     07461000 T  0003:0819:2
                    PROCEDURE RELSESTMT;                                                   07461250 T  0003:0819:2
PRT(771) = RELSESTMT
                        BEGIN                                                              07461500 T  0003:0819:2
                        LABEL DCR,PARENCHECK,EXIT;                                         07461750 T  0003:0819:2
                                                                                  START OF SEGMENT **********   73
                        IF STEPI≠LEFTPAREN THEN                                            07462000 T  0073:0000:0
                             BEGIN  ERR(105); GO EXIT  END;                                07462250 T  0073:0001:2
                        IF STEPI=UNKNOWNID THEN                                            07462500 T  0073:0002:1
```

```
          BEGIN  ERR(100); GO EXIT   END;                                                                07462750 T  0073:0003:3
     IF ELCLASS=PROCID OR RANGE(BOOPROCID,INTPROCID) THEN                                               07463000 T  0073:0005:3
          BEGIN                                                                                         07463250 T  0073:0007:3
          EMITPAIR(ELBAT[I],ADDRESS,LOD); EMITPAIR(38,COM);                                             07463500 T  0073:0008:0
          EMITO(DEL); STEPIT; GO PARENCHECK;                                                            07463750 T  0073:0010:1
          END;                                                                                          07464000 T  0073:0012:1
     IF RANGE(BOOARRAYID,INTARRAYID) THEN                                                               07464250 T  0073:0012:1
          BEGIN                                                                                         07464500 T  0073:0013:3
          REL:=TRUE;AEXP; REL:=FALSE;                                                                   07464750 T  0073:0014:0
          IF TABLE(I-2) = FACTOP THEN                                                                   07465000 T  0073:0016:0
          BEGIN STACKCT:=STACKCT-1;                                                                     07465250 T  0073:0017:3
               EMITPAIR(38,COM); EMITO(DEL); GO PARENCHECK;                                             07465500 T  0073:0019:3
               END                                                                                      07465750 T  0073:0021:3
        ELSE  BEGIN % DATACOM RELEASE.                                                                  07466000 T  0073:0021:3
DCR;                                                                                                    07466250 T  0073:0022:0
               EMITL(2); EMITO(XCH); EMITL(0); EMITO(XCH);                                              07466500 T  0073:0023:2
               EMITL(0); EMITPAIR(32,COM); EMITO(DEL);                                                  07466750 T  0073:0026:0
               EMITO(DEL); EMITO(DEL); EMITO(DEL); GO PARENCHECK;                                       07467000 T  0073:0028:1
               END;                                                                                     07467250 T  0073:0031:2
          END;                                                                                          07467500 T  0073:0031:2
     IF ELCLASS≠FILEID AND ELCLASS≠SUPERFILEID THEN % DATACOM RELEASE.                                  07467750 T  0073:0031:2
          BEGIN AEXP; GO DCR; END;                                                                      07468000 T  0073:0033:2
     CHECKER(ELBAT[I]); PASSFILE;                                                                       07468250 T  0073:0034:1
     IF ELCLASS = COMMA THEN EMITO(DUP);                                                                07468500 T  0073:0036:0
COMMENT  THIS WILL FETCH DESCRIPTOR POINTING TO I/O DESCRIPTOR;                                         07468750 T  0073:0038:0
     CHECKPRESENCE;                                                                                     07469000 T  0073:0038:0
COMMENT  THIS WILL CAUSE PRESENCE BIT INTERRUPT IF PREVIOUS I/O IS                                      07469250 T  0073:0038:1
          NOT COMPLETED;                                                                                07469500 T  0073:0038:1
     EMITO(DUP); EMITO(LOD); EMITO(XCH);                                                                07469750 T  0073:0038:1
     IF ELCLASS = COMMA THEN                                                                            07470000 T  0073:0040:1
          BEGIN                                                                                         07470250 T  0073:0041:3
          EMITO(DUP); EMITO(LOD); STEPIT; AEXP;                                                         07470500 T  0073:0042:0
          EMITD(38,8,10); EMITO(XCH); EMITO(STD); EMITO(XCH);                                           07470750 T  0073:0044:1
          END;                                                                                          07471000 T  0073:0048:0
     EMITO(PRL); EMITO(DEL);                                                                            07471250 T  0073:0048:0
PARENCHECK;                                                                                             07471500 T  0073:0049:3
     IF ELCLASS=RTPAREN THEN STEPIT ELSE ERR(104);                                                      07471750 T  0073:0050:0
EXIT;                                                                                                   07472000 T  0073:0053:2
     END RELSESTMT;                                                                                     07472250 T  0073:0053:2
                                                                                                  73 IS   54 LONG, NEXT SEG    3


                    COMMENT  DOSTMT HANDLES THE DO STATEMENT;                                           07481000 T  0003:0819:2
                    PROCEDURE DOSTMT;                                                                   07482000 T  0003:0819:2
PRT(772) = DOSTMT
                         BEGIN INTEGER TL;                                                              07483000 T  0003:0819:2
                                                                                               START OF SEGMENT **********   74
STACK(F+2) = TL
                              DIALA ← DIALB ← 0;                                                        07484000 T  0074:0000:0
                           ADJUST;                                                                      07484100 T  0074:0001:2
                           STEPIT; TL←L; STMT; IF ELCLASS ≠ UNTILV THEN ERR(131)                        07485000 T  0074:0001:3
                           ELSE BEGIN                                                                   07486000 T  0074:0005:2
                             STEPIT; BEXP; EMITB(BBC,BUMPL,TL) END                                      07487000 T  0074:0006:0
                         END DOSTMT;                                                                    07488000 T  0074:0009:2
                                                                                                  74 IS   12 LONG, NEXT SEG    3
```

```
                COMMENT  WHILESTMT COMPILES THE WHILE STATEMENT;                                  07489000 T  0003:081912
                PROCEDURE WHILESTMT;                                                             07490000 T  0003:081912
PRT(773) = WHILESTMT
                    BEGIN INTEGER BACK,FRONT;                                                    07491000 T  0003:081912
                                                                                  START OF SEGMENT **********   75
STACK(F+2) = BACK
STACK(F+3) = FRONT
                        DIALA = DIALB = 0;                                                       07492000 T  0075:000010
                        ADJUST;                                                                  07492100 T  0075:000112
                        STEPIT; BACK = L; BEXP; FRONT = BUMPL;                                   07493000 T  0075:000013
                        IF ELCLASS # DOV THEN ERR(132) ELSE                                      07494000 T  0075:000512
                          BEGIN STEPIT; STMT; EMITB(BBW,BUMPL,BACK);                             07495000 T  0075:000712
                             CONSTANTCLEAN; EMITB(BFC,FRONT,L) END END WHILESTMT;                07496000 T  0075:001112
                                                                                    75 IS   15 LONG, NEXT SEG    3



                COMMENT  GOSTMT COMPILES GO TO STATEMENTS.  GOSTMT LOOKS AT THE                  07497000 T  0003:081912
                         EXPRESSION.  IF IT IS SIMPLE ENOUGH WE GO DIRECTLY,                     07498000 T  0003:081912
                         OTHERWISE A CALL ON THE MCP IS GENERATED IN ORDER TO GET                07499000 T  0003:081912
                         STORAGE RETURNED.  SEE DEXP AND GENGO;                                  07500000 T  0003:081912
                PROCEDURE GOSTMT;                                                                07501000 T  0003:081912
PRT(774) = GOSTMT
                    BEGIN                                                                        07502000 T  0003:081912
                        REAL ELBW;                                                               07503000 T  0003:081912
                                                                                  START OF SEGMENT **********   76
STACK(F+2) = ELBW
                        LABEL GOMCP,EXIT;                                                        07504000 T  0076:000010
                        IF STEPI = TOV THEN STEPIT;                                              07505000 T  0076:000010
                        IF ELCLASS = LABELID THEN TB1 = TRUE                                     07506000 T  0076:000210
                        ELSE IF ELCLASS = SWITCHID THEN TB1 = FALSE ELSE GO GOMCP;               07507000 T  0076:000313
                        IF NOT LOCAL(ELBAT[I]) THEN GO GOMCP;                                    07508000 T  0076:000611
                        IF TB1 THEN BEGIN GOGEN(ELBAT[I],BFW); STEPIT;                           07509000 T  0076:000810
                        CONSTANTCLEAN; GO EXIT END;                                              07510000 T  0076:001011
                        ELBW = ELBAT[I];                                                         07511000 T  0076:001113
                        IF ELBW < 0                                                              07512000 T  0076:001211
                        THEN BEGIN COMMENT THIS IS A MONITORED SWITCH;                           07513000 T  0076:001211
                                  EMITO(MKS);     PASSALPHA(ELBW);EMITPAIR(GNAT(                 07514000 T  0076:001313
                                  CHARI),LOD);    PASSMONFILE(TAKE(GIT(ELBW)),                   07515000 T  0076:001513
                                 SWITMONFILE);                                        %109-      07516000 P  0076:001810
                                 EMITNUM(O&CARDNUMBER[1:4:44]);                       %109-      07516100 C  0076:001912
                                 EMITV(GNAT(                                          %109-      07516200 C  0076:002011
                                  PRINT));                                                       07517000 T  0076:002112
                             END;                                                                07518000 T  0076:002210
                        BANA; EMITPAIR(JUNK,ISD);                                                07519000 T  0076:002210
                        IF (GT1 = TAKE(GT2 = GIT(ELBW))),[24:12] = 0                             07520000 T  0076:002313
                             AND ELBW.ADDRESS = 0 THEN BEGIN                                     07521000 T  0076:002610
                             PUT(GT1&(BUMPL)[24:36:12],GT2);                                     07522000 T  0076:002811
                             EMITB(BBW,L,GT4+GT1,[36:12]);                                       07523000 T  0076:003113
                             EMITB(BFW,GT4+13,L+3);                                              07524000 T  0076:003410
                             EMITO(NOP); EMITO(NOP); EMITO(NOP) END                              07525000 T  0076:003610
                          ELSE BEGIN CALLSWITCH(ELBW); EMITO(BFW) END;                           07526000 T  0076:003811
                        GO EXIT;                                                                 07527000 T  0076:004011
                GOMCP:  GOTOG = FALSE; DEXP;                                                     07528000 T  0076:004112
```

```
            IF GOTOG THEN                                                          07529000 T  0076:0042:0
                 BEGIN EMITO(MKS); EMITL(9); EMITV(5); EMITO(BFW) END              07529100 T  0076:0042:1
               ELSE BEGIN EMITO(PRTE); EMITO(LOD); EMITO(BFW) END;                 07529200 T  0076:0046:0
   EXIT:END GOSTMT;                                                                07530000 T  0076:0048:1
                                                                          76 IS   52 LONG, NEXT SEG    3



COMMENT GOGEN GENERATES CODE TO GO TO A LABEL, GIVEN THAT LABEL AS A               07531000 T  0003:0819:2
            PARAMETER.  GOGEN ASSUMES THAT THE  LABEL IS LOCAL.  THE               07532000 T  0003:0819:2
            PARAMETER BRANCH TYPE TELL WHETHER THE JUMP IS CONDITIONAL             07533000 T  0003:0819:2
            OR NOT;                                                                07534000 T  0003:0819:2
PROCEDURE GOGEN(LABELBAT,BRANCHTYPE);                                              07535000 T  0003:0819:2
            VALUE LABELBAT,BRANCHTYPE;                                             07536000 T  0003:0819:2
            REAL LABELBAT,BRANCHTYPE;                                              07537000 T  0003:0819:2
            BEGIN                                                                  07538000 T  0003:0819:2
              IF BOOLEAN(GT1←TAKE(GT2←GIT(LABELBAT))),[1:1]                        07539000 T  0003:0819:2
                THEN EMITB(BRANCHTYPE,BUMPL,GT1,[36:12])                           07540000 T  0003:0822:1
   COMMENT  LABELR SETS THE SIGN OF THE ADDITIONAL INFO FOR A LABEL                07541000 T  0003:0825:2
            NEGATIVE WHEN THE LABEL IS ENCOUNTERED,  SO THIS MEANS                 07542000 T  0003:0825:3
            THAT WE NOW KNOW WHERE TO GO;                                          07543000 T  0003:0825:3
                 ELSE BEGIN EMIT(GT1); EMIT(BRANCHTYPE);                           07544000 T  0003:0825:3
                 PUT(GT1&L[36:36:12],GT2) END END GOGEN;                           07545000 T  0003:0828:0



COMMENT SIMPGO IS USED ONLY BY THE IF STMT ROUTINE.  IT DETERMINES IF              07546000 T  0003:0830:0
            A STATEMENT IS A SIMPLE GO TO STATEMENT;                               07547000 T  0003:0830:0
BOOLEAN PROCEDURE SIMPGO;                                                          07548000 T  0003:0830:0
        BEGIN LABEL EXIT;                                                          07549000 T  0003:0830:0
                                                                        START OF SEGMENT **********   77
             IF ELCLASS = GOV                                                      07550000 T  0077:0000:0
               THEN BEGIN                                                          07551000 T  0077:0000:0
                  IF STEPI = TOV THEN STEPIT;                                      07552000 T  0077:0001:2
                  IF ELCLASS = LABELID THEN                                        07553000 T  0077:0003:2
                  IF LOCAL(ELBAT[I]) THEN                                          07554000 T  0077:0004:0
                     BEGIN SIMPGO ← TRUE; GO EXIT END;                             07555000 T  0077:0005:3
                       I ← I-1; ELCLASS ← GOV END;                                 07556000 T  0077:0007:2
 EXIT:  END SIMPGO;                                                                07557000 T  0077:0009:2
                                                                         77 IS   13 LONG, NEXT SEG    3



COMMENT IFSTMT COMPILES IF STATEMENTS,  SPECIAL CARE IS TAKEN TO                   07558000 T  0003:0830:0
            OPTIMIZE CODE IN THE NEIGHBORHOOD OF THE JUMPS,  TO SOME               07559000 T  0003:0830:0
            EXTENT SUPPERFULOUS BRANCHING IS AVOIDED;                              07560000 T  0003:0830:0
PROCEDURE IFSTMT;                                                                  07561000 T  0003:0830:0
        BEGIN REAL T1,T2; LABEL EXIT;                                              07562000 T  0003:0830:0
                                                                        START OF SEGMENT **********   78
STACK(F+2) = T1
STACK(F+3) = T2
             IFCLAUSE;                                                             07563000 T  0078:0000:0
             IF SIMPGO                                                             07564000 T  0078:0000:1
               THEN BEGIN                                                          07565000 T  0078:0000:1
                  T1 ← ELBAT[I];                                                   07566000 T  0078:0001:3
```

```
                IF STEPI = ELSEV                                                    07567000 T  0078:0002:1
                      THEN BEGIN                                                    07568000 T  0078:0003:2
                         STEPIT;                                                    07569000 T  0078:0004:0
                         IF SIMPGO                                                  07570000 T  0078:0004:1
                           THEN BEGIN                                               07571000 T  0078:0004:1
                             GOGEN(ELBAT[I],BFC); GOGEN(T1,BFW);                    07572000 T  0078:0005:3
                STEPIT; GO TO EXIT END ELSE BEGIN EMITLNG;GOGEN(T1,BFC);            07573000 T  0078:0007:3
                     STMT ; GO TO EXIT  END END ;                                   07574000 T  0078:0010:1
                     EMITLNG; GOGEN(T1,BFC);                                        07575000 T  0078:0011:3
                     GO EXIT END;                                                   07576000 T  0078:0013:2
                T1 ← BUMPL; STMT;                                                   07577000 T  0078:0013:3
                IF ELCLASS ≠ ELSEV THEN                                             07578000 T  0078:0016:0
                  BEGIN DIALA ← DIALB ← 0; EMITB(BFC,T1,L); GO EXIT END;            07579000 T  0078:0016:1
                STEPIT;                                                             07580000 T  0078:0020:0
                IF SIMPGO                                                           07581000 T  0078:0020:1
                  THEN BEGIN                                                        07582000 T  0078:0020:1
                     T2 ← L; L ←T1-2;GOGEN(ELBAT[I],BFC); L ← T2;                   07583000 T  0078:0021:3
                     STEPIT; GO EXIT END;                                           07584000 T  0078:0025:3
                T2 ← BUMPL; CONSTANTCLEAN;                                          07585000 T  0078:0026:1
                EMITB(BFC,T1,L); STMT; EMITB(BFW,T2,L);                             07586000 T  0078:0029:2
EXIT;  END  IFSTMT;                                                                 07587000 T  0078:0032:0
                                                                        78 IS   35 LONG, NEXT SEG    3


COMMENT  LABELR HANDLES LABELED STATEMENTS.  IT PUTS L INTO THE                     07588000 T  0003:0830:0
              ADDITIONAL INFO AND MAKES ITS SIGN NEGATIVE.  IT COMPILES             07589000 T  0003:0830:0
              AT THE SAME TIME ALL THE PREVIOUS FORWARD REFERENCES SET              07590000 T  0003:0830:0
              UP FOR IT BY GOGEN.  (THE ADDITIONAL INFO LINKS TO A LIST             07591000 T  0003:0830:0
              IN THE CODE ARRAY OF ALL FORWARD REFERENCES);                         07592000 T  0003:0830:0
          PROCEDURE LABELR;                                                         07593000 T  0003:0830:0
PRT(775) = LABELR
                    BEGIN LABEL EXIT, ROUND;                                        07594000 T  0003:0830:0
                                                                      START OF SEGMENT **********   79
          DEFINE ELBATWORD=RR9#,LINK=GT2#,INDEX=GT3#,ADDITIONAL                     07595000 T  0079:0000:0
                      =GT4#,NEXTLINK=GT5#;                                          07596000 T  0079:0000:0
                    DO BEGIN ADJUST; IF STEPI ≠ COLON THEN                          07597000 T  0079:0000:0
                        BEGIN ERR(133); GO TO EXIT END;                             07598000 T  0079:0001:3
                    XMARK(LBLREF); % THIS WILL SORT AHEAD OF DECLARATION %116-      07598100 C  0079:0003:2
                                   % WHEN WE GET AROUND TO THE XREF.     %116-      07598200 C  0079:0007:3
                         IF NOT LOCAL(ELBATWORD ← ELBAT[I-1])                       07599000 T  0079:0007:3
                           THEN BEGIN FLAG(134); GO TO ROUND END;                   07600000 T  0079:0008:1
                         LINK ← (ADDITIONAL ← TAKE(INDEX ← GIT(ELBATWORD)))         07601000 T  0079:0011:3
                              .[36:12];                                             07602000 T  0079:0013:2
                         IF ADDITIONAL < 0 THEN                                     07603000 T  0079:0014:1
                           BEGIN FLAG(135); GO TO ROUND END;                        07604000 T  0079:0015:3
                         WHILE LINK ≠ 0                                             07605000 T  0079:0017:2
                           DO BEGIN                                                 07606000 T  0079:0018:0
                              NEXTLINK ← GET(LINK-2);                               07607000 T  0079:0019:2
                             EMITB(GET(LINK-1),LINK,L);                             07608000 T  0079:0021:2
                         LINK ← NEXTLINK;                                           07609000 T  0079:0023:2
                         IF LASTENTRY ≥ 126 THEN % DONT LET EMITNUM DO IT           07609100 T  0079:0024:0
                         BEGIN REAL C; % HOLD L FOR A WHILE                         07609200 T  0079:0024:1
PRT(776) = *SEGMENT DESCRIPTOR*
                                                                      START OF SEGMENT **********   80
          STACK(F+2) = C
```

```
                        COMMENT THIS IS TO ALLOW FOR MORE THAN 56 LONG          07609300 T  0080:000:0
                             (>1023 WORD) FORWARD REFERENCES TO A LABEL;        07609400 T  0080:000:0
                        C + BUMPL;                                              07609500 T  0080:000:0
                        CONSTANTCLEAN;                                          07609600 T  0080:000:3
                        EMITB(BFW,C,L) END;END;                                 07609700 T  0080:000:0
PRT(777) = *SEGMENT DESCRIPTOR*

                                                                        80 IS     4 LONG, NEXT SEG   79
                    PUT(=ADDITIONAL&L[36:36:12],INDEX);                         07610000 T  0079:002:1
                IF ELBATWORD < 0                                                07611000 T  0079:002:1
                THEN BEGIN COMMENT THIS LABEL IS EITHER APPEARS IN A DUMP       07612000 T  0079:002:2
                                OR MONITOR DECLARATION;                         07613000 T  0079:003:0
                        IF RR1=ADDITIONAL.LABLMONFILE ≠ 0                       07614000 T  0079:003:0
                        THEN BEGIN COMMENT THIS CODE IS FOR MONITORED           07615000 T  0079:003:2
                                LABELS;                                         07616000 T  0079:003:0
                                EMITO(MKS); PASSALPHA(ELBATWORD);               07617000 T  0079:003:0
                                EMITPAIR(GNAT(CHARI),LOD);                      07618000 T  0079:003:3
                                PASSMONFILE(RR1);                       %109-   07619000 P  0079:003:2
                                EMITNUM(0&CARDNUMBER[1:4:44]);          %109-   07619100 C  0079:003:0
                                EMITV(GNAT(PRINTI));                            07620000 T  0079:003:3
                            END;                                                07621000 T  0079:003:2
                        IF RR1=ADDITIONAL.DUMPEE ≠ 0                            07622000 T  0079:003:2
                        THEN BEGIN COMMENT EMIT CODE TO INCREMENT THE           07623000 T  0079:004:0
                                LABEL COUNTER;                                  07624000 T  0079:004:2
                                EMITV(RR1); EMITL(1);   EMITO(ADD);             07625000 T  0079:004:2
                                EMITPAIR (RR1,STD);                             07626000 T  0079:004:3
                                IF RR1=ADDITIONAL.DUMPOR ≠ 0                    07627000 T  0079:004:1
                                THEN BEGIN COMMENT EMIT CODE TO CALL            07628000 T  0079:004:3
                                        THE DUMP ROUTINE;                       07629000 T  0079:004:1
                                                                                07630000 T  0079:004:1
                                                                                07631000 T  0079:004:1
                                                                                07632000 T  0079:004:1
                                                STUFFF(RR1); EMITO              07633000 T  0079:004:1
                                                (XCH);EMITO(COC);               07634000 T  0079:004:3
                                                                                07635000 T  0079:004:2
                                                                                07636000 T  0079:004:2
                                                                                07637000 T  0079:004:2
                                                                                07638000 T  0079:004:2
                                                                                07639000 T  0079:004:2
                                                                                07640000 T  0079:004:2
                                        EMITO(DEL);                             07641000 T  0079:004:2
                                    END;                                        07642000 T  0079:004:3
                                END;                                            07643000 T  0079:004:3
                            END;                                                07644000 T  0079:004:3
                    END;                                                        07645000 T  0079:004:3
ROUND;      ERRORTOG + TRUE END UNTIL STEPI ≠ LABELID;                          07646000 T  0079:005:2:0
  EXIT; END LABELR;                                                       79 IS   56 LONG, NEXT SEG    3


PROCEDURE CASESTMT;                                                             07646100 T  0003:083:0:0
BEGIN COMMENT  THE CASE STATEMENT HAS THE FOLLOWING FORM:                       07646110 T  0003:083:0:0
            CASE <ARITH EXP> OF BEGIN <COMPOUND TAIL>                           07646120 T  0003:083:0:0
   AT EXECUTION THE CASE STATEMENT SELECTS ONE OF THE STATEMENTS                07646130 T  0003:083:0:0
   IN THE <COMPOUND TAIL>, DEPENDING ON THE VALUE OF THE <ARITH EXP>.           07646140 T  0003:083:0:0
   ONLY THE SELECTED STATEMENT IS EXECUTED AND CONTROL RESUMES AFTER            07646150 T  0003:083:0:0
   THE <COMPOUND TAIL>.  IF THERE ARE N STATEMENTS IN THE                       07646160 T  0003:083:0:0
   <COMPOUND TAIL>, THEY MAY BE CONSIDERED NUMBERED 0,1,...,N-1.                07646170 T  0003:083:0:0
```

```
AND THE <ARITH EXP> MUST TAKE ON ONLY THESE VALUES.  OTHER VALUES
WILL RESULT IN AN INVALID INDEX TERMINATION OF THE OBJECT PROGRAM.
THE STATEMENTS IN THE <COMPOUND TAIL> MAY BE ANY EXECUTABLE
STATEMENTS, INCLUDING COMPOUND STATEMENTS, BLOCKS, CASE STATEMENTS
AND NULL STATEMENTS.   THE CODE GENERATED IS AS FOLLOWS:
          <ARITH EXP>
          OPDC ARRAY
          BFW
          STMT 0
          BRANCH TO RESUME
          STMT 1
          BRANCH TO RESUME
               .
               .
               .
          STMT N-1
     RESUME:
"ARRAY" IS COMPILED AS A TYPE-2 SEGMENT OF N WORDS AND IS
CHANGED TO A DATA ARRAY AT THE FIRST REFERENCE.  IT IS SUBSCRIPTED
BY THE VALUE OF <ARITH EXP> AND CONTAINS SYLLABLE COUNTS
FOR THE BRANCH TO EACH OF THE N STATEMENTS.  THE BRANCH TO RESUME
IS OMITTED FOR A NULL STATEMENT.  INSTEAD, THE INITIAL BRANCH
TRANSFERS TO RESUME DIRECTLY:
REAL LINK, TEMP, N, ADR, PRT, NULL;                                  07646180 T  0003:083010
                                                                     07646190 T  0003:083010
                                                                     07646200 T  0003:083010
                                                                     07646210 T  0003:083010
                                                                     07646220 T  0003:083010
                                                                     07646230 T  0003:083010
                                                                     07646240 T  0003:083010
                                                                     07646250 T  0003:083010
                                                                     07646260 T  0003:083010
                                                                     07646270 T  0003:083010
                                                                     07646280 T  0003:083010
                                                                     07646290 T  0003:083010
                                                                     07646300 T  0003:083010
                                                                     07646310 T  0003:083010
                                                                     07646320 T  0003:083010
                                                                     07646330 T  0003:083010
                                                                     07646340 T  0003:083010
                                                                     07646350 T  0003:083010
                                                                     07646360 T  0003:083010
                                                                     07646370 T  0003:083010
                                                                     07646375 T  0003:083010
                                                                     07646380 T  0003:083010
                                                                     07646385 T  0003:083010
                                                                     07646390 T  0003:083010
                                                          START OF SEGMENT **********    81
STACK(F+2) = LINK
STACK(F+3) = TEMP
STACK(F+4) = N
STACK(F+5) = ADR
STACK(F+6) = PRT
STACK(F+7) = NULL
                    BOOLEAN GOTOG;                                   07646395 T  0081:000010
STACK(F+10) = GOTOG
                    REAL ARRAY TEDOC[0:7, 0:127];                    07646400 T  0081:000010
STACK(F+11) = TEDOC
                    LABEL LOOP, XIT;                                 07646410 T  0081:000210
                    LINK ← N ← NULL ← 0;                             07646420 T  0081:000210
                    STEP;T; AEXP;                                    07646430 T  0081:000313
                    IF STEP ≠ BEGINV THEN BEGIN ERR( 70); GO TO XIT END;  07646440 T  0081:000411
                    EMITV(PRT:=GETSPACE(TRUE,-3)); % CASE STMNT, DESCR,  07646450 T  0081:000713
                    EMITO(BFW); ADR ← L;                             07646460 T  0081:000913
                LOOP:                                                07646470 T  0081:001112
                    ERRORTOG ← TRUE;                                 07646475 T  0081:001210
                    IF STEP = SEMICOLON THEN                         07646480 T  0081:001211
                    BEGIN COMMENT NULL STATEMENT;                    07646485 T  0081:001313
                       TEDOC[N,[38:3], N,[41:7]] ← NULL;             07646490 T  0081:001410
                       NULL ← N ← N+1; GO TO LOOP;                   07646495 T  0081:001712
                    END;                                             07646500 T  0081:001913
                    TEDOC[N,[38:3], N,[41:7]] ← L-ADR;  N ← N + 1;   07646510 T  0081:001913
                    IF GOTOG ≠ SIMPGO THEN ELBAT[I←I-1] ← ELCLASS ← GOV;  07646515 T  0081:002410
                    STMT;                                            07646520 T  0081:002811
                    IF ELCLASS = SEMICOLON THEN                      07646525 T  0081:002912
                    BEGIN IF NOT GOTOG THEN                          07646530 T  0081:002913
                          BEGIN EMIT(LINK);LINK ← L ← L+1; END;      07646533 T  0081:003011
                          GO TO LOOP;                                07646535 T  0081:003313
                    END ELSE IF ELCLASS = ENDV THEN                  07646538 T  0081:003410
                    BEGIN IF NOT GOTOG THEN                          07646540 T  0081:003513
```

```
      BEGIN EMIT(LINK); LINK ← L ← L+1; END;                                                07646543 T  0081:0036:1
      TEDOC[N,[38:3], N,[41:7]]← L-ADR;                                                     07646545 T  0081:0039:3
      N ← N+1;                                                                              07646548 T  0081:0043:2
END;                                                                                        07646550 T  0081:0044:0
IF ELCLASS ≠ ENDV THEN BEGIN ERR( 71); GO TO LOOP END;                                      07646555 T  0081:0044:0
N := N-1  ;                                                                                 07646556 T  0081:0046:1
WHILE NULL ≠ 0 DO                                                                           07646560 T  0081:0048:0
BEGIN TEMP ← TEDOC[(NULL←NULL-1),[38:3], NULL,[41:7]];                                      07646565 T  0081:0049:2
   TEDOC[NULL,[38:3], NULL,[41:7]] ← L-ADR;                                                 07646570 T  0081:0053:2
   NULL ← TEMP;                                                                             07646575 T  0081:0056:1
END;                                                                                        07646580 T  0081:0057:2
ENDTOG ← TRUE;                                                                              07646585 T  0081:0057:3
COMMENT  SKIP ANY COMMENTS AFTER "END";                                                     07646590 T  0081:0058:1
DO STOPDEFINE ← TRUE UNTIL STEPI ≤ ENDV AND ELCLASS ≥ UNTILV                                07646595 T  0081:0058:1
   OR NOT ENDTOG;                                                                           07646600 T  0081:0061:2
ENDTOG ← FALSE;                                                                             07646610 T  0081:0063:2
COMMENT  DEFINE TEDOC AS TYPE-2 SEGMENT;                                                    07646620 T  0081:0063:3
MOVECODE(TEDOC, EDOC);                                                                      07646630 T  0081:0063:3
BUILDLINE ← BOOLEAN(2×REAL(BUILDLINE)) ;                                                    07646635 T  0081:0066:0
TEMP := SGNO; IF LISTER OR SEGSTOG THEN SEGMENTSTART;                                       07646640 T  0081:0067:2
SGNO ← SGAVL;                                                                               07646650 T  0081:0070:0
Z ← PROGDESCBLDR(LDES, 0, PRT);                                                             07646660 T  0081:0071:2
SEGMENT(-N, SGNO, TEMP);                                                                    07646670 T  0081:0072:1
SGAVL ← SGAVL + 1; SGNO ← TEMP;                                                             07646680 T  0081:0074:0
BUILDLINE ← BUILDLINE.[46:1] ;                                                              07646685 T  0081:0076:0
MOVECODE(TEDOC, EDOC);                                                                      07646690 T  0081:0077:3
COMMENT  FIX UP BRANCHES TO RESUME POINT;                                                   07646700 T  0081:0079:3
IF (L-ADR)>1019 THEN ADJUST;%                                                               07646705 T  0081:0079:3
WHILE LINK ≠ 0 DO                                                                           07646710 T  0081:0082:0
BEGIN TEMP ← GET(LINK-2);                                                                   07646720 T  0081:0083:2
   EMITB(BFW, LINK, L);                                                                     07646730 T  0081:0085:2
   LINK ← TEMP;                                                                             07646740 T  0081:0086:0
   IF LASTENTRY ≥ 126 THEN                                                                  07646750 T  0081:0087:2
   BEGIN REAL C;                                                                            07646760 T  0081:0087:3
PRT(1000) = *SEGMENT DESCRIPTOR*

STACK(F+12) = C                                                                 START OF SEGMENT **********   82
                  COMMENT  PERMITS SEVERAL LONG BRANCHES IF NECESSARY;                      07646770 T  0082:0000:0
                  C ← BUMPL;                                                                07646780 T  0082:0000:0
                  CONSTANTCLEAN;                                                            07646790 T  0082:0001:3
                  EMITB(BFW, C, L);                                                         07646800 T  0082:0002:0
                  END;                                                                      07646810 T  0082:0003:3
PRT(1001) = *SEGMENT DESCRIPTOR*

                  END;                                                           82 IS    5 LONG, NEXT SEG   81
               XIT;                                                                     07646820 T  0081:0089:2
               END CASESTMT;                                                            07646830 T  0081:0089:3
                                                                                        07646840 T  0081:0090:0
                                                                                81 IS   96 LONG, NEXT SEG    3



               COMMENT  THE FOLLOWING PROCEDURE HANDLES THE FILL STATEMENT,             07647000 T  0003:0830:0
                        IT EMITS CODE TO PASS THE ROW TO BE FILLED AND TO PASS          07647500 T  0003:0830:0
                        THE INDEX IN THE SEGMENT DICTIONARY OF THE FILL SEGMENT.        07648000 T  0003:0830:0
                        THESE SEGMENTS LOOK LIKE ANY OTHER SEGMENT TO THE MCP.          07648500 T  0003:0830:0
                        NO FILL SEGMENT IS EVER BROUGHT INTO CORE.THE SEGMENT           07649000 T  0003:0830:0
```

```
                    RESIDES ON THE DISK AND IS READ INTO THE ROW DESIGNATED                07649500 T  0003!083!0
                    BY THE FILL STATEMENT EVERY TIME THE FILL STATEMENT IS                 07650000 T  0003!083!0
                    EXECUTED,STRINGCONSTANTS,LITERAL ,AND NONLITERAL NUMBERS               07650500 T  0003!083!0
                    ARE ALL CONVERTED BY THE SCANNER AND NUMBER BUILDER,OCTAL              07651000 T  0003!083!0
                    NUMBERS LOOK LIKE IDENTIFIERS TO FILLSTMT AND ARE CONVERTED            07651500 T  0003!083!0
                    BY OCTIZE.AFTER BUILDING THE SEGMENT AN ENTRY IS MADE IN               07652000 T  0003!083!0
                    PDPRT TO SUPPLY INFO TO BUILD A DISK DESCRIPTOR IN THE                 07652500 T  0003!083!0
                    SEGMENT DICTIONARY.THE COMMUNICATE LITERAL IS 7;                       07653000 T  0003!083!0
          PROCEDURE FILLSTMT;                                                              07653500 T  0003!083!0
PRT(1002) = FILLSTMT
            BEGIN                                                                          07654000 T  0003!083!0
            LABEL EXIT;                                                                    07654500 T  0003!083!0
                                                                                  START OF SEGMENT **********   83
            DEFINE    PARENCOUNTER = RR1#,                                                 07655000 T  0083!000!0
                      T            = RR2#,                                                 07655500 T  0083!000!0
                      J            = RR3#;                                                 07656000 T  0083!000!0
            ARRAY     TEDOC[0:7,0:127], FILLTEMP[0:1022];                                  07656500 T  0083!000!0
STACK(F+2) = TEDOC
STACK(F+3) = FILLTEMP
            BOOLEAN PROCEDURE FILLIT(A); ARRAY A[0];                                     07657000 T  0083!000!3
PRT(1003) = FILLIT
                BEGIN                                                                    07657500 T  0083!000!3
                REAL     T1, T2, T3;                                                     07658000 T  0083!000!3
                                                                                  START OF SEGMENT **********   84
STACK(F+3) = T1
STACK(F+4) = T2
STACK(F+5) = T3
                BOOLEAN   BOO;                                                           07658500 T  0084!000!0
STACK(F+6) = BOO
LABEL     CHECK, GOOFUP, EXIT;                                                           07659000 T  0084!000!0
PARENCOUNTER!=PARENCOUNTER+1;                                                            07659500 T  0084!000!0
WHILE T<1023 DO                                                                          07660000 T  0084!000!2
     BEGIN                                                                               07660500 T  0084!000!2
     IF STEPI>IDMAX THEN                                                                 07661000 T  0084!000!2
          BEGIN                                                                          07661500 T  0084!000!0
          IF ELCLASS=LITNO THEN                                                          07662000 T  0084!000!1
            IF TABLE(I+1)=LEFTPAREN THEN                                                 07662500 T  0084!000!3
               BEGIN                                                                     07663000 T  0084!000!3
               T1!=ELBAT[I].ADDRESS; T2!=T;                                              07663500 T  0084!000!0
               STEPIT; IF FILLIT(A) THEN GO GOOFUP;                                      07664000 T  0084!001!1
               IF T1=0 THEN T!=T2                                                        07664500 T  0084!001!2
               ELSE BEGIN                                                                07665000 T  0084!001!1
                    IF (T3!=(T1-1)x(T-T2))+T>1022 THEN                                   07665500 T  0084!001!3
                         BEGIN ERROR(305); GO GOOFUP END;%>1023                          07666000 T  0084!001!2
                    MOVE(T3,A[T2],A[T]); T!=T+T3;                                        07666500 T  0084!002!1
                    END;                                                                 07667000 T  0084!002!0
               GO CHECK;                                                                 07667500 T  0084!002!0
               END REPEAT PART;                                                          07668000 T  0084!002!1
          IF (BOO!=ELCLASS=ADOP) THEN STEPIT;                                            07668500 T  0084!002!1
          IF ELCLASS≠LITNO AND ELCLASS≠NONLITNO THEN                                     07669000 T  0084!002!1
            IF ELCLASS≠STRING AND(ELCLASS≠STRNGCON OR BOO) THEN                          07669500 T  0084!002!1
               BEGIN ERROR(302); GO GOOFUP END; % WHAT!SIT.                              07670000 T  0084!003!2
          IF BOO THEN C!=C&ELBAT[I=1][1:21:1];                                           07670500 T  0084!003!2
          IF ELCLASS=STRING THEN                                                         07671000 T  0084!003!0
               BEGIN                                                                     07671500 T  0084!003!2
               IF (T2!=T+(COUNT+7)DIV 8-1)>1022 THEN                                     07672000 T  0084!003!3
                    BEGIN ERROR(305); GO GOOFUP END; % > 1023,                           07672500 T  0084!004!1
```

```
                    T3:="       "; MOVE(1,T3,A[T2]);                                    07673000 T  0084:004211
                    MOVECHARACTERS(COUNT,ACCUM[1],3,A[T],0);                            07673500 T  0084:004512
                    T:=T2;                                                              07674000 T  0084:004712
                    END                                                                 07674500 T  0084:004810
                ELSE MOVE(1,C,A[T]);                                                    07675000 T  0084:004810
                END                                                                     07675500 T  0084:005113
          ELSE IF COUNT≤19 AND ACCUM[1],[18:18]="OCT" THEN                              07676000 T  0084:005113
                BEGIN % GET RID OF "OCT" FOR OCTIZE.                                    07676500 T  0084:005411
                MOVECHARACTERS(COUNT-3,ACCUM[1],6,ACCUM[1],3);                          07677000 T  0084:005512
                IF OCTIZE(ACCUM[1],A[T],19-COUNT,COUNT-3) THEN                          07677500 T  0084:005810
                  FLAG(303); % NON-OCTAL CHARACTER.                                     07678000 T  0084:006113
                END                                                                     07678500 T  0084:006211
              ELSE BEGIN ERROR(302); GO GOOFUP END; % WHATISIT.                         07679000 T  0084:006211
              T:=T+1;                                                                   07679500 T  0084:006610
CHECK:                                                                                  07680000 T  0084:006713
              IF STEPI≠COMMA THEN GO EXIT;                                              07680500 T  0084:006810
              END T LOOP;                                                               07681000 T  0084:006913
          ERROR(305); % > 1023 ITEMS IN LIST.                                           07681500 T  0084:007010
GOOFUP:                                                                                 07682000 T  0084:007011
          FILLIT:=TRUE;                                                                 07682500 T  0084:007112
EXIT:                                                                                   07683000 T  0084:007113
          PARENCOUNTER:=PARENCOUNTER-REAL(ELCLASS=RTPAREN);                             07683500 T  0084:007210
          END RECURSIVE FILLIT;                                                         07684000 T  0084:007313
                                                                                84 IS   78 LONG, NEXT SEG   83


    IF STEPI<BOOARRAYID OR ELCLASS>INTARRAYID THEN                                      07684500 T  0083:000313
        BEGIN                                                                           07685000 T  0083:000513
        IF ELCLASS=FILEID OR ELCLASS=SUPERFILEID THEN                                   07685500 T  0083:000610
          MAKEALABEL ELSE ERROR(300); % NO ARRAY ID.                                    07686000 T  0083:000810
        GO EXIT;                                                                        07686500 T  0083:001010
        END;                                                                            07687000 T  0083:001011
VARIABLE(FL); IF TABLE(I-2)≠FACTOP THEN FLAG(304); % NOT ARR. ROW.                      07687500 T  0083:001011
XMARK(ASSIGNREF); % FILL STATEMENT                                          %116-       07687600 C  0083:001411
IF ELCLASS≠WITHV THEN                                                                   07688000 T  0083:001811
        BEGIN ERROR(301); GO EXIT END; % MISSING "WITH".                                07688500 T  0083:001913
STREAMTOG:=TRUE;                                                                        07689000 T  0083:002112
IF TABLE(I+1)≤IDMAX THEN                                                                07689500 T  0083:002210
  IF Q="7INQUI" THEN                                                                    07690000 T  0083:002313
        BEGIN                                                                           07690500 T  0083:002512
        STREAMTOG:=FALSE; I:=I+1; STEPIT;                                               07691000 T  0083:002513
        EMITPAIR(9,COM); EMITO(DEL);                                                    07691500 T  0083:002810
        GO EXIT;                                                                        07692000 T  0083:002913
        END;                                                                            07692500 T  0083:003210
EMITNUM(SGAVL); EMITPAIR(7,COM); EMITO(DEL); EMITO(DEL);                                07693000 T  0083:003210
IF LISTER OR SEGSTOG THEN SEGMENTSTART;                                                 07693500 T  0083:003512
MOVFCODE(TEDOC,EDOC); PARENCOUNTER:=T:=0;                                               07694000 T  0083:003713
BUILDLINE:=BOOLEAN(2*REAL(BUILDLINE)) ;                                                 07694500 T  0083:004112
IF FILLIT(FILLTEMP) THEN % DO NOTHING.                                                  07695000 T  0083:004210
ELSE IF PARENCOUNTER≠1 THEN ERROR(306) % ODD # OF PARENS.                               07695500 T  0083:004313
ELSE BEGIN                                                                              07696000 T  0083:004611
        FOR J:=0 STEP 32 UNTIL T DO                                                     07696500 T  0083:004712
          MOVE(32,FILLTEMP[J],EDOC[J,[38:3],J,[41:7]]);                                 07697000 T  0083:004810
        SEGMENT(T,SGAVL,SGNO);                                                          07697500 T  0083:005410
        END;                                                                            07698000 T  0083:005512
MOVFCODE(TEDOC,EDOC); STREAMTOG:=FALSE;                                                 07698500 T  0083:005512
```

```
    BUILDLINE:=BUILDLINE.[46:1] ; SGAVL:=SGAVL+1;                                07699000 T  0083:0058:0
EXIT:                                                                            07699500 T  0083:0060:1
    END FILLSTMT;                                                                07700000 T  0083:0061:2
                                                                          83 IS   65 LONG, NEXT SEG    3



 COMMENT STMT DIRECTS TRAFFIC TO THE VARIOUS STATEMENT ROUTINES.  SOME        07710000 T  0003:0830:0
              CARE IS TAKEN TO PICK UP EXTRANEOUS DECLARATIONS.  THIS         07711000 T  0003:0830:0
              WILL SOMETIMES CAUSE ADDITIONAL ERROR MESSAGES.  THIS IS        07712000 T  0003:0830:0
              AN IMPERFECT ANALYSIS OF BEGIN-END PAIRS;                       07713000 T  0003:0830:0
 PROCEDURE STMT ;                                                            07714000 T  0003:0830:0
        BEGIN                                                                07715000 T  0003:0830:0
          LABEL AGAIN,LERR,LDEC,LPROC,LSPROC,LVAR,LAB,LREAD,LWRITE,          07716000 T  0003:0830:0
                                                                  START OF SEGMENT **********   85
             LSPACE,LCLOSE,LLOCK,LRWND,LDBL,LFOR,LWHILE,LDO,LFILL,LIF,       07717000 T  0085:0000:0
             LGO, LRELSE, LBEG, LBRK, EXIT;                                  07718000 T  0085:0000:0
          SWITCH S :=                                                        07719000 T  0085:0000:0
             LPROC, LERR,  LSPROC,LERR,  LERR,  LERR,  LERR,                 07720000 T  0085:0002:1
             LPROC, LPROC, LPROC, LPROC, LVAR,  LVAR,  LVAR,                 07721000 T  0085:0002:1
LVAR,                                                                        07722000 T  0085:0002:1
             LVAR,  LVAR,  LVAR,  LVAR,  LAB,   LERR,  LERR,                 07723000 T  0085:0002:1
             LERR,  LERR,  LERR,  LDEC,  LREAD, LWRITE,LSPACE,               07724000 T  0085:0002:1
             LCLOSE,LLOCK, LRWND, LDBL,  LFOR,  LWHILE,LDO,                  07725000 T  0085:0002:1
             EXIT,  EXIT,  EXIT,  LFILL, EXIT,  LIF,   LGO,                  07726000 T  0085:0002:1
             LRELSE,LBEG;                                                    07727000 T  0085:0002:1
        COMMENT THESE ADDITIONS ARE BEING MADE TO FORCE                      07727010 T  0085:0025:3
        CONSTANTCLEAN ACTION WHEN IT APPEARS THAT CONSTANTS WILL BE          07727020 T  0085:0025:3
        GENERATED IN THE STACK WHICH ARE TOO FAR AWAY AND CREL               07727030 T  0085:0025:3
        ADDRESSING IS NOT POSSIBLE;                                          07727040 T  0085:0025:3
        IF LASTENTRY ≠0 THEN                                                 07727050 T  0085:0025:3
             BEGIN GT2 := INFO [0:255];                                      07727055 T  0085:0026:1
                DO GT1 := GT2 UNTIL GT2:=GET(GT1) = 4095;                    07727060 T  0085:0029:2
                  IF L= GT1  > 400 THEN                                      07727065 T  0085:0032:0
                     BEGIN  GT1 := BUMPL;                                    07727070 T  0085:0033:2
                       CONSTANTCLEAN;                                        07727075 T  0085:0035:3
                       EMITB(BFW, GT1,L);                                    07727080 T  0085:0036:0
                     END;                                                    07727085 T  0085:0037:2
             END;                                                            07727090 T  0085:0037:2
             STACKCT := 0;                                          %A       07727100 T  0085:0037:2
AGAIN:       GO TO S[ELCLASS-SWITCHID];                                      07728000 T  0085:0038:0
             IF ELCLASS = 0 THEN                                             07728500 T  0085:0040:1
                BEGIN UNKNOWNSTMT; GO TO EXIT END;                           07729000 T  0085:0041:2
             IF ELCLASS=FAULTID THEN BEGIN FAULTSTMT; GO EXIT END;           07729100 T  0085:0044:0
             IF ELCLASS=FILEID OR ELCLASS=SUPERFILEID THEN                   07729190 T  0085:0046:0
                BEGIN GT1:=FILEATTRIBUTEHANDLER(FS); GO EXIT END ;           07729200 T  0085:0048:0
             FLAG(145);                                                      07729500 T  0085:0050:0
LERR:        ERR(144); GO TO EXIT;                                           07730000 T  0085:0051:2
LDEC:        FLAG(146);                                                      07731000 T  0085:0052:0
             IF TABLE(I-2) = ENDV AND MODE > 0                               07732000 T  0085:0053:3
               THEN BEGIN I := I-2; ELCLASS := ENDV; GO TO EXIT END;         07733000 T  0085:0055:3
             I := I-1; ERRORTOG := TRUE; BLOCK(FALSE);                       07734000 T  0085:0059:3
             ELCLASS := TABLE(I:=I-1); GO TO EXIT;                           07735000 T  0085:0062:0
                                                                             07735500 T  0085:0065:2
                                                                             07735510 T  0085:0065:2
                                                                             07735520 T  0085:0065:2
```

```
                LPROC:      PROCSTMT(TRUE); GO TO EXIT;                                  07736000 T  0085:0065:2
                LSPROC:     STRMPROCSTMT;  GO TO EXIT;                                   07737000 T  0085:0066:0
                LVAR:       VARIABLE(FS); GO TO EXIT;                                    07738000 T  0085:0068:0
                LAB:        LABELR; GO TO AGAIN;                                         07739000 T  0085:0069:2
                LREAD:      READSTMT; GO TO EXIT;                                        07740000 T  0085:0071:2
                LWRITE:     WRITESTMT; GO TO EXIT;                                       07741000 T  0085:0072:0
                LSPACE:     SPACESTMT; GO TO EXIT;                                       07742000 T  0085:0073:2
                LCLOSE:     CLOSESTMT; GO TO EXIT;                                       07743000 T  0085:0074:0
                LLOCK:      LOCKSTMT; GO TO EXIT;                                        07744000 T  0085:0075:2
                LRWND:      RWNDSTMT; GO TO EXIT;                                        07745000 T  0085:0076:0
                LDBL:       DBLSTMT; GO TO EXIT;                                         07746000 T  0085:0077:2
                LFOR:       FORSTMT; GO TO EXIT;                                         07747000 T  0085:0078:0
                LWHILE:     WHILESTMT; GO TO EXIT;                                       07748000 T  0085:0079:2
                LDO:        DOSTMT; GO TO EXIT;                                          07749000 T  0085:0080:0
                LFILL:      FILLSTMT; GO TO EXIT;                                        07750000 T  0085:0081:2
                LIF:        IFSTMT; GO TO EXIT;                                          07751000 T  0085:0082:0
                LGO:        GOSTMT; GO TO EXIT;                                          07752000 T  0085:0084:0
                LRELSE:     RELSESTMT; GO TO EXIT;                                       07753000 T  0085:0085:2
                LBEG:       IF STEPI = DECLARATORS                                       07754000 T  0085:0086:0
                              THEN BEGIN I := I-1; BLOCK(FALSE) END                      07755000 T  0085:0086:1
                              ELSE COMPOUNDTAIL;                                         07756000 T  0085:0089:3
                EXIT:   END STMT;                                                        07757000 T  0085:0090:1
                                                                                  85 IS   92 LONG, NEXT SEG    3


          PROCEDURE CMPLXSTMT; FORWARD ;                                                 07777777 T  0003:0830:0
PRT(1004) = CMPLXSTMT
                PROCEDURE UNKNOWNSTMT;                                                   07800000 T  0003:0830:0
                      BEGIN LABEL XXX,E;                                                 07801000 T  0003:0830:0
                                                                              START OF SEGMENT **********   86
                          REAL J,N,C;                                                    07802000 T  0086:0000:0
STACK(F+2) = J
STACK(F+3) = N
STACK(F+4) = C
                          IF Q = "5BREAK" THEN                                           07803000 T  0086:0000:0
                              BEGIN EMIT(0);                                             07804000 T  0086:0000:1
                                    EMIT(48);                                            07805000 T  0086:0002:0
                                    EMITO(COM);                                          07806000 T  0086:0002:1
                                    EMITO(DEL);                                          07807000 T  0086:0003:3
                                    STEPIT;                                              07808000 T  0086:0004:0
                                    GO TO XXX;                                           07809000 T  0086:0004:1
                              END;                                                       07810000 T  0086:0007:2
                          IF Q="7COMPL" THEN BEGIN CMPLXSTMT; GO XXX END ;               07810100 T  0086:0007:2
                          IF Q = "3ZIP00" THEN                                           07811000 T  0086:0011:2
                              BEGIN IF TABLE(I+1) = WITHV THEN                           07812000 T  0086:0011:3
                                        BEGIN STEPIT;                                    07813000 T  0086:0014:0
                                              IF STEPI < BOOARRAYID OR ELCLASS >         07814000 T  0086:0015:2
                                                 INTARRAYID THEN                         07814100 T  0086:0016:0
                                                    IF ELCLASS=FILEID OR                 07814200 T  0086:0017:2
                                                       ELCLASS=SUPERFILEID THEN          07814300 T  0086:0018:0
                                                         PASSFILE ELSE                   07814400 T  0086:0019:2
                                                         GO E ELSE                       07814500 T  0086:0020:0
                                              BEGIN                                      07814600 T  0086:0020:0
                                              VARIABLE(FL);                              07815000 T  0086:0022:0
                                              IF TABLE(I-2) ≠ FACTOP THEN GO TO E;       07816000 T  0086:0022:1
```

```
                    END;
                    EMIT(16); EMITO(COM); EMITO(DEL);
                    GO TO XXX;
               END;
          N ← 1; C ← 8
     END ELSE
IF Q = "5CHAIN" THEN
     BEGIN N ← 1; C ← 37 END ELSE
IF Q = "4WHENO" THEN
     BEGIN N ← 0; C ← 6 END ELSE
IF Q = "4WAITO" THEN
     BEGIN N ← 1; C ← 2 END ELSE
IF Q = "4CASEO" THEN BEGIN CASESTMT; GO TO XXX END ELSE
IF Q="4SORTO" THEN BEGIN SORTSTMT; GO XXX END ELSE
IF Q="5MERGE" THEN BEGIN MERGESTMT; GO XXX END ELSE
IF Q="6SEARC" THEN
BEGIN IF STEPI≠LEFTPAREN THEN
      BEGIN ERR(105);  GO TO XXX END;
      IF STEPI=FILEID OR ELCLASS=SUPERFILEID THEN
          PASSFILE ELSE GO TO E;
      IF ELCLASS≠COMMA THEN GO TO E;
      IF STEPI<BOOARRAYID OR ELCLASS>INTARRAYID THEN
        GO TO E;
      XMARK(ASSIGNREF); % SEARCH STATEMENT          %116-
      VARIABLE(FL);
      IF TABLE(I-2)≠FACTOP THEN GO TO E;
      IF ELCLASS≠RTPAREN THEN
      BEGIN ERR(104);  GO TO XXX END;
      EMITPAIR(30,COM);  EMITO(DEL);  EMITO(DEL);
       STEPIT;  GO TO XXX;
END ELSE
IF Q="4SEEKO" THEN
BEGIN IF STEPI≠LEFTPAREN THEN
     BEGIN ERR(105); GO TO XXX; END;
     IF STEPI≠FILEID AND  ELCLASS≠SUPERFILEID THEN
     GO TO E ELSE
     BEGIN EMITL(0); EMITL(0); PASSFILE;
          IF ELCLASS≠LEFTPAREN THEN
          BEGIN ERR(105); GO TO XXX; END;
          STEPIT; AEXP; EMITO(XCH);
          IF ELCLASS≠RTPAREN THEN
          BEGIN ERR(104); GO TO XXX; END;
          IF STEPI≠RTPAREN THEN
          BEGIN ERR(104); GO TO XXX; END;
          EMITPAIR(32,COM); EMITO(DEL); EMITO(DEL);
          EMITO(DEL); EMITO(DEL); STEPIT;
     END; GO TO XXX;
END  ELSE
IF Q="6UNLOC" THEN
BEGIN IF STEPI≠LEFTPAREN THEN
     BEGIN ERR(105);  GO TO XXX END;
     STEPIT;  VARIABLE(FL);  L ← L-1;
     IF TABLE(I-2)≠FACTOP THEN FLAG(208);
     EMITO(DUP);  EMITO(LOD);  EMITL(0);
     EMITD(43,3,5);  EMITO(XCH);  EMITO(STD);
     IF ELCLASS=RTPAREN THEN STEPIT ELSE ERR(104);
     GO TO XXX
```

```
07816100 T  0086:0025:2
07817000 T  0086:0025:2
07818000 T  0086:0027:2
07819000 T  0086:0027:3
07820000 T  0086:0027:3
07821000 T  0086:0028:1
07821100 T  0086:0029:2
07821200 T  0086:0030:1
07822000 T  0086:0032:1
07823000 T  0086:0034:1
07824000 T  0086:0036:1
07825000 T  0086:0039:3
07825500 T  0086:0041:3
07826000 T  0086:0048:0
07827000 T  0086:0052:0
07828000 T  0086:0056:0
07829000 T  0086:0057:2
07830000 T  0086:0058:1
07831000 T  0086:0062:0
07832000 T  0086:0064:0
07833000 T  0086:0065:2
07834000 T  0086:0066:0
07835000 T  0086:0068:0
07835500 C  0086:0069:2
07836000 T  0086:0073:2
07837000 T  0086:0074:0
07838000 T  0086:0076:0
07839000 T  0086:0077:2
07840000 T  0086:0078:1
07841000 T  0086:0081:2
07842000 T  0086:0082:0
07843000 T  0086:0082:0
07844000 T  0086:0083:3
07845000 T  0086:0085:2
07846000 T  0086:0088:0
07847000 T  0086:0090:0
07848000 T  0086:0090:0
07849000 T  0086:0092:1
07850000 T  0086:0093:3
07851000 T  0086:0095:2
07852000 T  0086:0097:2
07853000 T  0086:0097:3
07854000 T  0086:0099:3
07855000 T  0086:0100:1
07856000 T  0086:0102:0
07857000 T  0086:0104:1
07858000 T  0086:0106:1
07859000 T  0086:0107:2
07859010 T  0086:0107:2
07859020 T  0086:0108:1
07859030 T  0086:0110:0
07859040 T  0086:0113:2
07859050 T  0086:0115:3
07859060 T  0086:0118:1
07859070 T  0086:0120:1
07859080 T  0086:0123:3
07859090 T  0086:0126:1
```

```
                    END ELSE                                                                07859100 T  0086:0127:2
               $ SET OMIT = NOT TSPOL                                                       07859900 T  0086:0127:2
                         BEGIN ERROR(100); GO TO XXX END;                                   07900000 T  0086:0127:2
                    IF STEPI ≠ LEFTPAREN THEN                                               07901000 T  0086:0128:1
                         BEGIN ERR(105); GO TO XXX END;                                     07902000 T  0086:0129:3
                    STEPIT; AEXP;                                                           07903000 T  0086:0131:3
                    FOR J ← 1 STEP 1 UNTIL N DO                                             07904000 T  0086:0132:1
                         BEGIN IF ELCLASS ≠ COMMA THEN                                      07905000 T  0086:0134:0
               E:                  BEGIN ERR(164); GO TO XXX END;                           07906000 T  0086:0134:1
                              STEPIT; AEXP;                                                 07907000 T  0086:0137:2
                         END;                                                               07908000 T  0086:0138:0
                    IF ELCLASS ≠ RTPAREN THEN                                               07909000 T  0086:0140:1
                         BEGIN ERR(104);GO TO XXX END;                                      07910000 T  0086:0141:2
                    EMITL(C); EMITO(COM);                                                   07911000 T  0086:0143:2
                    FOR J ← 0 STEP 1 UNTIL N DO EMITO(DEL);                                 07912000 T  0086:0144:1
                    STEPIT;                                                                 07913000 T  0086:0149:2
               XXX: END;                                                                    07914000 T  0086:0149:3
                                                                                      86 IS  153 LONG, NEXT SEG    3


          PROCEDURE FAULTSTMT; COMMENT THIS IS WHAT HAPPENS FOR THE"<FAULTYPE>←"            07920000 T  0003:0830:0
                              KIND OF STATEMENT, FOR THE RUN-TIME ERROR MESS;               07921000 T  0003:0830:0
            BEGIN REAL ELBW,STR; DEFINE ADRES=ELBW.ADDRESS#;                                07922000 T  0003:0830:0
                                                                                  START OF SEGMENT **********   87
STACK(F+2) = ELBW
STACK(F+3) = STR
                 CHECKER(ELBW←ELBAT[I]); STR←IF FAULTOG THEN SND ELSE STD;                  07923000 T  0087:0000:0
                 FAULTOG ← BOOLEAN(1) OR FAULTOG; COMMENT TELLS DEXP TO MESS                07923100 T  0087:0003:3
                                                  WITH FAULTLEVEL;                          07923150 T  0087:0005:2
                 IF STEPI≠ASSIGNOP THEN ERR (60) ELSE                                       07924000 T  0087:0005:2
                 IF STEPI=LITNO THEN BEGIN EMIT(0); STEPIT END ELSE                         07925000 T  0087:0007:2
                 IF ELCLASS=FAULTID THEN FAULTSTMT ELSE DEXP;                               07925100 T  0087:0010:1
                 EMITPAIR(ADRES,STR);                                                       07926000 T  0087:0013:3
                 FAULTOG←FALSE&(ELBW.LVL<LEVEL OR FAULTOG.[46:1])[46:47:1];                 07926100 T  0087:0015:2
            END FAULTSTMT   NOW WASNT THAT SIMPLE;                                          07927000 T  0087:0019:2
                                                                                      87 IS   22 LONG, NEXT SEG    3


          PROCEDURE KLUDGE(T); VALUE T; INTEGER T;                                          07930000 T  0003:0830:0
PRT(1005) = KLUDGE
            BEGIN COMMENT KLUDGE HANDLES ARRAY-ROW READS AND WRITES FOR                     07931000 T  0003:0830:0
                      THOSE CASES WHICH DO NOT NEED TO GO THROUGH THE                       07932000 T  0003:0830:0
                      FORMATTING INTRINSICS,  A NEW MCP INTRINSIC IS                        07933000 T  0003:0830:0
                      USED, TO FURTHER IMPROVE SPEED/DECREASE CORE USE;                     07934000 T  0003:0830:0
                 LABEL EXIT;                                                                07935000 T  0003:0830:0
                                                                                  START OF SEGMENT **********   88
                 L ← ABS(T);                                                                07936000 T  0088:0000:0
                 AEXP;                                                                      07937000 T  0088:0001:2
                 IF ELCLASS≠COMMA THEN BEGIN ERR(426); GO TO EXIT; END;                     07938000 T  0088:0001:3
                 IF STEPI<BOOARRAYID OR ELCLASS>INTARRAYID THEN                             07939000 T  0088:0004:0
                 BEGIN ERR(429); GO TO EXIT; END;                                           07940000 T  0088:0006:0
                 VARIABLE(FL); IF TABLE(I-2)≠FACTOP THEN                                    07941000 T  0088:0007:3
                 BEGIN ERR(427); GO TO EXIT; END;                                           07942000 T  0088:0010:0
                 IF ELCLASS≠RTPAREN THEN BEGIN ERR(428); GO TO EXIT; END;                   07943000 T  0088:0012:0
```

```
            EMITO(XCH);                                                                  07944000 T  0088:0014:1
            IF T<0 THEN    COMMENT FROM WRITE...(<0 IS FROM READ);                       07945000 T  0088:0015:2
            BEGIN EMITPAIR(JUNK,STD); EMITO(XCH); EMITV(JUNK); END;                      07946000 T  0088:0016:0
            IF T>0 THEN IF TABLE(I+1)=LFTBRKET THEN                                      07947000 T  0088:0019:2
            BEGIN GOGOGO ← FALSE;% JUST TO MAKE SURE...                                  07948000 T  0088:0022:0
                 HANDLETHETAILENDOFAREADORSPACESTATEMENT;%                               07949000 T  0088:0023:2
                 L ← L-1;% REMOVE THE OPDC ON INPUTINT...                                07950000 T  0088:0023:3
                 EMITO(DEL); EMITO(DEL);% REMOVE LABEL WORDS...                          07951000 T  0088:0025:2
            END ELSE STEPIT ELSE STEPIT;% WALTZ ON BY...                                 07952000 T  0088:0026:1
            EMITV(GNAT(SUPERMOVER));% BET YOU THOUGHT I'D NEVER DO IT                    07953000 T  0088:0028:1
       EXIT; END THIS HAIRY KLUDGE;%                                                     07954000 T  0088:0029:3
                                                                                  88 IS   31 LONG, NEXT SEG    3



COMMENT FORSTMT IS REPONSIBLE FOR THE COMPILATION OF FOR STATEMENTS,                     08000000 T  0003:0830:0
        IF THE FOR STATEMENT HAS A SINGLE STEP-UNTIL ELEMENT SUCH                        08001000 T  0003:0830:0
        THAT THE INITIAL VALUE, THE STEP AND THE FINAL VALUE ARE                         08002000 T  0003:0830:0
        ALL OF THE FORM V,+V, OR -V WHERE V IS A VARIABLE OR A                           08003000 T  0003:0830:0
        CONSTANT, THEN THE CODE TAKES ON A MORE EFFICIENT FORM,                          08004000 T  0003:0830:0
        IN OTHER CASES THE CODE IS SOMEWHAT LESS EFFICIENT, SINCE                        08005000 T  0003:0830:0
        THE BODY OF THE FOR STATEMENT BECOMES A SUBROUTINE.  THE                         08006000 T  0003:0830:0
        STEP ALSO BECOMES A SUBROUTINE IF IT IS NOT SIMPLE;                              08007000 T  0003:0830:0
PROCEDURE FORSTMT;                                                                       08008000 T  0003:0830:0
      BEGIN                                                                              08009000 T  0003:0830:0
       OWN REAL B,STMTSTART,REGO,RETURNSTORE,ADDRES,V,VRET,                              08010000 T  0003:0830:0
                                                                                START OF SEGMENT **********   89

PRT(1006) = B
PRT(1007) = STMTSTART
PRT(1010) = REGO
PRT(1011) = RETURNSTORE
PRT(1012) = ADDRES
PRT(1013) = V
PRT(1014) = VRET
                                BRET;                                                    08011000 T  0089:0000:0
PRT(1015) = BRET
                    OWN BOOLEAN SIGNA,SIGNB,SIGNC, INT,                                  08012000 T  0089:0000:0
PRT(1016) = SIGNA
PRT(1017) = SIGNB
PRT(1020) = SIGNC
PRT(1021) = INT
                                CONSTANA,CONSTANB,CONSTANC;                              08013000 T  0089:0000:0
PRT(1022) = CONSTANA
PRT(1023) = CONSTANB
PRT(1024) = CONSTANC
                        DEFINE SIMPLEB = SIGNC#,          FORMALV = SIGNA#,              08014000 T  0089:0000:0
                               SIMPLEV = CONSTANA#, A = V#, Q = REGO#,                   08015000 T  0089:0000:0
                               OPDC = TRUE#, DESC = FALSE#, K = BRET#;                   08016000 T  0089:0000:0
                        LABEL EXIT;                                                      08017000 T  0089:0000:0
                COMMENT FORCLASS CHECKS FOR THE APPROPRIATE WORD STEP, UNTIL, OR DO--     08017100 T  0089:0000:0
                        IF A CONSTANT IS FOUND, IT STORES OFF THE VALUE (FROM C) AT      08017200 T  0089:0000:0
                        INFO[0,K] AND STUFFS K INTO THE ELBAT WORD, SO THAT TABLE CAN    08017300 T  0089:0000:0
                        RECONSTRUCT THE CONSTANT EHEN WE SCAN ELBAT AGAIN;               08017400 T  0089:0000:0
                    BOOLEAN PROCEDURE FORCLASS(CLSS); VALUE CLSS; INTEGER CLSS;          08017500 T  0089:0000:0
PRT(1025) = FORCLASS
                        IF STEP1   = CLSS THEN FORCLASS ← TRUE ELSE                      08017600 T  0089:0000:0
```

```
                    IF ELCLASS ≥ NONLITNO AND ELCLASS ≤ STRNGCON THEN                    08017700 T  0089:0002:0
                    BEGIN INFO[0,K←K+1] ← C;                                             08017800 T  0089:0004:1
                          ELBAT[I] ← 0&COMMENTV[2:41:7]&K[16:37:11]                      08017900 T  0089:0008:1
                    END FORCLASS;                                                        08017950 T  0089:0011:3


               COMMENT PLUG EMITS EITHER AN OPERAND CALL ON A VARIABLE OR A CALL ON A    08018000 T  0089:0014:0
                       CONSTANT DEPENDING ON THE REQUIREMENTS;                           08019000 T  0089:0014:0
               PROCEDURE PLUG(C,A); VALUE C,A; REAL A; BOOLEAN C;                        08020000 T  0089:0014:0
PRT(1026) = PLUG
                    IF C THEN EMITNUM (A) ELSE BEGIN                                     08021000 T  0089:0014:0
                              CHECKER (A);                                               08021100 T  0089:0017:2
                              EMITV(A.ADDRESS) END;                                      08021200 T  0089:0017:3


               COMMENT SIMPLE DETERMINES IF AN ARITHMETIC EXPRESSION IS + OR - A         08022000 T  0089:0019:2
                       CONSTANT OR A SIMPLE VARIABLE.  IT MAKES A THROUGH REPORT         08023000 T  0089:0019:2
                       ON ITS ACTIVITY.  IT ALSO MAKES PROVISION FOR THE RESCAN          08024000 T  0089:0019:2
                       OF ELBAT (THIS IS THE ACTION WITH K = SEE CODE IN THE             08025000 T  0089:0019:2
                       TABLE ROUTINE FOR FURTHER DETAILS);                               08026000 T  0089:0019:2
               BOOLEAN PROCEDURE SIMPLE(B,A,S); BOOLEAN B,S; REAL A;                     08027000 T  0089:0019:2
PRT(1027) = SIMPLE
                    BEGIN                                                                08028000 T  0089:0019:2
                         S ← IF STEPI ≠ ADOP THEN FALSE ELSE ELBAT[I].ADDRESS            08029000 T  0089:0019:2
                                   = SUB;                                                08030000 T  0089:0022:1
                         IF ELCLASS = ADOP THEN STEPIT;                                  08031000 T  0089:0024:1
                         IF ELCLASS ≥ NONLITNO AND ELCLASS ≤ STRNGCON                    08032000 T  0089:0026:0
                           THEN BEGIN K ← K+1; SIMPLE ← TRUE;                            08033000 T  0089:0027:2
                              ELBAT[I] ← 0&COMMENTV[2:41:7]&K[16:37:11];                 08034000 T  0089:0031:2
                              INFO[0,K] ← A ← C; B ← TRUE END                            08035000 T  0089:0034:1
                         ELSE BEGIN                                                      08036000 T  0089:0038:1
                          B ← FALSE; A ← ELBAT[I];                                       08037000 T  0089:0039:2
                          SIMPLE ← REALID ≤ ELCLASS AND ELCLASS ≤ INTID END;             08038000 T  0089:0041:2
                    END SIMPLE;                                                          08039000 T  0089:0043:3


               COMMENT TEST EMITS THE STEP-UNTIL ELEMENT TEST;                           08040000 T  0089:0045:2
               PROCEDURE TEST;                                                           08041000 T  0089:0045:2
PRT(1030) = TEST
                    BEGIN                                                                08042000 T  0089:0045:2
                         IF NOT CONSTANB THEN                                            08043000 T  0089:0045:2
                           BEGIN EMITO(SUB); IF SIMPLEB THEN EMITV(B.ADDRESS)            08044000 T  0089:0046:1
                              ELSE BEGIN                                                 08045000 T  0089:0049:2
                                EMITL(2+L-BRET);                                         08046000 T  0089:0051:2
                                EMITB(BBW,BUMPL,B);                                      08047000 T  0089:0053:2
                              END;                                                       08048000 T  0089:0055:3
                              EMITO(MUL); EMIT(0) END;                                   08049000 T  0089:0055:3
                              EMITO(IF SIGNB THEN GEQ ELSE LEQ); EMIT (0); L←L+1         08050000 T  0089:0057:2
                    END TEST;                                                            08051000 T  0089:0060:1
```

```
                BOOLEAN PROCEDURE SIMPI(ALL); VALUE ALL; REAL ALL;                    08052000 T  00891006113
PRT(1031) = SIMPI
                        BEGIN                                                         08053000 T  00891006113
                             CHECKER(VRET+ALL);                                       08054000 T  00891006113
                             ADDRES + ALL.ADDRESS;                                    08055000 T  00891006313
                             FORMALV + ALL.[9:2] = 2;                                 08056000 T  00891006512
                             IF T + ALL.CLASS > INTARRAYID OR T < BOOID OR            08057000 T  00891006712
                                GT1 + (T-BOOID) MOD 4 < 1 THEN                        08058000 T  00891006913
                                  ERR(REAL(T ≠ 0) × 51 + 100);                        08059000 T  00891007210
                             INT + GT1 = 3;                                           08060000 T  00891007512
                             SIMPI + T ≤ INTID END SIMPI;                             08061000 T  00891007611


                COMMENT  STORE EMITS THE CODE FOR THE STORE INTO THE FOR INDEX;       08062000 T  00891008010
                PROCEDURE STORE(S); VALUE S; BOOLEAN S;                               08063000 T  00891008010
PRT(1032) = STORE
                        BEGIN                                                         08064000 T  00891008010
                          IF FORMALV THEN BEGIN EMITO(XCH); S + FALSE END             08065000 T  00891008010
                           ELSE BEGIN                                                 08066000 T  00891008313
                             EMITL(ADDRES);                                           08067000 T  00891008410
                             IF ADDRES  > 1023 THEN EMITO(PRTE) END;                  08068000 T  00891008512
                              T  + (REAL(S)+1)×16;                                    08069000 T  00891008712
                             EMITO((IF INT THEN T+512 ELSE 4×T)+4) END STORE;         08070000 T  00891008912


                COMMENT CALL EFFECTS A CALL ON THE INDEX;                             08071000 T  00891009312
                PROCEDURE CALL(S); VALUE S; BOOLEAN S;                                08072000 T  00891009312
PRT(1033) = CALL
                             BEGIN                                                    08073000 T  00891009312
                              IF SIMPLEV                                              08074000 T  00891009312
                                THEN IF S THEN EMITV(ADDRES) ELSE EMITN(ADDRES)       08075000 T  00891009410
                                ELSE BEGIN                                            08076000 T  00891009713
                                  EMITL(2+L-VRET);                                    08077000 T  00891009811
                                  EMITB(BBW,BUMPL,V);                                 08078000 T  00891010011
                                  IF S THEN EMITO(LOD) END END CALL;                  08079000 T  00891010312


                PROCEDURE FORLIST(NUMLE); VALUE NUMLE; BOOLEAN NUMLE;                 08080000 T  00891010512
PRT(1034) = FORLIST
                        BEGIN                                                         08081000 T  00891010512
                PROCEDURE FIX(STORE,BACK,FORWART,START);                              08082000 T  00891010512
                                                                                  START OF SEGMENT **********   90
PRT(1035) = FIX
                        VALUE STORE,BACK,FORWART,START;                               08083000 T  00901000010
                        REAL  STORE,BACK,FORWART,START;                               08084000 T  00901000010
                        BEGIN                                                         08085000 T  00901000010
                             EMITB(GET(FORWART-1),FORWART,START);                     08086000 T  00901000010
                             IF RETURNSTORE ≠ 0                                       08087000 T  00901000210
                               THEN BEGIN                                             08088000 T  00901000211
                                  L + STORE; EMITNUM(B-BACK);                         08089000 T  00901000313
                                  EMITPAIR(RETURNSTORE,STD) END END FIX;              08090000 T  00901000610
```

```
                        INTEGER BACKFIX, FORWARDBRANCH, FOOT, STOREFIX;                  08091000 T  00901000713
STACK(F+2) = BACKFIX
STACK(F+3) = FORWARDBRANCH
STACK(F+4) = FOOT
STACK(F+5) = STOREFIX
                        LABEL BRNCH,EXIT;                                                08092000 T  00901000713
                        STOREFIX ← L; Q ← REAL(MODE=0)+3;                                08093000 T  00901000713
                        FOR K ← 1 STEP 1 UNTIL Q DO EMITO(NOP);                          08094000 T  00901001011
                        IF NUMLE                                                         08095000 T  00901001513
                         THEN BEGIN                                                      08096000 T  00901001513
                             BACKFIX ← L;                                                08097000 T  00901001611
                             IF FORMALV THEN CALL(DESC) END                              08098000 T  00901001712
                         ELSE BACKFIX ← V ← REAL(SIMPLEV)-1;                             08099000 T  00901001912
                        DIALA ← DIALB ← 0;                                               08100000 T  00901002210
                        AEXP; DIALA ← DIALB ← 0;                                         08101000 T  00901002312
                COMMENT PICK UP FIRST ARITHMETIC EXPRESSION;                             08102000 T  00901002512
                        IF  ELCLASS = STEPV                                              08103000 T  00901002512
                           THEN BEGIN                                                    08104000 T  00901002512
                COMMENT HERE WE HAVE A STEP ELEMENT;                                     08105000 T  00901002610
                             BACKFIX ← BUMPL;                                            08106000 T  00901002610
                COMMENT LEAVE ROOM FOR FORWARD JUMP;                                     08107000 T  00901002810
                             IF FORMALV THEN CALL(DESC); CALL(OPDC);                     08108000 T  00901002810
                COMMENT FETCH INDEX;                                                     08109000 T  00901003112
                             IF I > 70 THEN BEGIN NXTELBT ← 1; I ← 0 END                 08110000 T  00901003112
                                 ELSE REGO ← I;                                          08111000 T  00901003313
                             IF SIMPLEB ← SIMPLE(CONSTANB,B,SIGNB) AND                   08112000 T  00901003512
                               (STEPI = UNTILV OR ELCLASS = WHILEV)                      08113000 T  00901003713
                                 THEN BEGIN                                              08114000 T  00901003811
                COMMENT WE HAVE A SIMPLE STEP FUNCTION;                                  08115000 T  00901004112
                                  PLUG(CONSTANB ,B);                                     08116000 T  00901004112
                                END ELSE BEGIN                                           08117000 T  00901004211
                COMMENT THE STEP FUNCTION IS NOT SIMPLE;  WE CONSTRUCT A                 08118000 T  00901004312
                             SUBROUTINE;                                                 08119000 T  00901004312
                                 I ← IF I < 4 THEN 0 ELSE REGO; STEPIT;                  08120000 T  00901004312
                                 SIGNB ← CONSTANB ← FALSE;                               08121000 T  00901004611
                                 EMIT(0); B ← L;                                         08122000 T  00901004811
                                 AEXP; EMITO(XCH);                                       08123000 T  00901005010
                                 BRET ← L;                                               08124000 T  00901005113
                                 EMITO(BFW) END;                                         08125000 T  00901005211
                             EMITO(REAL(SIGNB)×32+ADD);                                  08126000 T  00901005312
                             EMITB(BFW,BACKFIX,L);                                       08127000 T  00901005512
                             IF ELCLASS = UNTILV                                         08128000 T  00901005611
                               THEN BEGIN  COMMENT  STEP-UNTIL ELEMENT;                  08129000 T  00901005611
                                  STORE(TRUE); IF FORMALV THEN CALL(OPDC);               08130000 T  00901005713
                                  STEPIT; AEXP; TEST END                                 08131000 T  00901006011
                               ELSE BEGIN  COMMENT  STEP-WHILE ELEMENT;                  08132000 T  00901006211
                                  IF ELCLASS ≠ WHILEV THEN                               08133000 T  00901006312
                                    BEGIN ERR(153); GO TO EXIT END;                      08134000 T  00901006313
                                  STEPIT; STORE(FALSE); BEXP END END                     08135000 T  00901006513
                          ELSE BEGIN                                                     08136000 T  00901006713
                COMMENT WE DO NOT HAVE A STEP ELEMENT;                                   08137000 T  00901006810
                             STORE(FALSE);                                               08138000 T  00901006810
                             IF ELCLASS = WHILEV                                         08139000 T  00901006912
                               THEN BEGIN                                                08140000 T  00901006912
```

```
                COMMENT   WE HAVE A WHILE ELEMENT;                                                    08141000 T  0090:007:0
                          STEPIT; BEXP END                                                            08142000 T  0090:007:0
                        ELSE BEGIN                                                                    08143000 T  0090:007:2
                COMMENT   ONE EXPRESSION ELEMENT;                                                     08144000 T  0090:007:3
                          IF ELCLASS ≠ COMMA THEN BEGIN                                               08145000 T  0090:007:3
                              EMITB(BFW,BUMPL,L+2); BACKFIX ← L END                                   08146000 T  0090:007:2
                           ELSE BACKFIX ← L + 2;                                                      08147000 T  0090:007:1
                          L ← L+1; EMIT(BFW); GO TO BRNCH END END;                                    08148000 T  0090:007:0
                COMMENT   THIS IS THE COMMON POINT;                                                   08149000 T  0090:008:1
                          IF ELCLASS = COMMA THEN EMITLNG; L ← L+1;                                   08150000 T  0090:008:1
                          EMIT(BFC);                                                                  08151000 T  0090:008:3
BRNCH:                    FORWARDBRANCH ← L; DIALA ← DIALB ← 0;                                       08152000 T  0090:008:1
                          IF ELCLASS = COMMA                                                          08153000 T  0090:008:2
                            THEN BEGIN                                                                08154000 T  0090:008:2
                               STEPIT;                                                                08155000 T  0090:008:0
                               FORLIST(TRUE);                                                         08156000 T  0090:008:1
                               FIX(STOREFIX,BACKFIX,FORWARDBRANCH,STMTSTART) END                      08157000 T  0090:008:3
                            ELSE BEGIN                                                                08158000 T  0090:009:3
                               IF ELCLASS ≠ DOV                                                       08159000 T  0090:009:0
                                  THEN BEGIN ERR(154); REGO←L; GO EXIT END;                           08160000 T  0090:009:1
                               STEPIT;                                                                08161000 T  0090:009:3
                               IF NUMLE THEN FOOT := GETSPACE(FALSE,-1); % TEMP.                      08162000 T  0090:009:0
                               IF LISTMODE THEN LISTELEMENT ELSE STMT;                                08163000 T  0090:009:1
                                                                                                      08164000 T  0090:010:2
                               IF NUMLE THEN BEGIN                                                    08165000 T  0090:010:2
                                 EMITV(RETURNSTORE ← FOOT); EMITO(BBW) END                            08166000 T  0090:010:3
                               ELSE BEGIN                                                             08167000 T  0090:010:0
                                 EMITB(BBW,BUMPL,BACKFIX); RETURNSTORE ← 0 END;                       08168000 T  0090:010:1
                               STMTSTART ← FORWARDBRANCH; B ← L;                                      08169000 T  0090:010:3
                               CONSTANTCLEAN; REGO ← L;                                               08170000 T  0090:010:3
                               FIX(STOREFIX,BACKFIX,FORWARDBRANCH,L) END;                             08171000 T  0090:011:2
                EXIT:       END FORLIST;                                                              08172000 T  0090:011:2
                                                                                          90 IS  116 LONG, NEXT SEG   89


                          REAL T1,T2,T3,T4;                                                           08173000 T  0089:010:2
STACK(F+2) = T1
STACK(F+3) = T2
STACK(F+4) = T3
STACK(F+5) = T4
                          NXTELBT ← 1; I ← 0;                                                         08174000 T  0089:010:2
                          STEPIT;                                                                     08175000 T  0089:010:1
                          IF SIMPI(VRET+ELBAT[I])                                                     08176000 T  0089:010:2
                            THEN BEGIN                                                                08177000 T  0089:010:3
                               IF STEP≠ ≠ ASSIGNOP THEN BEGIN ERR(152); GO EXIT END;                  08178000 T  0089:010:3
                               XMARK(ASSIGNREF); % FOR STATEMENT                        %116-        08178100 C  0089:011:0
                               T1 ← L; IF FORMALV THEN EMITN(ADDRES);                                 08179000 T  0089:011:1
                               K ← 0;                                                                 08180000 T  0089:011:2
                               IF SIMPLE(CONSTANA,A,SIGNA) THEN                                       08181000 T  0089:012:0
                          IF FORCLASS(STEPV) THEN                                                     08182000 T  0089:012:1
                               IF SIMPLE(CONSTANB,B,SIGNB) THEN                                       08183000 T  0089:012:0
                          IF FORCLASS(UNTILV) THEN                                                    08184000 T  0089:012:1
                               IF SIMPLE(CONSTANC,Q,SIGNC) THEN                                       08185000 T  0089:012:0
                          IF FORCLASS(DOV) THEN                                                       08186000 T  0089:013:2
                                 BEGIN                                                                08187000 T  0089:013:1
```

```
            PLUG(CONSTANA,A);                                                    08188000 T  0089:0133:2
            IF SIGNA THEN EMITO(CHS);                                            08189000 T  0089:0134:1
            RETURNSTORE ← BUMPL; ADJUST; CONSTANTCLEAN;                          08190000 T  0089:0136:1
            STMTSTART ← L;                                                       08191000 T  0089:0139:3
STEPIT;                                                                          08192000 T  0089:0140:1
T1 ← ((((4096 × RETURNSTORE+STMTSTART)×2+                                        08193000 T  0089:0141:2
     REAL(CONSTANB))×2+                                                          08194000 T  0089:0143:2
     REAL(CONSTANC))×2+                                                          08195000 T  0089:0144:1
     REAL(SIGNB))×2+                                                             08196000 T  0089:0145:3
     REAL(SIGNC);                                                                08197000 T  0089:0147:2
T2 ← VRET;                                                                       08198000 T  0089:0148:0
T3 ← B;                                                                          08199000 T  0089:0149:2
T4 ← Q;                                                                          08200000 T  0089:0150:0
IF LISTMODE THEN LISTELEMENT ELSE STMT;                                          08201000 T  0089:0151:2
SIGNC ← BOOLEAN(T1,[47:1]);                                                      08202000 T  0089:0154:1
SIGNB ← BOOLEAN(T1,[46:1]);                                                      08203000 T  0089:0156:0
CONSTANC ← BOOLEAN(T1,[45:1]);                                                   08204000 T  0089:0157:3
CONSTANB ← BOOLEAN(T1,[44:1]);                                                   08205000 T  0089:0159:2
STMTSTART ← T1,[32:12];                                                          08206000 T  0089:0160:1
RETURNSTORE ← T1,[20:12];                                                        08207000 T  0089:0162:0
VRET ← T2;                                                                       08208000 T  0089:0163:3
B ← T3;                                                                          08209000 T  0089:0164:1
Q ← T4;                                                                          08210000 T  0089:0165:3
SIMPLEV← SIMPI(VRET);                                                            08211000 T  0089:0166:1
            IF FORMALV THEN EMITN(ADDRES); EMITV(ADDRES);                        08212000 T  0089:0168:1
            PLUG(CONSTANB,B);                                                    08213000 T  0089:0171:3
            EMITO(IF SIGNB THEN SUB ELSE ADD);                                   08214000 T  0089:0173:2
            EMITB(BFW,RETURNSTORE,L);                                            08215000 T  0089:0175:3
            STORE(TRUE);                                                         08216000 T  0089:0177:2
               IF FORMALV THEN CALL(OPDC);                                       08217000 T  0089:0178:0
               PLUG(CONSTANC,Q);                                                 08218000 T  0089:0180:0
               IF SIGNC THEN EMITO(CHS);                                         08219000 T  0089:0182:0
              SIMPLEB ← TRUE; TEST; EMITLNG;                                     08220000 T  0089:0183:3
              EMITB(BBC,BUMPL,STMTSTART);                                        08221000 T  0089:0186:0
              GO TO EXIT END;                                                    08222000 T  0089:0188:1
         I ← 2; K ← 0;                                                           08223000 T  0089:0189:2
         SIMPLEV ← SIMPI(VRET);                                                  08224000 T  0089:0190:1
         V ← T1 END                                                              08225000 T  0089:0192:1
      ELSE BEGIN                                                                 08226000 T  0089:0193:3
         EMIT(0); V ← L; SIMPLEV ← FALSE; FORMALV ← TRUE;                        08227000 T  0089:0194:0
         VARIABLE(FR); EMITO(XCH); VRET ← L; EMITO(BFW);                         08228000 T  0089:0198:0
         IF ELCLASS≠ASSIGNOP THEN BEGIN ERR(152); GO EXIT END;                   08229000 T  0089:0201:2
         END;                                                                    08230000 T  0089:0203:3
      STEPIT; FORLIST(FALSE); L ← REGO;                                          08231000 T  0089:0203:3
EXIT; K ← 0 END FORSTMT;                                                         08232000 T  0089:0206:0
                                                                    89 IS  211 LONG, NEXT SEG    3


PROCEDURE HANDLETHETAILENDOFAREADORSPACESTATEMENT;                               08233000 T  0003:0830:0
      BEGIN COMMENT THIS ROUTINE CHECK FOR ACTION LABELS IN READ AND             08234000 T  0003:0830:0
           SPACE STATEMENTS AND GENERATES THE APPROPRIATE CODE;                  08235000 T  0003:0830:0
           LABEL PASSPARLABL;  COMMENT WHEN I REACH THIS LABEL A                 08236000 T  0003:0830:0
                                                          START OF SEGMENT **********   91
                              COLON HAS JUST BEEN DETECTED;                      08237000 T  0091:0000:0
           LABEL EXIT;  COMMENT THE LABEL EXIT APPEARS AFTER THE LAST            08238000 T  0091:0000:0
```

```
                    EXECUTABLE STATEMENT IN THIS ROUTINE;
IF STEPI = LFTBRKET
THEN BEGIN COMMENT THIS CODE HANDLES PARITY AND END OF
          FILE LABELS;
     IF STEPI # COLON THEN DEXP ELSE EMIT(0);
     IF ELCLASS # COLON THEN EMIT(0) ELSE
               BEGIN STEPIT; DEXP END;
                    IF CHECK(RTBRKET,433)
                    THEN GO TO EXIT;
                     COMMENT ERROR 433 MEANS MISSING RIGHT BRACKET
                    IN READ OR SPACE STATEMENT;
                    STEPIT;
               END
          ELSE BEGIN COMMENT THERE ARE NOT ANY ACTION LABELS IN THIS
                    CASE;
                    EMITL(0);       EMITL(0);
               END;
IF GOGOGO THEN BEGIN EMIT(0);  EMIT(0);  EMIT(0);
                    EMITV(13)
               END ELSE EMITV(GNAT(INTERPTI));
               GOGOGO ← FALSE;%
EXIT:;
END HANDLETHETAILENDORAREADORSPACESTATEMENT;
```

```
08239000 T  0091:000010
08240000 T  0091:000010
08241000 T  0091:000011
08242000 T  0091:000113
08243000 T  0091:000113
08244000 T  0091:000411
08245000 T  0091:000611
08246000 T  0091:000810
08247000 T  0091:000810
08248000 T  0091:000810
08249000 T  0091:000810
08250000 T  0091:000810
08251000 T  0091:000810
08252000 T  0091:000810
08253000 T  0091:000810
08254000 T  0091:000810
08255000 T  0091:000810
08256000 T  0091:000810
08257000 T  0091:000810
08258000 T  0091:000810
08259000 T  0091:000810
08260000 T  0091:000810
08261000 T  0091:000810
08262000 T  0091:000810
08263000 T  0091:000810
08264000 T  0091:000810
08265000 T  0091:000810
08266000 T  0091:000810
08267000 T  0091:000810
08268000 T  0091:000810
08269000 T  0091:000810
08270000 T  0091:000810
08271000 T  0091:000810
08272000 T  0091:000810
08273000 T  0091:000810
08274000 T  0091:000810
08275000 T  0091:000810
08276000 T  0091:000810
08277000 T  0091:000810
08278000 T  0091:000811
08279000 T  0091:001010
08280000 T  0091:001010
08281000 T  0091:001010
08282000 T  0091:001011
08283000 T  0091:001011
08284000 T  0091:001112
08285000 T  0091:001112
08286000 T  0091:001211
08287000 T  0091:001211
08287100 T  0091:001513
08287200 T  0091:001513
08287300 T  0091:001810
08288000 T  0091:001811
08289000 T  0091:001912
91 IS   20 LONG, NEXT SEG    3
```

```
                DEFINE EMITNO(EMITNO1)=BEGIN EMITL(0); EMITL(EMITNO1)END#,        08289010 T  0003!083010
                       EMITTIME=BEGIN EMITN(2);  EMIT0(259); AEXP ;               08289020 T  0003!083010
                                      EMITPAIR(JUNK,ISN); EMIT0(965) END#;        08289030 T  0003!083010
PROCEDURE READSTMT;                                                               08290000 T  0003!083010
        BEGIN COMMENT READSTMT GENERATES CODE TO CALL INTERPTI(WHICH IS            08291000 T  0003!083010
              SHORT FOR INTERPRET INPUT) AN INTRINSIC PROCEDURE ON THE             08292000 T  0003!083010
              DRUM, PASSING TO IT PARAMETERS DETERMINED BY THE FORMAT OF           08293000 T  0003!083010
              THE READ OR SPACE STATEMENT.                                         08294000 T  0003!083010
                THE SPACE STATEMENT IS HANDLED AS A SPECIAL CASE OF READ           08295000 T  0003!083010
              STATEMENT WHERE ZERO WORDS ARE READ IN A FORWARD OR                  08296000 T  0003!083010
              REVERSE DIRECTION DEPENDING ON THE SIGN OF THE ARITHMETIC            08297000 T  0003!083010
              EXPRESSION IN THE SPACE STATEMENT.                                   08298000 T  0003!083010
                I HAVE LISTED BELOW THE VARIOUS CASES CONSIDERED BY THE            08299000 T  0003!083010
              READSTMT PROCEDURE AND THE CORRESPONDING PARAMETERS WHICH            08300000 T  0003!083010
              ARE PASSED TO INTERPTI.                                              08301000 T  0003!083010
              ************************************************************         08302000 T  0003!083010
              <DIRECTION INDICATOR>::=REVERSE/<EMPTY>                              08303000 T  0003!083010
              <FILE PART>::=<FILE IDENTIFIER>/<SUPERFILE DESIGNATOR>               08304000 T  0003!083010
              <RELEASE INDICATOR>::=[NO]/<EMPTY>                                   08305000 T  0003!083010
              <ACTION LABELS>::=[<END OF FILE LABEL>;<PARITY LABEL>]/              08306000 T  0003!083010
                                [<END OF FILE LABEL>]/[;<PARITY LABEL>]/           08307000 T  0003!083010
                                <EMPTY>                                            08308000 T  0003!083010
              CIMI IS THE CHARACTER MODE INPUT EDITING ROUTINE.                    08309000 T  0003!083010
              POWERSOFTEN IS A TABLE OF POWERS OF TEN USED FOR                     08310000 T  0003!083010
                        CONVERSION.                                                08311000 T  0003!083010
              FILE IS A DATA DESCRIPTOR DESCRIBING THE I/O DESCRIPTOR.             08312000 T  0003!083010
              ACTION TYPE IS A FOUR VALUED PARAMETER.  IT MAY BE + OR-.            08313000 T  0003!083010
              1 OR 2.  THE SIGN OF THE VALUE INDICATES FORWARD OR                  08314000 T  0003!083010
              REVERSE DIRECTION FOR + AND - RESPECTIVELY.  THE                     08315000 T  0003!083010
              VALUE IS ONE MORE THAN THE NUMBER OF RECORDS TO BE                   08316000 T  0003!083010
              PROCESSED.                                                           08317000 T  0003!083010
        END OF FILE LABEL IS A DATA DESCRIPTOR POINTING TO A LABEL                 08318000 T  0003!083010
                          DESCRIPTOR FOR END OF FILE JUMPS.                        08319000 T  0003!083010
        PARITY LABEL IS A DATA DESCRIPTOR POINTING TO A LABEL                      08320000 T  0003!083010
                    DESCRIPTOR FOR PARITY CONDITION JUMPS.                         08321000 T  0003!083010
        + OR - N IS SIMILAR TO ACTION TYPE.  IT CONTAINS THE EXACT                 08322000 T  0003!083010
                 DISTANCE AND DIRECTION TO    SPACE RATHER THAN ONE                08323000 T  0003!083010
                 GREATER THAN THE NUMBER OF RECORDS TO BE SPACED AS                08324000 T  0003!083010
                 IN ACTION TYPE.                                                   08325000 T  0003!083010
        LIST ROUTINE DESCRIPTOR IS AN ACCIDENTAL ENTRY PROGRAM                     08326000 T  0003!083010
                              DESCRIPTRO WHICH WILL EITHER RETURN                  08327000 T  0003!083010
                              AN ADDRESS OR VALUE DEPENDING ON                     08328000 T  0003!083010
                              THE CALL.                                            08329000 T  0003!083010
        N IS THE VALUE OF THE ARITHMETIC EXPRESSION IN READ STMT.                  08330000 T  0003!083010
        READ<DIRECTION INDICATOR>(<FILE PART><RELEASE INDICATOR>)                  08331000 T  0003!083010
            <ACTION LABELS>                                                        08332000 T  0003!083010
          -  -  -  -  -  -  -  -  -  -  -  -  -  -  -                              08333000 T  0003!083010
        (CIMI,POWERSOFTEN,FILE,ACTION TYPE,0,0,0,END OF FILE LABEL                 08334000 T  0003!083010
         ,PARITY LABEL)                                                            08335000 T  0003!083010
          ** ** ** ** ** ** ** ** ** ** ** ** ** ** **                             08336000 T  0003!083010
        READ<DIRECTION INDICATOR>(<FILE PART><RELEASE INDICATOR>,                  08337000 T  0003!083010
            <FORMAT PART>)<ACTION LABELS>                                          08338000 T  0003!083010
          -  -  -  -  -  -  -  -  -  -  -  -  -  -  -                              08339000 T  0003!083010
        (CIMI,POWERSOFTEN,FILE,ACTION TYPE,FORMAT INDEX,FORMAT                     08340000 T  0003!083010
         ARRAY DESCRIPTOR,0,END OF FILE LABEL,PARITY LABEL)                        08341000 T  0003!083010
          ** ** ** ** ** ** ** ** ** ** ** ** ** ** **                             08342000 T  0003!083010
        SPACE(<FILE PART>,<ARITHMETIC EXPRESSION>)<ACTION LABELS>                  08343000 T  0003!083010
          -  -  -  -  -  -  -  -  -  -  -  -  -  -  -                              08344000 T  0003!083010
```

```
(CIMI,POWERSOFTEN,FILE,+ OR - N,0,0,1,END OF FILE LABEL,                08345000 T  0003!083010
 PARITY LABEL)                                                          08346000 T  0003!083010
  **  **  **  **  **  **  **  **  **  **  **  **  **  **  **            08347000 T  0003!083010
READ<DIRECTION INDICATOR>(<FILE PART><RELEASE INDICATOR>,               08348000 T  0003!083010
     <FORMAT PART>,<LIST>)<ACTION LABELS>                               08349000 T  0003!083010
   -   -   -   -   -   -   -   -   -   -   -   -   -   -   -            08350000 T  0003!083010
(CIMI,POWERSOFTEN,FILE,ACTION TYPE,FORMAT INDEX,FORMAT                  08351000 T  0003!083010
 ARRAY DESCRIPTOR,LIST ROUTINE DESCRIPTOR,END OF FILE                   08352000 T  0003!083010
 LABEL,PARITY LABEL)                                                    08353000 T  0003!083010
  **  **  **  **  **  **  **  **  **  **  **  **  **  **  **            08354000 T  0003!083010
READ<DIRECTION INDICATOR>(<FILE PART><RELEASE INDICATOR>,               08355000 T  0003!083010
     *,<LIST>)<ACTION LABELS>                                           08356000 T  0003!083010
   -   -   -   -   -   -   -   -   -   -   -   -   -   -   -            08357000 T  0003!083010
(CIMI,POWERSOFTEN,FILE,ACTION TYPE,0,0,LIST ROUTINE                     08358000 T  0003!083010
 DESCRIPTOR,END OF FILE LABEL,PARITY LABEL)                             08359000 T  0003!083010
  **  **  **  **  **  **  **  **  **  **  **  **  **  **  **            08360000 T  0003!083010
READ<DIRECTION INDICATOR>(<FILE PART><RELEASE INDICATOR>,               08361000 T  0003!083010
     <ARITHMETIC EXPRESSION>,<ROW DESIGNATOR>)<ACTION                   08362000 T  0003!083010
     LABELS>                                                            08363000 T  0003!083010
   -   -   -   -   -   -   -   -   -   -   -   -   -   -   -            08364000 T  0003!083010
(CIMI,POWERSOFTEN,FILE,ACTION TYPE,0,N,ROW DESCRIPTOR,                  08365000 T  0003!083010
 END OF FILE LABEL,PARITY LABEL)                                        08366000 T  0003!083010
  **  **  **  **  **  **  **  **  **  **  **  **  **  **  **            08367000 T  0003!083010
READ<DIRECTION INDICATOR>(<FILE PART><RELEASE INDICATOR>,               08368000 T  0003!083010
     <MULTIPLYING OPERATOR>,<LIST>)<ACTION LABELS>                      08369000 T  0003!083010
   -   -   -   -   -   -   -   -   -   -   -   -   -   -   -            08370000 T  0003!083010
(CIMI,POWERSOFTEN,FILE,ACTION TYPE,1,0,LIST ROUTINE                     08371000 T  0003!083010
 DESCRIPTOR,END OF FILE LABEL,PARITY LABEL)                             08372000 T  0003!083010
**********************************************************;             08373000 T  0003!083010
DEFINE REVERSETOG = RRB1#;  COMMENT REVERSETOG IS SET TRUE              08374000 T  0003!083010
                                                  START OF SEGMENT **********   92
                            IF THE STATEMENT BEING COMPILED             08375000 T  0092!000010
                            IS A READ REVERSE, OTHERWISE IT             08376000 T  0092!000010
                            IS SET FALSE;                               08377000 T  0092!000010
LABEL EXIT;  COMMENT EXIT APPEARS AFTER THE LAST                        08378000 T  0092!000010
          EXECUTABLE STATEMENT IN READSTMT;                             08379000 T  0092!000010
LABEL CHKACTIONLABELS;  COMMENT THE CODE AT THIS LABEL                  08380000 T  0092!000010
                        ASSUMES I IS POINTING AT THE RIGHT              08381000 T  0092!000010
                        PARENTHESIS;                                    08382000 T  0092!000010
LABEL PASSLIST;  COMMENT THE CODE AT PASSLIST EXPECTS I TO               08383000 T  0092!000010
                 BE POINTING AT THE LAST QUANTITY IN THE                08384000 T  0092!000010
                 SECOND PARAMETER;                                      08385000 T  0092!000010
LABEL READXFORM;                                                        08385100 T  0092!000010
INTEGER LISTADDRESS; COMMENT TEMP TO HOLD LIST ADD DESC;                08385500 T  0092!000010
STACK(F+2) = LISTADDRESS
BOOLEAN SEEKTOG,LOCKTOG,GRABTOG;%                                       08385600 T  0092!000010

STACK(F+3) = SEEKTOG
STACK(F+4) = LOCKTOG
STACK(F+5) = GRABTOG
BOOLEAN MAYI;  COMMENT TRUE IF "FILE" IS ARRAY ROW;                     08385700 T  0092!000010

STACK(F+6) = MAYI
INTEGER HOLD;  COMMENT L MAY GET CUT BACK TO HERE;                      08385800 T  0092!000010

STACK(F+7) = HOLD
IF STEPI = LEFTPAREN                                                    08386000 T  0092!000010
THEN REVERSETOG←SEEKTOG←FALSE                                           08387000 T  0092!000011
ELSE BEGIN COMMENT THIS HAD BETTER SAY REVERSE;                         08388000 T  0092!000113
          REVERSETOG←ACCUM[1]="7REVER";                                 08389000 T  0092!000312
```

```
          LOCKTOG←ELCLASS=LOCKV;                                          08390000 T  00921000411
          SEEKTOG←ACCUM[1]="4SEEKO";                                      08390500 T  00921000610
          IF REVERSETOG OR LOCKTOG OR SEEKTOG THEN STEPIT                 08391000 T  00921000713
                                    ELSE BEGIN ERR(420);                  08392000 T  00921000912
                                               GO TO EXIT;                08393000 T  00921001313
                                         END;                             08394000 T  00921001410
          IF CHECK(LEFTPAREN,421)                                         08395000 T  00921001410
          THEN GO TO EXIT;                                                08396000 T  00921001411
           COMMENT ERROR 421 MEANS MISSING LEFT                           08397000 T  00921001610
          PARENTHESIS IN READ REVERSE STATEMENT;                          08398000 T  00921001610
     END;                                                                 08399000 T  00921001610
EMITO(MKS);                                                               08400000 T  00921001610
     IF STEPI ≥ BOOARRAYID AND ELCLASS ≤ INTARRAYID THEN                  08401000 T  00921001611
          BEGIN VARIABLE(FL);                                             08401020 T  00921001811
               IF TABLE(I-2) ≠ FACTOP THEN                                08401030 T  00921002010
                    BEGIN ERR(422); GO TO EXIT END;                       08401040 T  00921002113
               MAYI ← TRUE;  HOLD ← L;                                    08401045 T  00921002313
               EMIT(11); EMIT(4); EMITO(280);                             08401050 T  00921002512
               EMITPAIR(GNAT(POWERSOFTEN),LOD);                           08401060 T  00921002712
               EMITO(XCH); EMITL(0); EMITL(1);                            08401070 T  00921002811
          END ELSE                                                        08401080 T  00921003112
BEGIN                                                                     08401090 T  00921003112
EMITPAIR(GNAT(POWERSOFTEN),LOD);                                          08402000 T  00921003113
IF NOT RANGE(FILEID,SUPERFILEID)                                          08403000 T  00921003312
THEN BEGIN COMMENT ERROR 422 MEANS MISSING FILE IN READ                   08404000 T  00921003313
          STATEMENT;                                                      08405000 T  00921003411
          ERR(422);       GO TO EXIT;                                     08406000 T  00921003411
     END;                                                                 08407000 T  00921003610
PASSFILE;                                                                 08408000 T  00921003610
IF ELCLASS = LFTBRKET                                                     08409000 T  00921003611
THEN BEGIN %%% COMPILES CODE FOR [NS],[NS,*],[NS,<AEXP>],                 08410000 T  00921003611
          %%% [*],[*,*],[*,<AEXP>],[<AEXP>],[<AEXP>,*],                  08410010 T  00921003713
          %%% AND [<AEXP>,<AEXP>]. THE FIRST (LEFTMOST)                  08410020 T  00921003713
          %%% <AEXP> IS THE READSEEKDISTADDRESS, RESIDING                 08410030 T  00921003713
          %%% IN THE C-FIELD OF THE DSKADDR. THE SECOND                   08411000 T  00921003713
          %%% <AEXP> IS THE WAIT-TIME, RESIDING IN THE                    08411010 T  00921003713
          %%% F-FIELD OF THE DSKADDR, AND ALSO TURNING-ON                 08411020 T  00921003713
          %%% THE EXP-SIGN BIT OF DSKADDR,X"S ARE EMPTIES                 08411030 T  00921003713
          %%% IN THE ABOVE, NS = NO OR STOP,                              08411040 T  00921003713
     STEPIT;  %%% STEP OVER [, AND POINT AT NEXT ITEM.                    08412000 T  00921003713
     IF RR1←IF ACCUM[1]="2NOOOO" THEN 1 ELSE                              08412010 T  00921003810
          IF ACCUM[1]="4STOPO" THEN 2 ELSE                                08412020 T  00921004011
        0 ≠ 0 THEN %%% HAVE [NS                                           08412030 T  00921004211
          IF STEPI=COMMA THEN %%% HAVE [NS,                               08412040 T  00921004410
               IF STEPI=FACTOP THEN %%% HAVE [NS,*                        08412050 T  00921004513
                    BEGIN                                                 08412060 T  00921004712
                    IF RR1=1 THEN EMITNO(1)                               08412070 T  00921004713
                    ELSE BEGIN EMITL(1); EMITL(2) END ;                   08412080 T  00921004810
                    STEPIT ;                                              08412090 T  00921005411
                    END                                                   08413000 T  00921005512
               ELSE                                                       08413010 T  00921005512
                    IF ACCUM[1]="4LOCKO" THEN                             08413012 T  00921005512
                    BEGIN %%% [NS,LOCK                                    08413014 T  00921005611
                         EMITL(1); EMITD(47,4,1);                         08413016 T  00921005712
                         STEPIT;                                          08413018 T  00921005912
                    END ELSE                                              08413020 T  00921005913
                    BEGIN %%% HAVE [NS,AEXP                               08413022 T  00921005913
                    IF RR1=2 THEN EMITL(1) ;                              08413030 T  00921006112
```

```
                    EMITTIME ;                                          08413040 T  00921006312
                    IF RR1=2 THEN                                       08413050 T  00921006611
                         BEGIN EMITO(LOR); EMITL(2) END                 08413060 T  00921006713
                    ELSE EMITL(1) ;                                     08413080 T  00921006913
                    END                                                 08413090 T  00921007011
               ELSE IF RR1=1 THEN EMITNO(1) %%% ONLY HAVE [NS           08413100 T  00921007011
                    ELSE BEGIN EMITL(1); EMITL(2) END                   08413110 T  00921007410
          ELSE IF ELCLASS=FACTOP THEN %%% HAVE [*                       08413120 T  00921007610
                    IF STEPI=COMMA THEN %%% HAVE [*,                    08413130 T  00921007712
                         IF STEPI=FACTOP THEN %%% HAVE [*,*             08414000 T  00921007811
                              BEGIN EMITNO(2); STEPIT END               08414010 T  00921008010
                         ELSE IF ACCUM[1]="4LOCKO" THEN                 08414012 T  00921008211
                         BEGIN %%% [*,LOCK                              08414014 T  00921008410
                              EMITL(1); EMITD(47,4,1);                  08414016 T  00921008411
                              STEPIT;                                   08414018 T  00921008611
                         END ELSE                                       08414020 T  00921008712
                         BEGIN EMITTIME; EMITL(2); END % [*,A            08414022 T  00921008712
                    ELSE EMITNO(2) %%% HAVE ONLY [*                     08414030 T  00921009313
               ELSE BEGIN %%% HAVE [AEXP                                08415000 T  00921009513
                    AEXP;EMITO(SSP);EMITL(1);EMITO(ADD);                08415010 T  00921009610
                    IF SEEKTOG THEN EMITO(CHS) ;                        08415020 T  00921009811
                    EMITPAIR(JUNK,ISN) ;                                08415030 T  00921010010
                    IF ELCLASS=COMMA THEN %%% HAVE [AEXP,               08416000 T  00921010112
                         IF STEPI=FACTOP THEN STEPIT %%%[AEXP,*         08416010 T  00921010210
                         ELSE IF ACCUM[1]="4LOCKO" THEN                 08416012 T  00921010410
                         BEGIN %%% [AEXP,LOCK                           08416014 T  00921010610
                              EMITL(1); EMITD(47,4,1);                  08416016 T  00921010611
                              STEPIT;                                   08416018 T  00921010811
                              END ELSE                                  08416020 T  00921010912
                              BEGIN EMITTIME; EMITO(LOR) END ;          08416022 T  00921010912
                         EMITL(2) ; %%% ABOVE ELSE WAS [AEXP,AEXP       08416030 T  00921011513
                         END ;                                          08417000 T  00921011610
                    IF CHECK(RTBRKET,424) THEN GO EXIT ELSE STEPIT ;    08417010 T  00921011610
               END                                                      08418000 T  00921011810
          ELSE IF ELCLASS=LEFTPAREN THEN                                08418100 T  00921011811
          BEGIN STEPIT; AEXP; IF ELCLASS=COMMA THEN                     08418200 T  00921011913
               IF STEPI≠FACTOP THEN%                                    08418250 T  00921012210
               BEGIN AEXP; EMITPAIR(JUNK,ISN) END ELSE%                 08418300 T  00921012313
               BEGIN EMITL(1); GRABTOG←TRUE; STEPIT END ELSE            08418350 T  00921012513
                    EMITPAIR(0,LNG);                                    08418400 T  00921012810
               EMITD(33,33,15);                                         08418500 T  00921012913
               EMITO(IF LOCKTOG THEN SSN ELSE SSP);                     08418600 T  00921013011
               EMITL(REAL(SEEKTOG)); EMITD(33,18,15);                   08418650 T  00921013312
               IF CHECK(RTPAREN,104) THEN GO EXIT;                      08418700 T  00921013512
               EMITL(REAL(GRABTOG)+2); STEPIT;%                         08418800 T  00921013611
          END                                                           08418900 T  00921013811
          ELSE BEGIN EMITL(0); EMITL(2); END;                           08419000 T  00921013811
          IF REVERSETOG                                                 08420000 T  00921014011
          THEN EMITO(CHS);                                              08421000 T  00921014011
          END;                                                          08421500 T  00921014210
          IF ELCLASS = RTPAREN                                          08422000 T  00921014210
          THEN BEGIN COMMENT NO FORMAT,NO LIST CASE;                    08423000 T  00921014210
               EMITL(0);       EMITL(0);       EMITL(0);                08424000 T  00921014312
               GOGOGO ← NOT MAYI;%                                      08424100 T  00921014513
               GO CHKACTIONLABELS;                                      08425000 T  00921014611
          END;                                                          08426000 T  00921014712
          IF CHECK(COMMA,424)                                           08427000 T  00921014712
```

```
                    THEN GO TO EXIT;                                                    08428000 T  00921014713
                     COMMENT ERROR 424 MEANS IMPROPER FILE DELIMITER IN READ          08429000 T  00921014811
                    STATEMENT;                                                          08430000 T  00921014811
                    IF STEPJ = FACTOP                                                   08431000 T  00921014811
                    THEN BEGIN COMMENT *.LIST CASE;                                     08432000 T  00921014912
                         EMITL(0);       EMITL(0);       GO PASSLIST;                   08433000 T  00921015010
                         END;                                                           08434000 T  00921015210
                    IF ELCLASS = MULOP                                                  08435000 T  00921015210
                    THEN BEGIN COMMENT FREE FIELD FORMAT CASE;                          08436000 T  00921015211
                         IF STEPI=MULOP THEN EMITL(2) ELSE                              08437000 T  00921015313
                              BEGIN EMITL(1); I←I-1; END ;                              08437050 T  00921015513
                         EMITL(0);  GO TO PASSLIST ;                                    08437075 T  00921015810
                         END;                                                           08438000 T  00921015913
                    IF RANGE(FRMTID,SUPERFRMTID)                                        08439000 T  00921015913
                    THEN BEGIN COMMENT THE SECOND PARAMETER IS A FORMAT;                08440000 T  00921016010
                              PASSFORMAT;                                               08441000 T  00921016112
READXFORM:                    IF TABLE(I+1) = COMMA                                     08442000 T  00921016113
                              THEN GO PASSLIST;                                         08443000 T  00921016312
                              STEPIT;                                                   08444000 T  00921016410
                              IF CHECK(RTPAREN,425)                                     08445000 T  00921016411
                              THEN GO TO EXIT;                                          08446000 T  00921016512
                               COMMENT ERROR 425 MEANS IMPROPER FORMAT                  08447000 T  00921016611
                              DELIMITER IN READ STATEMENT;                              08448000 T  00921016611
                              EMITL(0);       GO CHKACTIONLABELS;                       08449000 T  00921016611
                         END;                                                           08450000 T  00921016713
                    IF Q:=ACCUM[1]="1<0000" THEN                                        08450010 T  00921016713
                         BEGIN EXPLICITFORMAT; GO TO READXFORM; END;                    08450020 T  00921016912
                    IF MAYI THEN                                                        08450100 T  00921017210
                    BEGIN KLUDGE(HOLD);                                                 08450200 T  00921017210
                         GO TO EXIT;                                                    08450300 T  00921017313
                    END ARRAY TO ARRAY CASE;                                            08450400 T  00921017410
                    EMITL(0);           AEXP;                                           08451000 T  00921017410
                    IF CHECK(COMMA,426)                                                 08452000 T  00921017513
                    THEN GO TO EXIT;                                                    08453000 T  00921017610
                     COMMENT ERROR 426 MEANS IMPROPER DELIMITER FOR SECOND              08454000 T  00921017712
                    PARAMETER;                                                          08455000 T  00921017712
                    STEPIT;                                                             08456000 T  00921017712
                    IF RANGE(BOOARRAYID,INTARRAYID)                                     08457000 T  00921017713
                    THEN BEGIN COMMENT THIS IS THE ROW DESIGNATOR CASE;                 08458000 T  00921017810
                              VARIABLE(FL);                                             08459000 T  00921017912
                              IF TABLE(I-2) ≠ FACTOP                                    08460000 T  00921018010
                              THEN BEGIN COMMENT ERROR 427 MEANS IMPROPER               08461000 T  00921018112
                                        ROW DESIGNATOR IN READ;                         08462000 T  00921018210
                                        ERROR(427); GO TO EXIT;                         08463000 T  00921018210
                                   END;                                                 08464000 T  00921018313
                              IF CHECK(RTPAREN,428)                                     08465000 T  00921018313
                              THEN GO TO EXIT;                                          08466000 T  00921018410
                               COMMENT ERROR 428 MEANS IMPROPER ROW DESIGNATOR          08467000 T  00921018512
                              DELIMITER IN READ STATEMENT;                              08468000 T  00921018512
                              GOGOGO ← TRUE;%                                           08468100 T  00921018512
                              GO CHKACTIONLABELS;                                       08469000 T  00921018610
                         END                                                            08470000 T  00921018611
                    ELSE BEGIN COMMENT ERROR 429 MEANS MISSING ROW DESIGNATOR;          08471000 T  00921018611
                              ERROR(429);     GO TO EXIT;                               08472000 T  00921018712
                         END;                                                           08473000 T  00921018810
PASSLIST:STEPIT;                                                                        08474000 T  00921018810
                    IF CHECK(COMMA,430)                                                 08475000 T  00921018913
                    THEN GO TO EXIT;                                                    08476000 T  00921019010
```

```
          COMMENT ERROR 430 MEANS IMPROPER DELIMITER PRECEEDING                08477000 T  0092:019112
          THE LIST IN A READ STATEMENT;                                        08478000 T  0092:019112
     IF STEPI ≠ LISTID AND ELCLASS ≠ SUPERLISTID                               08479000 T  0092:019112
          THEN BEGIN                                                           08480000 T  0092:019211
               RR←LISTGEN;                                                     08481000 T  0092:019313
               I←I-1;                                                          08482000 T  0092:019411
               GO TO CHKACTIONLABELS                                           08483000 T  0092:019610
           END;                                                                08484000 T  0092:019611
     CHECKER(ELBAT[I]);                                                        08484500 T  0092:019611
     IF ELCLASS = SUPERLISTID THEN                                             08485000 T  0092:019713
          BEGIN    COMMENT  SUBSCRIPTED SWITCH LIST ID;                       08486000 T  0092:019810
               LISTADDRESS ←ELBAT[I],ADDRESS;                                  08488000 T  0092:019811
               BANA;                                                           08489000 T  0092:020010
               EMITV(LISTADDRESS);                                             08489500 T  0092:020011
               IF LISTADDRESS > 1023 THEN EMITO(PRTE);                         08489510 T  0092:020113
               EMITO(LOD); I←I-1  END                                          08489520 T  0092:020313
          ELSE   BEGIN      COMMENT A COMMON LIST;                           08489530 T  0092:020513
               EMITPAIR (ELBAT[I],ADDRESS,LOD);                                08489550 T  0092:020610
               END;                                                            08489560 T  0092:020713
     STEPIT;                                                                   08489570 T  0092:020713
     IF CHECK(RTPAREN,449) THEN GO TO EXIT;                                    08489580 T  0092:020810
     COMMENT  449  IS  IMPROPER LIST DELIMETER IN READ STATEMENT;              08489590 T  0092:021010
                                                                               08490000 T  0092:021010
CHKACTIONLABELS:HANDLETHETAILENDOFAREADORSPACESTATEMENT;                        08491000 T  0092:021010
     EXIT:;                                                                    08492000 T  0092:021011
     END READSTMT;                                                             08493000 T  0092:021112
```

92 IS 215 LONG, NEXT SEG 3

```
          REAL PROCEDURE FILEATTRIBUTEINDX(T) ; % RETURNS A ZERO IF THE NEXTSCANND   08493010 T  0003:083010
PRT(1036) = FILEATTRIBUTEINDX
          VALUE T;  BOOLEAN T ;                % ITEM IS NOT A FILE ATTRIBUTE,      08493015 T  0003:083010
              BEGIN                            % RETURNS THE ASSOCIATED INDEX IF   08493020 T  0003:083010
              REAL I ;                         % IT IS A FILE ATTRIBUTE,            08493030 T  0003:083010
                                                                          START OF SEGMENT **********  93
STACK(F+3) = I
              LABEL EXIT ;                                                         08493040 T  0093:000010
              STOPDEFINE←T ; % MAY DISALLOW DEFINES IN FILE-ATTRIBUTE PART,        08493050 T  0093:000010
              STEPIT ; % NOW POINTED AT ATTRIBUTE (STEPIT TURNS OFF STOP DEFINE),  08493060 T  0093:000011
              IF I←FILEATTRIBUTES[0]=0 THEN                                        08493070 T  0093:000112
                   BEGIN                                                           08493080 T  0093:000211
                   FILL FILEATTRIBUTES[*] WITH % NON-ASSGNBL ATTRBTS HAVE ,[1:1]=1 08493090 T  0093:000312
                                               % BOOLEAN ATTRIBUTES HAVE ,[2:1]=1, 08493091 T  0093:000313
                                               % ALPHA ATTRIBUTES HAVE ,[3:1]=1.   08493092 T  0093:000313
                        % THIS NEXT NUMBER IS THE CURRENT # OF FILE ATTRIBUTES:    08493093 T  0093:000313
                        17                                                         08493094 T  0093:000313
                        ,"6ACCES"%***ANY ADDITIONAL ATTRIBUTES MUST BE INSERTED***  08493095 T  0093:000313
                                                                          START OF SEGMENT **********  94
                        ,"5MYUSE"%******IMMEDIATELY AFTER THE LAST ATTRIBUTE******  08493096 T  0093:000512
                        ,"4SAVE0"                                                  08493097 T  0093:000512
                        ,"80THER" % "0THERUSE",                                    08493098 T  0093:000512
                        ,"404MFID0"                                                08493099 T  0093:000512
                        ,"403FID00"                                                08493100 T  0093:000512
                        ,"4REEL0"                                                  08493101 T  0093:000512
                        ,"4DATE0"                                                  08493102 T  0093:000512
```

```
                ,"5CYCLE"                                                                08493103 T  0093!000512
                ,"4TYPEO"                                                                08493104 T  0093!000512
                ,"5AREAS"                                                                08493105 T  0093!000512
                ,"8AREAS" % "ARFASIZE",                                                  08493106 T  0093!000512
                ,"2EU000"                                                                08493107 T  0093!000512
                ,"5SPEED"                                                                08493108 T  0093!000512
                ,"9TIMEL" % "TIMELIMIT"                                                  08493109 T  0093!000512
                ,"+081OSTA" % "!OSTATUS"                                                 08493110 T  0093!000512
                ,"9SENSI" % "SENSITIVE"                                                  08493111 T  0093!000512
                % THIS CARD MERELY OCCUPIES A SEQUENCE NUMBER,                           08493120 T  0093!000512
                ; % END OF FILL STATEMENT,                                               08493130 T  0093!000512
                                                                                 94 IS   18 LONG, NEXT SEG   93
          I←FILEATTRIBUTES[0] ;                                                          08493140 T  0093!000512
          END ;                                                                          08493150 T  0093!000610
     FOR I←1 STEP -1 UNTIL 1 DO IF FILEATTRIBUTES[I].[12:36]=Q THEN                      08493160 T  0093!000610
          BEGIN FILEATTRIBUTEINDX←I; GO EXIT END ;                                       08493170 T  0093!000811
EXIT:                                                                                    08493180 T  0093!001211
     END OF FILEATTRIBUTEINDX ;                                                          08493190 T  0093!001312
                                                                                 93 IS   16 LONG, NEXT SEG    3



COMMENT FILEATTRIBUTEHANDLER HANDLES FILE ATTRIBUTE STUFF, IT CONSTRUCTS                 08493200 T  0003!083010
        A CALL ON FILEATTRIBUTES INTRINSIC.IT IS CALLED BY 5 PROCEDURES:                 08493210 T  0003!083010
           1, STMT: PASSES N=FS, AND TELLS FAH TO EXPECT AN ASSIGNOP,                    08493220 T  0003!083010
                    FAH WILL TELL FILEATTRIBUTES TO CHANGE THE ATTRIBUTE                 08493230 T  0003!083010
                    AND XIT,                                                             08493240 T  0003!083010
           2, ACTUALPARAPART:                                                            08493250 T  0003!083010
                    PASSES N=FA, AND TELLS FAH THAT THE FILE DESC HAS                    08493260 T  0003!083010
                    ALREADY BEEN EMITTED,  IT ALSO TELLS FAH TO LEAVE                    08493270 T  0003!083010
                    THE VALUE OF THE ATTRIBUTE IN THE TOP OF THE STACK,                  08493280 T  0003!083010
                3, PRIMARY:                                                              08493290 T  0003!083010
                    PASSES N=FP, AND TELLS FAH TO HANDLE AN ASSIGNOP                     08493300 T  0003!083010
                    IF THERE IS ONE (BY CALLING AEXP OR BEXP, DEPENDING                  08493310 T  0003!083010
                    ON THE TYPE OF ATTRIBUTE) OR JUST TO EMIT A ZERO, IF                 08493320 T  0003!083010
                    THERE IS AN ASSIGNOP, THEN FAH TELLS FILEATTRIBUTES                  08493330 T  0003!083010
                    TO BOTH CHANGE THE ATTRIBUTE AND LEAVE THE VALUE                     08493340 T  0003!083010
                    IN THE TOP OF THE STACK, OTHERWISE, FAH TELLS FILE-                  08493350 T  0003!083010
                    ATTRIBUTES TO ONLY LEAVE THE VALUE OF THE REQUIRED                   08493360 T  0003!083010
                    ATTRIBUTE IN THE TOP OF THE STACK, IN ALL CASES,                     08493370 T  0003!083010
                    FAH WILL RETURN THE TYPE OF ATTRIBUTE COMPILED                       08493380 T  0003!083010
                    (ATYPE OR BTYPE),                                                    08493390 T  0003!083010
                4, BOOPRIM:                                                              08493400 T  0003!083010
                    PASSES N=FP, AND DOES THE SAME AS #3 (ABOVE),                        08493410 T  0003!083010
                5, IODEC:                                                                08493420 T  0003!083010
                    PASSES N=FIO, AND TELLS FAH THAT A MKS & FILE DESC                   08493430 T  0003!083010
                    HAVE ALREADY BEEN EMITTED, THE ATTRIBUTEINDX IS                      08493440 T  0003!083010
                    DETERMINED BY IODEC, AND IS PASSED VIA GT1,                          08493450 T  0003!083010
          END OF COMMENT ;                                                               08493460 T  0003!083010
          INTEGER PROCEDURE FILEATTRIBUTEHANDLER(N); VALUE N; REAL N ;                   08493470 T  0003!083010
              BEGIN                                                                      08493480 T  0003!083010
              REAL ATTRIBUTEINDX ;                                                       08493490 T  0003!083010
                                                                              START OF SEGMENT **********  95
STACK(F+3) = ATTRIBUTEINDX
              BOOLEAN ASSOP ;                                                            08493500 T  0095!000010
STACK(F+4) = ASSOP
```

```
    LABEL DONESOME,DONEMORE,EXIT ;                                          08493510 T  00951000010
    IF N=FA THEN GO TO DONESOME ELSE IF N=FIO THEN                          08493520 T  00951000010
         BEGIN ATTRIBUTEINDX←GT1; IF STEPI≠RELOP THEN I←I-1; ASSOP←TRUE;    08493530 T  00951000210
         EMITL(0); EMITL(0); %%% DUM1 PARAMETER...FOR POSSIBLE FUTR USE,    08493540 T  00951000610
         GO TO DONEMORE ;                                                   08493550 T  00951000810
         END ;                                                              08493560 T  00951000811
    EMIT0(MKS);  PASSFILE ; % MARK THE STACK & STACK A FILE DESCRIPTOR,     08493570 T  00951000811
    IF ELCLASS≠PERIOD THEN ERR(290) ELSE                                    08493580 T  00951001010
         BEGIN                                                              08493590 T  00951001210
DONESOME:                                                                   08493600 T  00951001211
         IF ATTRIBUTEINDX←FILEATTRIBUTEINDX(TRUE)=0 THEN ERR(291) ELSE      08493610 T  00951001312
              BEGIN                                                         08493620 T  00951001610
STEPIT;IF FALSE THEN BEGIN COMMENT$$DELETE THIS CARD TO GET ACTION LABEL    08493625 T  00951001611
              IF STEPI=LFTBRKET THEN                                        08493630 T  00951001810
                   BEGIN                                                    08493640 T  00951001810
                   STEPIT;DEXP;IF CHECK(RTBRKET,433)THEN GO EXIT;STEPIT;    08493650 T  00951001810
                   END                                                      08493660 T  00951002010
              ELSE EMITL(0) ;                                               08493670 T  00951002011
              EMITL(0) ; %%% DUM1 PARAMETER...FOR POSSIBLE FUTURE USE,      08493675 T  00951002210
              IF ASSOP←ELCLASS=ASSIGNOP THEN                                08493680 T  00951002211
                   BEGIN                                                    08493700 T  00951002410
IF N≠FS THEN FLAG(295);%**DELETE THIS CARD TO ALLOW GENRL FILATT ASSGNMT    08493705 T  00951002411
 DONEMORE:         IF BOOLEAN(FILEATTRIBUTES[ATTRIBUTEINDX],[1:1])          08493710 T  00951002611
                   THEN FLAG(293) ;                                         08493720 T  00951002712
                   STEPIT ;                                                 08493730 T  00951002912
                   IF BOOLEAN(FILEATTRIBUTES[ATTRIBUTEINDX],[2:1])          08493740 T  00951002913
                   THEN BEXP ELSE AEXP ;                                    08493750 T  00951003010
                   END                                                      08493760 T  00951003211
              ELSE IF N=FS THEN BEGIN ERR(292); GO EXIT END                 08493770 T  00951003211
              ELSE EMITL(0) ;                                               08493780 T  00951003513
              EMITNUM(IF ATTRIBUTEINDX= 1 THEN "6ACCESS" ELSE               08493790 T  00951003712
              IF ATTRIBUTEINDX= 4 THEN "6OTHRUS" ELSE                       08493795 T  00951003912
              IF ATTRIBUTEINDX=12 THEN "6ARASIZ" ELSE                       08493800 T  00951004112
              IF ATTRIBUTEINDX=15 THEN "6TIMLMT" ELSE                       08493805 T  00951004312
              IF ATTRIBUTEINDX=16 THEN "6IOSTAT" ELSE                       08493810 T  00951004512
              IF ATTRIBUTEINDX=17 THEN "6SNSTIV" ELSE                       08493812 T  00951004712
              0 & FILEATTRIBUTES[ATTRIBUTEINDX][6:12:36]                    08493820 T  00951004912
              & FILEATTRIBUTES[ATTRIBUTEINDX][1:3:1]) ;                     08493830 T  00951005011
            , EMITL((ATTRIBUTEINDX-1) & REAL(N=FP OR N=FA)[39:47:1]         08493840 T  00951005210
                   & REAL(ASSOP)[38:47:1]) ;                                08493850 T  00951005513
              EMITPAIR(GNAT(POWERSOFTEN),LOD); EMITV(GNAT(FILATTINT)) ;     08493860 T  00951005712
              FILEATTRIBUTEHANDLER←                                         08493870 T  00951005913
                   IF BOOLEAN(FILEATTRIBUTES[ATTRIBUTEINDX],[2:1])          08493880 T  00951005913
                   THEN BTYPE ELSE ATYPE ;                                  08493890 T  00951006010
              END ;                                                         08493900 T  00951006211
         END ;                                                              08493910 T  00951006211
    EXIT:                                                                   08493920 T  00951006211
    END OF FILEATTRIBUTEHANDLER ;                                           08493930 T  00951006312
                                                                95 IS   72 LONG, NEXT SEG    3


PROCEDURE SPACESTMT;                                                        08494000 T  00031083010
         BEGIN COMMENT THE SPACE STATEMENT IS BEST THOUGHT OF AS A          08495000 T  00031083010
              SUBSET OF THE READ STATEMENT WHERE ZERO WORDS ARE READ,       08496000 T  00031083010
              FOR THE EXACT SYNTAX FOR THE SPACE STATEMENT AND THE          08497000 T  00031083010
```

```
        PARAMETERS PASSED TO THE INTERPT ROUTINE SEE THE COMMENTS          08498000 T  0003!083!0
        FOR THE READ STATEMENT;                                            08499000 T  0003!083!0
        LABEL EXIT;  COMMENT EXIT APPEARS AFTER THE LAST                   08500000 T  0003!083!0
                                                                  START OF SEGMENT **********   96
                    EXECUTABLE STATEMENT IN SPACESTMT;                     08501000 T  0096!000!0
        STEPIT;                                                            08502000 T  0096!000!0
        IF CHECK(LEFTPAREN,434)                                            08503000 T  0096!000!1
        THEN GO TO EXIT;                                                   08504000 T  0096!000!2
          COMMENT ERROR 434 MEANS MISSING LEFT PARENTHESIS IN              08505000 T  0096!000!0
        SPACE STATEMENT;                                                   08506000 T  0096!000!0
        STEPIT;                                                            08507000 T  0096!000!0
        IF NOT RANGE(FILEID,SUPERFILEID)                                   08508000 T  0096!000!1
        THEN BEGIN COMMENT ERROR 435 MEANS IMPROPER FILE                   08509000 T  0096!000!2
                  IDENTIFIER IN SPACE STATEMENT;                           08510000 T  0096!000!1
                  ERROR(435);     GO TO EXIT;                              08511000 T  0096!000!1
             END;                                                          08512000 T  0096!000!3
        EMITO(MKS);                                                        08513000 T  0096!000!3
        EMITPAIR(GNAT(                                                     08514000 T  0096!000!1
        POWERSOFTEN),LOD); PASSFILE;                                       08515000 T  0096!000!2
        EMITL(0);                                                          08515100 T  0096!000!1
        IF CHECK(COMMA,436)                                                08516000 T  0096!000!2
        THEN GO TO EXIT;                                                   08517000 T  0096!000!3
         COMMENT ERROR 436 MEANS MISSING COMMA IN SPACE STATEMENT;         08518000 T  0096!001!2
        STEPIT;                AEXP;                                       08519000 T  0096!001!2
        IF CHECK(RTPAREN,437)                                              08520000 T  0096!001!0
        THEN GO TO EXIT;                                                   08521000 T  0096!001!1
         COMMENT ERROR 437 MEANS MISSING RIGHT PARENTHESIS IN              08522000 T  0096!001!3
        SPACE STATEMENT;                                                   08523000 T  0096!001!3
        EMITL(0);            EMITL(0);           EMITL(1);                 08524000 T  0096!001!3
        HANDLETHETAILENDOFAREADORSPACESTATEMENT;                           08525000 T  0096!001!0
EXIT:;                                                                     08526000 T  0096!001!1
END SPACESTMT;                                                             08527000 T  0096!001!2
                                                        96 IS   18 LONG, NEXT SEG   3
```

```
PROCEDURE WRITESTMT;                                                       08528000 T  0003!083!0
        BEGIN COMMENT WRITESTMT GENERATES CODE TO CALL INTERPTO, AN         08529000 T  0003!083!0
             INTRINSIC PROCEDURE ON THE DRUM, PASSING TO IT PARAMETERS     08530000 T  0003!083!0
             DETERMINED BY THE FORMAT OF THE WRITE STATEMENT,              08531000 T  0003!083!0
               I HAVE LISTED BELOW THE VARIOUS CASES CONSIDERED BY THE     08532000 T  0003!083!0
             WRITESTMT PROCEDURE AND THE CORRESPONDING PARAMETERS WHICH    08533000 T  0003!083!0
             ARE PASSED TO INTERPTO,                                       08534000 T  0003!083!0
             ********************************************************      08535000 T  0003!083!0
               FOR AN EXPLANATION OF THE PARAMETERS AND SYNTACTICAL        08536000 T  0003!083!0
             UNITS NOT DESCRIBED HERE, SEE THE COMMENTS FOR THE            08537000 T  0003!083!0
             READSTMT ROUTINE,                                             08538000 T  0003!083!0
             <CARRIAGE CONTROL>::= [DBL]/[PAGE]/[NO]/<ARITHMETIC           08539000 T  0003!083!0
                                  EXPRESSION>/<EMPTY>                      08540000 T  0003!083!0
             CHARI IS THE CHARACTER MODE OUTPUT EDITING ROUTINE SIMILAR    08541000 T  0003!083!0
                  TO CIMI FOR INPUT,                                       08542000 T  0003!083!0
             <CARRIAGE>   <EMPTY> [DBL] [PAGE] [NO] <ARITHMETIC EXP>       08543000 T  0003!083!0
             CHANNEL SKIP    0      0     0     0    EXPRESSIONS VALUE     08544000 T  0003!083!0
             LINESKIP        1      2     4     8          0               08545000 T  0003!083!0
             WRITE(<FILE PART><CARRIAGE CONTROLS>)/                        08546000 T  0003!083!0
                -  -  -  -  -  -  -  -  -  -  -  -  -  -  -                08547000 T  0003!083!0
             (CHARI,POWERSOFTEN,FILE,CHANNEL SKIP,LINE SKIP,0,0,0)         08548000 T  0003!083!0
```

```
  **  **  **  **  **  **  **  **  **  **  **  **  **  **                08549000 T  00031083010
WRITE(<FILE PART><CARRIAGE CONTROL>,<FORMAT PART>)/                     08550000 T  00031083010
  -   -   -   -   -   -   -   -   -   -   -   -                         08551000 T  00031083010
(CHARI,POWERSOFTEN,FILE,CHANNEL SKIP,LINE SKIP,FORMAT                   08552000 T  00031083010
 INDEX,FORMAT ARRAY DESCRIPTOR,0)                                       08553000 T  00031083010
  **  **  **  **  **  **  **  **  **  **  **  **  **  **  **            08554000 T  00031083010
WRITE(<FILE PART><CARRIAGE CONTROL>,<FORMAT PART>,<LIST>)/              08555000 T  00031083010
  -   -   -   -   -   -   -   -   -   -   -   -   -                     08556000 T  00031083010
(CHARI,POWERSOFTEN,FILE,CHANNEL SKIP,LINE SKIP,FORMAT                   08557000 T  00031083010
 INDEX,FORMAT ARRAY DESCRIPTOR,LIST ROUTINE DESCRIPTOR)                 08558000 T  00031083010
  **  **  **  **  **  **  **  **  **  **  **  **  **  **  **            08559000 T  00031083010
WRITE(<FILE PART><CARRIAGE CONTROL>,*,<LIST>)/                          08560000 T  00031083010
  -   -   -   -   -   -   -   -   -   -   -   -   -                     08561000 T  00031083010
(CHARI,POWERSOFTEN,FILE,CHANNEL SKIP,LINE SKIP,0,0,LIST                 08562000 T  00031083010
 ROUTINE DESCRIPTOR)                                                    08563000 T  00031083010
  **  **  **  **  **  **  **  **  **  **  **  **  **  **  **            08564000 T  00031083010
WRITE(<FILE PART>(CARRIAGE CONTROL>,<ARITHMETIC EXPRESSION              08565000 T  00031083010
     >,<ROW DESIGNATOR>)                                                08566000 T  00031083010
  -   -   -   -   -   -   -   -   -   -   -   -   -   -   -             08567000 T  00031083010
(CHARI,POWERSOFTEN,FILE,CHANNEL SKIP,LINE SKIP,0,N,ARRAY                08568000 T  00031083010
 ROW DESCRIPTOR)                                                        08569000 T  00031083010
  **  **  **  **  **  **  **  **  **  **  **  **  **  **  **  **;        08570000 T  00031083010
LABEL EXIT;  COMMENT EXIT APPEARS AFTER THE LAST                        08571000 T  00031083010
                                                          START OF SEGMENT **********   97
          EXECUTABLE STATEMENT IN WRITESTMT;                            08572000 T  00971000010
LABEL CHKSECOND;  COMMENT I IS NOW POINTING AT THE COMMA                08573000 T  00971000010
                  SEPARATING THE FIRST AND SECOND                       08574000 T  00971000010
                  PARAMETERS;                                           08575000 T  00971000010
LABEL ONEPARENSH;  COMMENT I IS POINTING AT THE RIGHT                   08576000 T  00971000010
                   PARENTHESIS AT THIS POINT AND I HAVE                 08577000 T  00971000010
                   JUST DISCOVERED THAT THIS IS THE ONE                 08578000 T  00971000010
                  PARAMETER CASE;                                       08579000 T  00971000010
          DEFINE ACCUM1 = RR1#;  COMMENT ACCUM1 IS USED AS A            08580000 T  00971000010
                                    TEMPORARY CELL FOR ACCUM[1];        08581000 T  00971000010
     %VOID                                                              08582000 T  00971000010
     %VOID                                                              08583000 T  00971000010
     %VOID                                                              08584000 T  00971000010
     %VOID                                                              08585000 T  00971000010
          LABEL PASSLIST;  COMMENT I IS POINTING AT THE COMMA           08586000 T  00971000010
                           PRECEEDING THE LIST WHEN THIS LABEL IS       08587000 T  00971000010
                           REACHED;                                     08588000 T  00971000010
          LABEL EMITCALL;  COMMENT I IS POINTING AT THE STATEMENT       08589000 T  00971000010
                           DELIMITER.  THE CODE AT EMITCALL EMITS THE   08590000 T  00971000010
                           CODE TO CALL INTERPTO;                       08591000 T  00971000010
          LABEL CHKRTPAREN;                                             08591100 T  00971000010
          LABEL WRITXFORM;                                              08591200 T  00971000010
        INTEGER LISTADDRESS; COMMENT TEMP TO HOLD LIST ADD DESC;        08591500 T  00971000010
STACK(F+2) = LISTADDRESS
        BOOLEAN LOCKTOG,ARC;                                            08591600 T  00971000010
STACK(F+3) = LOCKTOG
STACK(F+4) = ARC
        INTEGER HOLD;%                                                  08591700 T  00971000010
STACK(F+5) = HOLD
          IF (LOCKTOG*STEPI=LOCKV) THEN STEPIT;                         08592000 T  00971000010
          IF CHECK(LEFTPAREN,438)                                       08593000 T  00971000211
          THEN GO TO EXIT;                                              08594000 T  00971000312
           COMMENT ERROR 438 MEANS MISSING LEFT PARENTHESIS IN A        08595000 T  00971000410
```

```
WRITE STATEMENT;                                                                08596000 T  0097:000410
     EMITO(MKS);                                                                08597000 T  0097:000410
     IF STEPI ≥ BOOARRAYID AND ELCLASS ≤ INTARRAYID THEN                        08597100 T  0097:000512
          BEGIN VARIABLE(FL);                                                   08597200 T  0097:000712
               IF TABLE(I-2) ≠ FACTOP THEN                                      08597300 T  0097:000810
                    BEGIN ERR(439); GO TO EXIT END;                             08597400 T  0097:001010
               ARC ← TRUE;  HOLD ← L;                                           08597450 T  0097:001113
               EMIT(11); EMIT(4); EMITO(280);                                   08597500 T  0097:001312
               EMITPAIR(GNAT(POWERSOFTEN),LOD);                                 08597600 T  0097:001513
               EMITO(XCH);                                                      08597700 T  0097:001712
          END ELSE                                                              08597800 T  0097:001713
BEGIN                                                                           08597900 T  0097:001713
IF NOT RANGE(FILEID,SUPERFILEID)                                                08598000 T  0097:001810
THEN BEGIN COMMENT ERROR 439 MEANS IMPROPER FILE                                08599000 T  0097:001811
          IDENTIFIER IN A WRITE STATEMENT;                                      08600000 T  0097:002010
          ERR(439);      GO TO EXIT;                                            08601000 T  0097:002010
     END;                                                                       08602000 T  0097:002112
                                                                                08603000 T  0097:002112
EMITPAIR(GNAT(                                                                  08604000 T  0097:002112
POWERSOFTEN),LOD); PASSFILE;                                                    08605000 T  0097:002113
END;                                                                            08605500 T  0097:002312
IF(RRB1←ELCLASS = COMMA) OR ELCLASS = RTPAREN                                   08606000 T  0097:002312
THEN BEGIN COMMENT STANDARD CARRIAGE CONTROL CASE;                              08607000 T  0097:002411
          EMITL(0);      EMITL(1);                                              08608000 T  0097:002610
          IF RRB1                                                               08609000 T  0097:002713
          THEN GO CHKSECOND;                                                    08610000 T  0097:002713
ONEPARENSH:STEPIT;          EMITL(0);       EMITL(0);                           08611000 T  0097:002811
          GOGOGO ← NOT ARC;%                                                    08611100 T  0097:003112
          EMITL(0);       GO EMITCALL;                                          08612000 T  0097:003210
     END;                                                                       08613000 T  0097:003312
IF ELCLASS=LEFTPAREN THEN                                                       08613100 T  0097:003312
BEGIN STEPIT; AEXP; EMITO(IF LOCKTOG THEN SSN ELSE SSP);                        08613200 T  0097:003410
     IF ELCLASS=COMMA THEN BEGIN STEPIT; AEXP END ELSE                          08613300 T  0097:003713
          EMITPAIR(0,LNG);                                                      08613400 T  0097:004010
     EMITD(33,33,15); EMIT(0);                                                  08613500 T  0097:004113
     IF CHECK(RTPAREN,104) THEN GO EXIT ELSE GO CHKRTPAREN                      08613600 T  0097:004313
END;                                                                            08613700 T  0097:004411
IF CHECK(LFTBRKET,440)                                                          08614000 T  0097:004513
THEN GO TO EXIT;                                                                08615000 T  0097:004610
 COMMENT ERROR 440 MEANS IMPROPER DELIMITER FOR FIRST                           08616000 T  0097:004712
PARARAMETER IN A WRITE STATEMENT;                                               08617000 T  0097:004712
STEPIT;                                                                         08618000 T  0097:004712
%%% THE FOLLOWING CODE COMPILES CODE FOR [DPN],[DPN,*],                         08619000 T  0097:004713
%%% [DPN,<AEXP>],[*],[*,*],[*,<AEXP>],[<AEXP>],[<AEXP>,*]                      08619010 T  0097:004713
%%% AND [<AEXP>,<AEXP>], WHERE DPN IS STOP, DBL, PAGE, OR                      08619020 T  0097:004713
%%% NO.  THE FIRST (LEFTMOST) <AEXP> IS THE CHANNELSKIP,                       08619030 T  0097:004713
%%% RIGHT JUSTIFIED TO ITS C-FIELD.  THE SECOND <AEXP> IS                      08619040 T  0097:004713
%%% THE WAIT-TIME, RESIDING IN THE F-FIELD OF CHANNELSKIP.                     08619050 T  0097:004713
%%% AND ALSO TURNING ON THE EXP-SIGN BIT OF CHANNELSKIP.                       08619060 T  0097:004713
%%% *"S ARE CONSIDERED TO BE EMPTIES.                                          08619070 T  0097:004713
IF ACCUM1←IF ACCUM1←ACCUM[1]="3DBL00" THEN 2 ELSE                               08619080 T  0097:004713
          IF ACCUM1="4PAGE0" THEN 4 ELSE                                        08619090 T  0097:005011
          IF ACCUM1="4STOP0" THEN 16 ELSE                                       08619095 T  0097:005211
          IF ACCUM1="2N0000" THEN 8 ELSE 0≠0 THEN %%% [DPN                      08620000 T  0097:005411
     IF STEPI=COMMA THEN %%% HAVE [DPN,                                         08620010 T  0097:005713
          IF STEPI=FACTOP THEN %%% HAVE [DPN,*                                  08620020 T  0097:005912
               BEGIN EMITNO(ACCUM1); STEPIT END                                 08621000 T  0097:006011
```

```
                    ELSE IF ACCUM[1]="6UNLOC" THEN %%% [NS,UNLOCK                    08621002 T  0097:0063:2
                    BEGIN EMITL(1); EMITD(47,4,1); STEPIT END                        08621004 T  0097:0069:2
                    ELSE BEGIN EMITTIME; EMITL(ACCUM1) END%[DPN,AEXP                 08621010 T  0097:0072:0
               ELSE EMITNO(ACCUM1) %%% HAVE ONLY [DPN                                08621020 T  0097:0078:1
          ELSE IF ELCLASS=FACTOP THEN %%% HAVE [*                                    08622000 T  0097:0080:1
                    IF STEPI=COMMA THEN %%% HAVE [*,                                 08622010 T  0097:0081:3
                         IF STEPI=FACTOP THEN %%% HAVE [*,*                          08623000 T  0097:0083:2
                              BEGIN EMITNO(1); STEPIT END                            08624000 T  0097:0084:1
                    ELSE IF ACCUM[1]="6UNLOC" THEN %%% [*,UNLOCK                     08624002 T  0097:0087:2
                    BEGIN EMITL(1); EMITD(47,4,1); STEPIT END                        08624004 T  0097:0088:1
                         ELSE BEGIN EMITTIME; EMITL(1) END %[*,AEXP                  08625000 T  0097:0091:3
                    ELSE EMITNO(1) %%% HAVE ONLY [*                                  08626000 T  0097:0098:1
               ELSE BEGIN AEXP; EMITO(SSP); EMITPAIR(JUNK,ISN);                      08627000 T  0097:0100:1
                         %% HAVE [AEXP                                               08627100 T  0097:0103:2
                    IF ELCLASS=COMMA THEN %%% HAVE [AEXP,                            08628000 T  0097:0103:2
                         IF STEPI=FACTOP THEN STEPIT %%%HAVE [AEXP,*                 08629000 T  0097:0104:0
                    ELSE IF ACCUM[1]="6UNLOC" THEN %%% [AEXP,UNLOCK                  08629002 T  0097:0106:0
                    BEGIN EMITL(1); EMITD(47,4,1); STEPIT END                        08629004 T  0097:0108:0
                         ELSE BEGIN EMITTIME; EMITO(LOR)END;%[AEXP,A                 08630000 T  0097:0111:2
                    EMITL(0) ; %%% 0 IS NO DPN.                                      08631000 T  0097:0117:3
                    END ;                                                            08632000 T  0097:0118:0
               IF CHECK(RTBRKET,441)                                                 08633000 T  0097:0118:0
               THEN GO TO EXIT;                                                      08634000 T  0097:0118:1
                COMMENT ERROR 441 MEANS MISSING RIGHT BRACKET IN CARRIAGE            08635000 T  0097:0120:0
               CONTROL PART;                                                         08636000 T  0097:0120:0
CHKRTPAREN:    IF STEPI = RTPAREN                                                    08637000 T  0097:0120:0
               THEN GO TO ONEPARENSH;                                                08638000 T  0097:0120:1
               IF CHECK(COMMA,442)                                                   08639000 T  0097:0121:3
               THEN GO TO EXIT;                                                      08640000 T  0097:0122:0
                COMMENT ERROR 442 MEANS ILLEGAL CARRIAGE CONTROL                     08641000 T  0097:0123:2
               DELIMITER IN A WRITE STATEMENT;                                       08642000 T  0097:0123:2
     CHKSECOND:STEPIT;                                                               08643000 T  0097:0123:2
               IF RANGE(FRMTID,SUPERFRMTID)                                          08644000 T  0097:0124:1
               THEN BEGIN COMMENT THIS IS THE FORMAT FORM OF THE WRITE;               08645000 T  0097:0125:2
                         PASSFORMAT;                                                 08646000 T  0097:0126:0
WRITXFORM:               IF STEPI = RTPAREN                                          08647000 T  0097:0126:1
                         THEN BEGIN COMMENT THIS IS THE TWO PARAMETER                08648000 T  0097:0127:3
                                   CASE OF THE WRITE;                                08649000 T  0097:0128:1
                                   STEPIT;     EMITL(0);    GO EMITCALL;             08650000 T  0097:0128:1
                              END;                                                   08651000 T  0097:0130:0
                         GO PASSLIST;                                                08652000 T  0097:0130:0
                    END;                                                             08653000 T  0097:0130:1
          IF ELCLASS=LFTBRKET THEN %%% FREE FIELD AT LEAST = [AEXP]/.                08653100 T  0097:0130:1
               BEGIN I←I-1;  BANA;  EMITO(SSP);  EMITPAIR(1,ADD);                    08653110 T  0097:0131:3
               IF ELCLASS≠MULOP THEN ERR(443)                                        08653120 T  0097:0135:3
               ELSE IF STEPI=MULOP THEN BEGIN EMITO(SSN); STEPIT END ;               08653125 T  0097:0137:2
               IF ELCLASS=LFTBRKET THEN %%% FREE FIELD = [AEXP]/[AEXP].              08653130 T  0097:0140:1
                    BEGIN I←I-1;  BANA;  EMITO(SSP);  EMITPAIR(1,ADD) END            08653140 T  0097:0141:3
               ELSE EMITL(1) ; %%% FREE FIELD = [AEXP]/.                             08653150 T  0097:0145:2
               GO TO PASSLIST ;                                                      08653160 T  0097:0146:1
               END                                                                   08653170 T  0097:0147:2
          ELSE IF ELCLASS=MULOP THEN %%% FREE FIELD AT LEAST = /.                    08653180 T  0097:0147:2
               BEGIN EMITL(1) ;                                                      08653190 T  0097:0148:1
               IF STEPI=MULOP THEN BEGIN EMITO(SSN); STEPIT END ;                    08653195 T  0097:0149:3
               IF ELCLASS=LFTBRKET THEN %%% FREE FIELD = /[AEXP].                    08653200 T  0097:0152:1
                    BEGIN I←I-1;  BANA;  EMITO(SSP);  EMITPAIR(1,ADD) END            08653210 T  0097:0153:2
               ELSE EMITL(1) ; %%% FREE FIELD = /.                                   08653220 T  0097:0157:2
```

```
GO TO PASSLIST ;
END OF SCANNING FOR FREE FIELD FORMAT ;
IF ELCLASS = FACTOP
THEN BEGIN COMMENT THIS IS THE ASTERISK FORM OF THE WRITE;
          EMITL(0);       EMITL(0);       STEPIT;
          GO PASSLIST;
     END;
IF ACCUM[1]="1<0000" THEN
     BEGIN EXPLICITFORMAT; GO TO WRITXFORM; END;
IF ARC THEN
BEGIN KLUDGE(=HOLD);
     GO TO EXIT;
END ARRAY TO ARRAY CASE;
EMITL(0);           AEXP;
IF CHECK(COMMA,443)
THEN GO TO EXIT;
 COMMENT ERROR 443 MEANS IMPROPER DELIMITER FOR SECOND
PARAMETER IN WRITE STATEMENT;
STEPIT;
IF RANGE(BOOARRAYID,INTARRAYID)
THEN BEGIN COMMENT THIS IS THE ROW DESIGNATOR CASE;
          VARIABLE(FL);
          IF TABLE(I=2) ≠ FACTOP
          THEN BEGIN COMMENT ERROR 444 MEANS IMPROPER ROW
                    DESIGNATOR IN A WRITE STATEMENT;
                    ERROR(444); GO TO EXIT;
               END;
          IF CHECK(RTPAREN,445)
          THEN GO TO EXIT;
           COMMENT ERROR 445 MEANS MISSING RIGHT
          PARENTHESIS AFTER A ROW DESIGNATOR IN A WRITE
          STATEMENT;
          GOGOGO ← TRUE;%
          STEPIT;          GO EMITCALL;
     END
     ELSE BEGIN COMMENT ERROR 446 MEANS MISSING ROW DESIGNATOR;
          ERROR(446);      GO TO EXIT;
          END;
PASSLIST:IF CHECK(COMMA,447)
     THEN GO TO EXIT;
      COMMENT ERROR 447 MEANS IMPROPER DELIMITER PRECEEDING A
     LIST IN A WRITE STATEMENT;
  IF STEPI ≠ LISTID AND ELCLASS ≠ SUPERLISTID
    THEN BEGIN  RR1←LISTGEN; GO TO EMITCALL  END;
  CHECKER(ELBAT[I]);
  IF ELCLASS = SUPERLISTID  THEN
     BEGIN  COMMENT  SUBSCRIPTED SWITCH LIST ID;
          LISTADDRESS←ELBAT[I],ADDRESS;
          BANA;
          EMITV(LISTADDRESS);
          IF LISTADDRESS > 1023 THEN EMITO(PRTE);
          EMITO(LOD);
     I←I-1; COMMENT STEP DOWN THE&I FROM BANA;
     END ELSE
      BEGIN COMMENT A COMMON LIST ID;
          EMITPAIR(ELBAT[I].ADDRESS,LOD);
          END;
  STEPIT;
```

```
08653230 T  0097:015:1
08653240 T  0097:015:2
08654000 T  0097:015:2
08655000 T  0097:015:2
08656000 T  0097:016:0
08657000 T  0097:016:0
08658000 T  0097:016:1
08658010 T  0097:016:1
08658020 T  0097:016:3
08658100 T  0097:016:2
08658200 T  0097:016:2
08658300 T  0097:016:2
08658400 T  0097:016:3
08659000 T  0097:016:3
08660000 T  0097:017:1
08661000 T  0097:017:2
08662000 T  0097:017:1
08663000 T  0097:017:1
08664000 T  0097:017:1
08665000 T  0097:017:2
08666000 T  0097:017:3
08667000 T  0097:017:1
08668000 T  0097:017:2
08669000 T  0097:017:1
08670000 T  0097:017:3
08671000 T  0097:017:3
08672000 T  0097:017:1
08673000 T  0097:017:1
08674000 T  0097:017:2
08675000 T  0097:018:1
08676000 T  0097:018:1
08677000 T  0097:018:1
08677100 T  0097:018:1
08678000 T  0097:018:2
08679000 T  0097:018:0
08680000 T  0097:018:0
08681000 T  0097:018:1
08682000 T  0097:018:0
08683000 T  0097:018:0
08684000 T  0097:018:1
08685000 T  0097:018:3
08686000 T  0097:018:3
08687000 T  0097:018:3
08688000 T  0097:018:2
08688500 T  0097:018:3
08689000 T  0097:019:1
08690000 T  0097:019:3
08692000 T  0097:019:0
08693000 T  0097:019:3
08694000 T  0097:019:0
08694500 T  0097:019:1
08695000 T  0097:019:1
08695500 T  0097:019:3
08696000 T  0097:019:1
08696500 T  0097:019:1
08696520 T  0097:019:2
08696530 T  0097:020:2
08696540 T  0097:020:2
```

```
          IF CHECK(RTPAREN,448) THEN GO TO EXIT;                                  08696550 T  0097:020:3
           COMMENT 448 IS IMPROPER LIST DELMETER IN WRITE STATEMENT;              08696560 T  0097:020:2
          STEPIT;                                                                 08697000 T  0097:020:2
EMITCALL:     IF ELCLASS=LFTBRKET AND NOT ARC THEN                               08698000 T  0097:020:3
              BEGIN EMITO(MKS);                                                  08698100 T  0097:020:3
                   IF STEPI ≠ COLON THEN DEXP ELSE EMIT(0);                      08698200 T  0097:020:1
                   IF ELCLASS≠COLON THEN EMIT(0) ELSE                            08698300 T  0097:021:0
                   BEGIN STEPIT; DEXP END;                                       08698400 T  0097:021:0
                   IF CHECK(RTBRKET,433) THEN GO EXIT;                           08698500 T  0097:021:3
                   EMITL(15); EMITV(5); STEPIT;                                  08698600 T  0097:021:2
              END;%                                                              08698700 T  0097:021:2
              IF GOGOGO THEN%                                                    08698750 T  0097:021:2
              BEGIN EMIT(0);  EMIT(0);  EMIT(0);%                                08698800 T  0097:021:3
                   EMIT(0);  EMIT(0);  EMITV(12);%                               08698850 T  0097:022:0
              END ELSE EMITV(GNAT(INTERPTO));%                                   08698900 T  0097:022:1
              GOGOGO ← FALSE;%                                                   08698950 T  0097:022:0
         EXIT:;                                                                  08699000 T  0097:022:2
         END WRITESTMT;                                                          08700000 T  0097:022:2
                                                                                97 IS  228 LONG, NEXT SEG    3


PROCEDURE LOCKSTMT;                                                              08701000 T  0003:083:0
         BEGIN COMMENT THE LOCK STATEMENT ROUTINE GENERATES CODE THAT            08702000 T  0003:083:0
              CALLS ON THE FILE CONTROL ROUTINE PASSING TO IT THE                08703000 T  0003:083:0
              FOLLOWING PARAMETERS FOR THE CORRESPONDING CASES.                  08704000 T  0003:083:0
              ***********************************************************        08705000 T  0003:083:0
              <LOCK STATEMENT>::=LOCK(<FILE PART>,SAVE)/                         08706000 T  0003:083:0
              -  -  -  -  -  -  -  -  -  -  -  -  -  -  -                        08707000 T  0003:083:0
(2,0,FILE,4)                                                                     08708000 T  0003:083:0
              **  **  **  **  **  **  **  **  **  **  **  **  **  **              08709000 T  0003:083:0
LOCK(<FILE PART>,RELEASE)                                                        08710000 T  0003:083:0
              -  -  -  -  -  -  -  -  -  -  -  -  -  -  -                        08711000 T  0003:083:0
(6,0,FILE,4);                                                                    08712000 T  0003:083:0
LABEL EXIT;  COMMENT THE LABEL EXIT APPEARS AFTER THE LAST                       08713000 T  0003:083:0
                                                                        START OF SEGMENT **********   98
              EXECUTABLE STATEMENT IN THE LOCK ROUTINE;                          08714000 T  0098:000:0
         DEFINE THISL = RR1#;  COMMENT THISL IS A TEMP CELL                      08715000 T  0098:000:0
                               FOR THE CURRENT L REGISTER;                       08716000 T  0098:000:0
         DEFINE LTEMP = RR2#;  COMMENT LTEMP CONTAINS THE                        08717000 T  0098:000:0
                               L REGISTER SETTING FOR THE                        08718000 T  0098:000:0
                               SAVE OR RELEASE LITERAL THAT                      08719000 T  0098:000:0
                               GETS PASSED TO KEN MEYERS;                        08720000 T  0098:000:0
STEPIT;                                                                          08721000 T  0098:000:0
IF CHECK(LEFTPAREN,450)                                                          08722000 T  0098:000:1
THEN GO TO EXIT;                                                                 08723000 T  0098:000:2
 COMMENT ERROR NUMBER 450 MEANS MISSING LEFT PARENTHESIS                         08724000 T  0098:000:2
IN A LOCK STATEMENT;                                                             08725000 T  0098:000:2
STEPIT;                                                                          08726000 T  0098:000:2
       IF NOT RANGE(FILEID,SUPERFILEID)                                          08727000 T  0098:000:1
THEN BEGIN COMMENT MUST BE READ-ONLY ARRAY TYPE LOCK;                            08728000 T  0098:000:2
     IF NOT RANGE(BOOARRAYID,INTARRAYID) THEN                                    08728100 T  0098:000:1
     BEGIN ERR(451);  GO TO EXIT END;                                            08728200 T  0098:000:3
     VARIABLE(FL);  L ← L-1;                                                     08728300 T  0098:000:3
     IF TABLE(I-2)≠FACTOP THEN FLAG(208);                                        08728400 T  0098:000:3
     EMITO(DUP);  EMITO(LOD);  EMITL(24);                                        08728500 T  0098:001:1
```

```
          EMITD(43,3,5); EMITO(XCH); EMITO(STD);                                08728600 T  0098!0014!1
          IF ELCLASS=RTPAREN THEN STEPIT ELSE ERR(104);                         08729000 T  0098!0017!3
          GO TO EXIT                                                            08730000 T  0098!0020!1
          END;                                                                  08731000 T  0098!0021!2
            PASFILE;                                                            08732000 T  0098!0021!2
       IF ELCLASS=RTPAREN THEN ELBAT[(I+I-2)+1].CLASS+                          08732100 T  0098!0021!3
                                                  RELEASEV ELSE                 08732200 T  0098!0025!2
       IF CHECK(COMMA,452)                                                      08733000 T  0098!0026!1
       THEN GO TO EXIT;                                                         08734000 T  0098!0027!3
        COMMENT ERROR 452 MEANS MISSING COMMA IN A LOCK STATEMENT               08735000 T  0098!0029!2
       ;                                                                        08736000 T  0098!0029!2
             THISL+L;          L+LTEMP;                                         08737000 T  0098!0029!2
       IF(RRB1+STEPI = RELEASEV) OR ELCLASS = DECLARATORS AND                   08738000 T  0098!0030!1
            ELBAT[I].ADDRESS=SAVEV OR ELCLASS=FACTOP                            08739000 T  0098!0032!1
       THEN EMITL(IF RRB1                                                       08740000 T  0098!0035!2
                  THEN 6                                                        08741000 T  0098!0036!1
                  ELSE IF ELCLASS=FACTOP THEN 8 ELSE 2)                         08742000 T  0098!0036!1
       ELSE BEGIN COMMENT ERROR 453 MEANS IMPROPER UNIT                         08743000 T  0098!0040!0
                  DISPOSITION PART;                                             08744000 T  0098!0041!2
                  ERROR(453);     GO TO EXIT;                                   08745000 T  0098!0041!2
            END;                                                                08746000 T  0098!0042!0
              L+THISL;                                                          08747000 T  0098!0042!0
       STEPIT;                                                                  08748000 T  0098!0043!2
       IF CHECK(RTPAREN,454)                                                    08749000 T  0098!0043!3
       THEN GO TO EXIT;                                                         08750000 T  0098!0044!0
        COMMENT ERROR 454 MEANS MISSING RIGHT PARENTHESIS IN A                  08751000 T  0098!0045!2
       LOCK STATEMENT;                                                          08752000 T  0098!0045!2
       STEPIT;                                                                  08753000 T  0098!0045!2
EXIT:;                                                                          08754000 T  0098!0045!3
END LOCKSTMT;                                                                   08755000 T  0098!0046!0

                                                                  98 IS   47 LONG, NEXT SEG    3


PROCEDURE CLOSESTMT;                                                            08756000 T  0003!0830!0
        BEGIN COMMENT THE CLOSE STATEMENT ROUTINE GENERATES CODE THAT           08757000 T  0003!0830!0
             CALLS ON THE FILE CONTROL ROUTINE PASSING TO IT THE                08758000 T  0003!0830!0
             FOLLOWING PARAMETERS FOR THE CORRESPONDING CASES.                  08759000 T  0003!0830!0
             ************************************************************       08760000 T  0003!0830!0
             <CLOSE STATEMENT>::=CLOSE(<FILE PART>,SAVE)/                       08761000 T  0003!0830!0
                - - - - - - - - - - - - - - -                                   08762000 T  0003!0830!0
             (3,0,FILE,4)                                                       08763000 T  0003!0830!0
               ** ** ** ** ** ** ** ** ** ** ** ** ** ** **                     08764000 T  0003!0830!0
             CLOSE(<FILE PART>,RELEASE)/                                        08765000 T  0003!0830!0
                - - - - - - - - - - - - - - -                                   08766000 T  0003!0830!0
             (7,0,FILE,4)                                                       08767000 T  0003!0830!0
               ** ** ** ** ** ** ** ** ** ** ** ** ** ** **                     08768000 T  0003!0830!0
             CLOSE(<FILE PART>,*)                                               08769000 T  0003!0830!0
                - - - - - - - - - - - - - - -                                   08770000 T  0003!0830!0
                (1,0,FILE,4)                                                    08771000 T  0003!0830!0
             <CLOSE STATEMENT> ::= CLOSE(<FILE PART>, PURGE)                    08771100 T  0003!0830!0
                -- -- -- -- -- --- -- -- -- -- -- --                            08771200 T  0003!0830!0
             (4,0,FILE,4)                                                       08771300 T  0003!0830!0
               ** ** ** ** ** ** *** ** ** ** ** ** ;                           08771400 T  0003!0830!0
             LABEL EXIT;  COMMENT THE LABEL EXIT APPEARS AFTER THE LAST         08772000 T  0003!0830!0
                                                        START OF SEGMENT ********** 99
                  EXECUTABLE STATEMENT IN THE CLOSESTMT ROUTINE;                08773000 T  0099!0000!0
```

```
        DEFINE THISL = RR1#;  COMMENT THISL IS A TEMP CELL                    08774000 T  00991000:0
                              FOR THE CURRENT LREGISTER;                      08775000 T  00991000:0
        DEFINE LTEMP = RR2#;  COMMENT LTEMP CONTAINS THE                      08776000 T  00991000:0
                              L REGISTER SETTING FOR THE                      08777000 T  00991000:0
                              SAVE OR RELEASE LITERAL THAT                    08778000 T  00991000:0
                              GETS PASSED TO KEN MEYERS;                      08779000 T  00991000:0
LABEL EMITREST;  COMMENT I IS POINTING AT THE UNIT                            08780000 T  00991000:0
                 DISPOTION PART AND CODE FOR THE LAST THREE                   08781000 T  00991000:0
                 PARAMETERS TO THE FILE CONTROL ROUTINE                       08782000 T  00991000:0
                 MUST NOW BE GENERATED;                                       08783000 T  00991000:0
STEPIT;                                                                       08784000 T  00991000:0
IF CHECK(LEFTPAREN,455)                                                       08785000 T  00991000:1
THEN GO TO EXIT;                                                              08786000 T  00991000:2
 COMMENT ERROR 455 MEANS MISSING LEFT PARENTHESIS IN A                        08787000 T  00991000:0
CLOSE STATEMENT;                                                              08788000 T  00991000:0
STEPIT;                                                                       08789000 T  00991000:0
       IF NOT RANGE(FILEID,SUPERFILEID)                                       08790000 T  00991000:1
THEN BEGIN COMMENT ERROR 456 MEANS IMPROPER FILE PART IN A                    08791000 T  00991000:2
          CLOSE STATEMENT;                                                    08792000 T  00991000:1
          ERROR(456);     GO TO EXIT;                                         08793000 T  00991000:1
     END;                                                                     08794000 T  00991000:3
       PASFILE;                                                               08795000 T  00991000:3
IF ELCLASS=RTPAREN THEN ELBAT[(I+I-2)+1].CLASS+                               08795100 T  00991000:0
                                                RELEASEV ELSE                 08795200 T  00991001:0
IF CHECK(COMMA,457)                                                           08796000 T  00991001:3
THEN GO TO EXIT;                                                              08797000 T  00991001:1
 COMMENT ERROR 457 MEANS MISSING COMMA IN A CLOSE                             08798000 T  00991001:3
STATEMENT;                                                                    08799000 T  00991001:3
       THISL+L;          L+LTEMP;                                             08800000 T  00991001:3
IF STEPI = RELEASEV                                                           08801000 T  00991001:2
THEN BEGIN COMMENT RELEASE  UNIT DISPOSITION PART CASE;                       08802000 T  00991001:3
          EMITL(7);       GO EMITREST;                                        08803000 T  00991001:1
     END;                                                                     08804000 T  00991001:0
IF ELCLASS = FACTOP                                                           08805000 T  00991001:0
THEN BEGIN COMMENT ASTERISK UNIT DISPOSITION PART CASE;                       08806000 T  00991001:0
          EMITL(1);       GO EMITREST;                                        08807000 T  00991001:2
     END;                                                                     08808000 T  00991002:1
IF ELCLASS = DECLARATORS AND ELBAT[I].ADDRESS = SAVEV                         08809000 T  00991002:1
THEN BEGIN COMMENT SAVE UNIT DISPOSTION PART CASE;                            08810000 T  00991002:0
          EMITL(3);       GO EMITREST;                                        08811000 T  00991002:3
     END;                                                                     08812000 T  00991002:1
IF ACCUM[1] ="5PURGE" THEN BEGIN COMMENT FILE PURGE;                          08812100 T  00991002:1
                                  EMITL(4); GO EMITREST;                      08812200 T  00991002:0
                               END;                                           08812300 T  00991002:2
ERROR(458);         GO TO EXIT;                                               08813000 T  00991002:2
 COMMENT ERROR 458 MEANS IMPROPER UNIT DISPOSITION PART                       08814000 T  00991003:0
IN A CLOSE STATEMENT;                                                         08815000 T  00991003:0
EMITREST:STEPIT;                                                              08816000 T  00991003:0
          L+THISL;                                                            08817000 T  00991003:3
       IF CHECK(RTPAREN,459)                                                  08818000 T  00991003:2
THEN GO TO EXIT;                                                              08819000 T  00991003:1
 COMMENT ERROR 459 MEANS MISSING RIGHT PARENTHESIS IN A                       08820000 T  00991003:0
CLOSE STATEMENT;                                                              08821000 T  00991003:0
          STEPIT;                                                             08822000 T  00991003:0
   EXIT:;                                                                     08823000 T  00991003:1
   END CLOSESTMT;                                                             08824000 T  00991003:2
                                                                      99 IS    36 LONG, NEXT SEG    3
```

```
PROCEDURE RWNDSTMT;                                                    08825000 T  0003:083:0
        BEGIN COMMENT THE REWIND STATEMENT ROUTINE GENERATES CODE THAT 08826000 T  0003:083:0
             CALLS ON THE FILE CONTROL ROUTINE PASSING TO IT THE       08827000 T  0003:083:0
             FOLLOWING PARAMETERS.                                     08828000 T  0003:083:0
             ********************************************************* 08829000 T  0003:083:0
             <REWIND STATEMENT>::=REWIND(<FILE PART>)                  08830000 T  0003:083:0
                - - - - - - - - - - - - - - - - - - - -                08831000 T  0003:083:0
             (0,0,FILE,4);                                             08832000 T  0003:083:0
             LABEL EXIT;  COMMENT THE LABEL EXIT APPEARS AFTER THE LAST 08833000 T  0003:083:0
                                                          START OF SEGMENT ********** 100
                       EXECUTABLE STATEMENT IN THE REWIND ROUTINE;     08834000 T  0100:000:0
                  DEFINE THISL = RR1#;  COMMENT THISL IS A TEMP CELL   08835000 T  0100:000:0
                                        FOR THE CURRENT L REGISTER;    08836000 T  0100:000:0
                  DEFINE LTEMP = RR2#;  COMMENT LTEMP CONTAINS THE     08837000 T  0100:000:0
                                        L REGISTER SETTING FOR THE     08838000 T  0100:000:0
                                        SAVE OR RELEASE LITERAL THAT   08839000 T  0100:000:0
                                        GETS PASSED TO KEN MEYERS;     08840000 T  0100:000:0
             STEPIT;                                                   08841000 T  0100:000:0
             IF CHECK(LEFTPAREN,460)                                   08842000 T  0100:000:1
             THEN GO TO EXIT;                                          08843000 T  0100:000:2
              COMMENT ERROR 460 MEANS MISSING LEFT PARENTHESIS IN A    08844000 T  0100:000:0
             REWIND STATEMENT;                                         08845000 T  0100:000:0
             STEPIT;                                                   08846000 T  0100:000:0
                    IF NOT RANGE(FILEID,SUPERFILEID)                   08847000 T  0100:000:1
                    THEN BEGIN COMMENT ERROR 461 MEANS IMPROPER FILE PART IN A 08848000 T  0100:000:2
                           REWIND STATEMENT;                           08849000 T  0100:000:1
                           ERROR(461);     GO TO EXIT;                 08850000 T  0100:000:1
                         END;                                          08851000 T  0100:000:3
                           PASFILE;                                    08852000 T  0100:000:3
             IF CHECK(RTPAREN,462)                                     08853000 T  0100:000:0
             THEN GO TO EXIT;                                          08854000 T  0100:000:1
              COMMENT ERROR 462 MEANS MISSING RIGHT PARENTHESIS IN A   08855000 T  0100:000:0
             REWIND STATEMENT;                                         08856000 T  0100:000:0
                    STEPIT;          THISL←L;          L←LTEMP;       08857000 T  0100:000:0
                    EMITL(0);        L←THISL;                         08858000 T  0100:001:0
        EXIT;                                                          08859000 T  0100:001:1
        END RWNDSTMT;                                                  08860000 T  0100:001:0
                                                        100 IS   13 LONG, NEXT SEG    3


                PROCEDURE EXPLICITFORMAT;                              08860050 T  0003:083:0
                    BEGIN INTEGER PRT; ARRAY TEDOC[0:7,0:127];         08860100 T  0003:083:0
                                                          START OF SEGMENT ********** 101
STACK(F+2) ← PRT
STACK(F+3) ← TEDOC
                    MOVECODE(TEDOC,EDOC);                              08860150 T  0101:000:0
                    GT5:=SGNO; GT1:=(2×SGAVL-1)&2[4:46:2]; SGNO:=SGAVL;08860200 T  0101:000:0
                    F := 0;  PRT := GETSPACE(TRUE,-4); % FORMAT DESCR. 08860250 T  0101:000:1
                    PRT := PROGDESCBLDR(LDES,0,PRT);                   08860300 T  0101:001:2
                    FLCLASS := "<";  TB1 := FORMATPHRASE;              08860350 T  0101:001:1
                    SEGMENT(-F,SGNO,GT5);  SGAVL := SGAVL+1;           08860400 T  0101:001:1
```

```
    SGNO := GT5;  MOVECODE(TEDOC,EDOC);                                    08860450 T  0101:0017:2
    IF LASTELCLASS ≠ ">" THEN ERR(136);                                    08860500 T  0101:0020:0
    IF ELCLASS = "," THEN ELBAT[I].CLASS := COMMA ELSE                     08860600 T  0101:0022:0
    IF ELCLASS = ")" THEN ELBAT[I].CLASS := RTPAREN ELSE                   08860650 T  0101:0026:0
    ELBAT[I].CLASS := 0;  I:=I-1;                                          08860700 T  0101:0030:0
    EMITL(0);  EMITPAIR(PRT,LOD);                                          08860750 T  0101:0034:1
END EXPLICITFORMAT;                                                        08860800 T  0101:0036:0
                                                                    101 IS   40 LONG, NEXT SEG    3


COMMENT   SORTSTMT AND MERGESTMT ANALYZE THEIR APPROPRIATE SYNTAXES        08861000 T  0003:0830:0
          AND CALL SORTI, PASSING THE FOLLOWING:                           08862000 T  0003:0830:0
 SORT:          MERGE:                                                     08863000 T  0003:0830:0
<AEXP>          0                   DISK SIZE,IF SPECIFIED                 08864000 T  0003:0830:0
<AEXP>          0                   CORE SIZE,IF SPECIFIED                 08865000 T  0003:0830:0
 0              0                   ALFA FLAG                              08866000 T  0003:0830:0
<AEXP>         <AEXP>               RECORD SIZE                            08867000 T  0003:0830:0
 PROG.DESC.     PROG.DESC.          DESCRIPTOR TO COMPARE PROCEDURE         08868000 T  0003:0830:0
 PROG.DESC.     PROG.DESC.          DESCRIPTOR TO HIVALUE PROCEDURE         08869000 T  0003:0830:0
 ...            2,3,4,5,6,7         NUMBER OF FILES TO MERGE, OR           08870000 T  0003:0830:0
 0,3,4,5        ...                 NUMBER OF SORTTAPES TO USE             08871000 T  0003:0830:0
 TP5            FL7                 SCRATCH TAPES FOR SORT,                08872000 T  0003:0830:0
 TP4            FL6                  OR MERGE FILES, POINTERS TO           08873000 T  0003:0830:0
 TP3            FL5                  TOP I/O DESCRIPTORS, OR ZERO          08874000 T  0003:0830:0
 TP2            FL4                  IF NOT USED.                          08875000 T  0003:0830:0
 TP1            FL3                                                        08876000 T  0003:0830:0
 0              FL2                 DISK FILES FOR SORT                    08877000 T  0003:0830:0
                DKO                 FL1                                    08878000 T  0003:0830:0
                0/1                 0                                      08879000 T  0003:0830:0
                0/1                 0/1     TRUE IF INPUT PROCEDURE        08880000 T  0003:0830:0
                INF                 0       TRUE IF OUTPUT PROCEDURE       08881000 T  0003:0830:0
                OUTF                OUTF    POINTER TO I/O DESC FOR INPUT  08882000 T  0003:0830:0
                                             OR OUTPUT FILE, OR MOTHER      08883000 T  0003:0830:0
                PD/0                0       OF WORK ARRAY.                 08884000 T  0003:0830:0
                PD/0                PD/0    INPUT PROCEDURE DESCRIPTOR     08885000 T  0003:0830:0
                0                   0       OUTPUT PROCEDURE               08886000 T  0003:0830:0
                0                   0                                      08887000 T  0003:0830:0
                0                   0                                      08888000 T  0003:0830:0
                LIT                 LIT     PRT INDEX OF MERGE INTRINSIC   08889000 T  0003:0830:0
                0                   0                                      08890000 T  0003:0830:0
                0                   1       SORT/MERGE FLAG                08891000 T  0003:0830:0
                ...                 MSCW                                   08892000 T  0003:0830:0
                0                           SORT-FILE MOTHER               08893000 T  0003:0830:0
                0                                 DESCRIPTORS              08894000 T  0003:0830:0
                0                                    .                     08895000 T  0003:0830:0
                0                                    .                     08896000 T  0003:0830:0
                0                                    .                     08897000 T  0003:0830:0
                0                                    .                     08898000 T  0003:0830:0
                MSCW;                                                      08900000 T  0003:0830:0
            PROCEDURE MERGESTMT;                                           08901000 T  0003:0830:0
               BEGIN INTEGER J,K,FILER,FILEND;                             08902000 T  0003:0830:0
                                                                  START OF SEGMENT **********  102

    STACK(F+2) := J
    STACK(F+3) := K
    STACK(F+4) := FILER
```

```
STACK(F+5) = FILEND
                    BOOLEAN OPTOG;                                                  08903000 T  0102:000:0
STACK(F+6) = OPTOG
                    LABEL QUIT;                                                     08904000 T  0102:000:0
                    STEPIT; IF CHECK(LEFTPAREN,367) THEN GO QUIT;                   08905000 T  0102:000:0
                    EMITO(MKS); EMITL(1); EMIT(0); EMITL(GNAT(MERGEI));             08906000 T  0102:000:0
                    EMIT(0); EMIT(0); EMIT(0);                                      08907000 T  0102:000:3
                    IF OPTOG+(STEPI=FILEID OR ELCLASS=SUPERFILEID) THEN EMIT(0)     08908000 T  0102:000:0
                    ELSE IF NOT OUTPROCHECK(ELBAT[I]) THEN GO QUIT ELSE             08909000 T  0102:001:2
                         EMITPAIR(ELBAT[I].ADDRESS,LOD);                            08910000 T  0102:001:3
                    EMIT(0);IF OPTOG THEN BEGIN PASSFILE; I+I-1 END ELSE            08911000 T  0102:001:3
                         EMITN(GNAT(SORTA));                                        08911100 T  0102:001:1
                    IF NOT COMMACHECK THEN GO QUIT;                                 08912000 T  0102:002:1
                    EMIT(0); EMITL(REAL(TRUE AND NOT OPTOG)); EMIT(0);              08913000 T  0102:002:3
                    FILER+BUMPL; IF NOT HVCHECK(ELBAT[I]) THEN GO QUIT;             08914000 T  0102:002:1
                    EMITPAIR(ELBAT[I].ADDRESS,LOD); IF NOT COMMACHECK THEN GO QUIT; 08915000 T  0102:002:3
                    IF NOT EQLESCHECK(ELBAT[I]) THEN GO QUIT;                       08916000 T  0102:003:0
                    EMITPAIR(ELBAT[I].ADDRESS,LOD); IF NOT COMMACHECK THEN GO QUIT; 08917000 T  0102:003:0
                    AEXP; EMITB(BFW,FILER,FILEND+BUMPL);                            08918000 T  0102:003:1
                    FOR J+1 STEP 1 WHILE ELCLASS=COMMA DO                           08919000 T  0102:003:0
                         BEGIN STEPIT; PASSFILE END;                                08920000 T  0102:004:0
                    FOR K+J STEP 1 UNTIL 7 DO EMIT(0);  J+J-1;                      08921000 T  0102:004:3
                    IF J>7 OR J<2 THEN BEGIN ERR(368); GO QUIT END;                 08922000 T  0102:004:2
                    EMITL(J); EMITB(BBW,BUMPL,FILER); EMITB(BFW,FILEND,L);          08923000 T  0102:005:1
                    IF CHECK(RTPAREN,369) THEN GO QUIT; STEPIT; EMITO(SSN);         08924000 T  0102:005:2
                    EMIT(0); EMIT(0); EMIT(0);                                      08925000 T  0102:006:0
               QUIT;  EMITV(GNAT(SORTI));                                           08926000 T  0102:006:0
                  END MERGESTMT;                                                    08927000 T  0102:006:0
```

102 IS 68 LONG, NEXT-SEG 3

```
                  PROCEDURE SORTSTMT;                                               08928000 T  0003:083:0
                     BEGIN BOOLEAN INPRO,OUTPRO;                                    08929000 T  0003:083:0
                                                                                 START OF SEGMENT **********  103
STACK(F+2) = INPRO
STACK(F+3) = OUTPRO
                      INTEGER A,J;                                                  08930000 T  0103:000:0
STACK(F+4) = A
STACK(F+5) = J
                      LABEL QUIT;   DEFINE RDS=1,280#;                          08931000 T  0103:000:0
                     STREAM PROCEDURE STUFFILE(IDLOC,FN,     SFN);                  08932000 T  0103:000:0
PRT(1037) = STUFFILE
                                      VALUE      FN,     SFN ;                     08933000 T  0103:000:0
                     BEGIN DI+IDLOC; DI+DI+5; DI+DC;                            08934000 T  0103:000:0
                          SI+LOC FN; SI+SI+5; DS+3 CHR; SI+SI+7;                08935000 T  0103:000:1
                          DS+11 LIT"0000000DSRT"; DS+CHR; SI+SI-1;              08936000 T  0103:000:3
                          DS+7 LIT"  5DSRT"; DS+CHR; SFN+DI; SI+LOC SFN;        08937000 T  0103:000:0
                          DI+IDLOC; DI+DI+5; SI+SI+5; DS+3 CHR;                 08938000 T  0103:000:0
                     END STUFFILE;                                              08939000 T  0103:000:2

                     BOOLEAN PROCEDURE INPROCHECK(ELBW); VALUE ELBW; REAL ELBW;     08940000 T  0103:000:2
PRT(1040) = INPROCHECK
```

```
                IF ELBW.CLASS≠BOOPROCID THEN ERR(363) ELSE                                  08941000 T  0103:0007:2
                IF BOOLEAN(ELBW.FORMAL) THEN INPROCHECK←TRUE  ELSE                          08941100 T  0103:0010:1
                IF TAKE(GT1←GIT(ELBW))≠1 THEN ERR(364) ELSE                                 08942000 T  0103:0013:2
                IF ARRAYCHECK(TAKE(GT1←1)) THEN ERR(365) ELSE                               08943000 T  0103:0017:2
                     INPROCHECK←TRUE;                                                       08944000 T  0103:0020:1



           IF SFILENO=0 THEN                                                                08945000 T  0103:0024:0
           BEGIN SFILENO←FILENO;                                                            08946000 T  0103:0025:3
                FOR J←1 STEP 1 UNTIL 7 DO                                                   08947000 T  0103:0027:2
                IF MKABS(IDARRAY[127])<IDLOC.[33:15]+3 THEN FLAG(40) ELSE                   08948000 T  0103:0028:0
                BEGIN STUFFILE(IDLOC.(IF J≤2 THEN 12 ELSE 2)&FILENO                         08949000 T  0103:0032:0
                                   [30:36:12],J);                                           08950000 T  0103:0035:3
                     FILENO←FILENO+1;                                                       08951000 T  0103:0037:2
           END; END;                                                                        08952000 T  0103:0038:1
           EMITO(MKS); EMITV(BLOCKCTR); EMITPAIR(1,ADD);                                    08953000 T  0103:0040:1
           EMITPAIR(BLOCKCTR,STD); EMIT(0); EMIT(0);                                        08954000 T  0103:0043:2
           EMITN(2); EMITPAIR(RDS); EMITPAIR(A←GNAT(SORTA),STD);                            08955000 T  0103:0045:3
           EMITO(MKS); EMITL(20); EMITL(1000); EMITL(3); EMITL(SFILENO);                    08956000 T  0103:0049:3
           EMIT(0); EMITO(LNG); EMITN(A); EMITO(INX); EMITL(2);                             08957000 T  0103:0053:2
           EMITL(1); EMITL(10); EMIT(0); EMIT(0); EMITL(10);                                08958000 T  0103:0057:2
           EMITL(8); EMITV(5);                                                              08959000 T  0103:0060:1
           EMIT(0); EMIT(0); EMIT(0); EMIT(0); EMIT(0); EMIT(0);                            08964000 T  0103:0062:0
           EMITL(GNAT(MERGEI)); EMIT(0); EMIT(0); EMIT(0); STEPIT;                          08965000 T  0103:0066:1
           IF CHECK(LEFTPAREN,355) THEN GO QUIT;                                            08966000 T  0103:0070:1
%     OUTPUT OPTION.                                                                         08966500 T  0103:0072:1
           IF STEPI=FILEID OR ELCLASS=SUPERFILEID THEN                                      08967000 T  0103:0072:1
                BEGIN EMIT(0); PASSFILE; I←I+1 END ELSE                                     08968000 T  0103:0074:1
                BEGIN IF NOT(OUTPRO←OUTPROCHECK(ELBAT[I]))THEN GO QUIT;                     08969000 T  0103:0077:3
                     EMITPAIR(ELBAT[I],ADDRESS,LOD); EMITL(A);EMITN(10)                     08970000 T  0103:0080:0
                END;                                                                        08971000 T  0103:0082:1
           IF NOT COMMACHECK THEN GO QUIT;                                                  08972000 T  0103:0083:2
%     INPUT OPTION.                                                                          08972500 T  0103:0084:0
           IF ELCLASS=FILEID OR ELCLASS=SUPERFILEID THEN                                    08973000 T  0103:0084:0
                BEGIN EMITPAIR(0,XCH); PASSFILE; I←I+1 END ELSE                             08974000 T  0103:0086:0
                IF NOT(INPRO←INPROCHECK(ELBAT[I])) THEN GO QUIT ELSE                        08975000 T  0103:0089:2
                BEGIN EMITPAIR(ELBAT[I],ADDRESS,LOD); EMITO(XCH);                           08976000 T  0103:0091:3
                     IF OUTPRO THEN EMITO(DUP) ELSE BEGIN EMITL(A);                         08977000 T  0103:0094:1
                          EMITN(10) END; END INPUT PRO;                                     08978000 T  0103:0097:2
                EMITL(REAL(OUTPRO)); EMITL(REAL(INPRO)); EMIT(0);                           08979000 T  0103:0098:0
                EMITO(LNG); EMITN(A); EMITO(INX); EMITO(LOD);                               08980000 T  0103:0100:0
                EMITPAIR(5,CDC); EMIT(0);                                                   08981000 T  0103:0103:2
                IF NOT COMMACHECK THEN GO QUIT;                                             08983000 T  0103:0105:2
%     NUMBER OF TAPES.                                                                       08983500 T  0103:0106:0
                EMIT(0); EMIT(0); EMIT(0); EMIT(0); EMIT(0);                                08984000 T  0103:0106:0
                EMITO(MKS); EMITN(A); EMITL(SFILENO+2); AEXP; I←I+1;                        08985000 T  0103:0109:3
                EMITL(14); EMITV(5);                                                        08986000 T  0103:0114:0
                IF NOT COMMACHECK THEN GO QUIT;                                             08987000 T  0103:0115:3
%     HIVALUE PROCEDURE.                                                                     08987500 T  0103:0116:1
                IF NOT HVCHECK(ELBAT[I]) THEN GO QUIT;                                      08988000 T  0103:0116:1
                EMITPAIR(ELBAT[I].ADDRESS,LOD);                                             08989000 T  0103:0118:0
                IF NOT COMMACHECK THEN GO QUIT;                                             08990000 T  0103:0120:0
%     COMPARE PROCEDURE.                                                                     08990500 T  0103:0121:2
                IF NOT EQLESCHECK(ELBAT[I]) THEN GO QUIT;                                   08991000 T  0103:0121:2
                EMITPAIR(ELBAT[I].ADDRESS,LOD);                                             08992000 T  0103:0122:1
```

```
%   RECORD LENGTH.
              IF NOT COMMACHECK THEN GO QUIT; AEXP; EMITO(SSN);                  08992500 T  0103:012410
              EMIT(0); EMITPARR(A,SND);                                          08993000 T  0103:012410
%   CORE SIZE.                                                                   08993500 T  0103:012611
              IF ELCLASS=COMMA THEN                                              08993900 T  0103:012810
                   BEGIN                                                         08994000 T  0103:012810
                   STEPIT;                                                       08994010 T  0103:012912
                   CORESZ:=                                                      08994020 T  0103:012913
                    MAX(IF ELCLASS=NONLITNO THEN C ELSE 12000,CORESZ);           08994030 T  0103:013010
                   AEXP;                                                         08994040 T  0103:013010
                   END                                                           08994060 T  0103:013512
              ELSE IF ELCLASS=RTPAREN THEN                                       08994070 T  0103:013513
                        BEGIN                                                    08994080 T  0103:013513
                        IF CORESZ<1023 THEN CORESZ:=12000;                       08994090 T  0103:013713
                        EMIT(0);                                                 08994500 T  0103:013810
                        END                                                      08994510 T  0103:014010
                   ELSE ERR(366);                                                08994520 T  0103:014112
%   DISK SIZE.                                                                   08994530 T  0103:014112
              IF ELCLASS=COMMA THEN BEGIN STEPIT; AEXP END ELSE                  08994900 T  0103:014313
                   IF ELCLASS=RTPAREN THEN EMIT(0) ELSE ERR(366);                08995000 T  0103:014313
              IF ELCLASS#RTPAREN THEN ERR(366) ELSE STEPIT;                      08995500 T  0103:014610
 QUIT:        EMITV(GNAT(SORT1));                                                08996000 T  0103:014913
    END SORTSTMT;                                                                08997000 T  0103:015211
                                                                                 08998000 T  0103:015410
                                                                    103 IS  157 LONG, NEXT SEG    3


 COMMENT THE PROGRAM ROUTINE DOES THE INITIALIZATION AND THE WRAPUP              09000000 T  0003:083010
         FOR THE REST OF THE COMPILER.  THE MAIN PROGRAM OF THE COMPILER         09001000 T  0003:083010
                    IS SIMPLY A CALL ON THE PROGRAM ROUTINE;                     09002000 T  0003:083010
                PROCEDURE PROGRAM;                                               09003000 T  0003:083010
PRT(1041) = PROGRAM
                    BEGIN                                                        09004000 T  0003:083010
                STREAM PROCEDURE MDESC(WD,TOLOC);VALUE WD;                       09005000 T  0003:083010
                                                                    START OF SEGMENT **********  104
PRT(1042) = MDESC
                  BEGIN DI←LOC WD; DS←  SET;SI← LOC WD; DI←TOLOC;DS←WDS END;      09006000 T  0104:000010



               COMMENT THE FOLLOWING PROCEDURE PRINTS OUT THE PRT, NAME, AND     09007000 T  0104:000113
                       SEGMENT NUMBER OF THE INTRINSIC PROCEDURES USED IN THE    09008000 T  0104:000113
                       OBJECT PROGRAM;                                           09009000 T  0104:000113
               STREAM PROCEDURE WRTINTRSC(SGNO,ALFA,PRT,FIL);                    09010000 T  0104:000113
PRT(1043) = WRTINTRSC
                                    VALUE SGNO,PRT;                              09011000 T  0104:000113
                  BEGIN  LOCAL COUNT,DEST;                                       09012000 T  0104:000113
                  DI:=FIL; DS:=4 LIT"PRT("; SI:=LOC PRT; SI:=SI+4; TALLY:=4;     09013000 T  0104:000210
                  3(IF SC="0" THEN % DONT PRINT LEADING ZEROES.                  09014000 T  0104:000313
                       BEGIN SI:=SI+1; TALLY:=TALLY+63  END ELSE JUMP OUT);      09015000 T  0104:000411
                  COUNT:=TALLY; DS:=COUNT CHR; DS:=4 LIT") = ";                  09016000 T  0104:000712
                  SI:=ALFA; SI:=SI+2; DEST:=DI; % SAVE DI.                       09017000 T  0104:000811
                  DI:=LOC COUNT; DS:=7 LIT"0"; DS:=CHR; % NO OF CHARS IN NAME.   09018000 T  0104:000912
                  DI:=DEST; DS:=COUNT CHR; % INT. NAME.                          09019000 T  0104:001112
                  DS:=29 LIT" INTRINSIC, SEGMENT NUMBER = ";                     09020000 T  0104:001113
                  SI:= LOC SGNO; DS:=4 DEC; DS:=LIT",";                          09021000 T  0104:001513
```

```
                DI:=DI-5; DS:=4 FILL; % JUNK LEADING BLANKS.                                    09022000 T  01041001611
                END WRTINTRSC;                                                                  09023000 T  01041001712



                        DEFINE STARTINTRSC=426#;                                                09024000 T  01041001713
                        LABEL L1;                                                               09025000 T  01041001713
                        IDLOC←MKABS(IDARRAY[0]);                                                09026000 T  01041001713
                        IDLOCTEMP ← IDLOC;                                                      09027000 T  01041002010
        FILL OPTIONS[*] WITH "5CHECK",0,          % 0, 1                                         09027002 T  01041002011
                                                                                        START OF SEGMENT ********** 105
                             "6DEBUG",0,          % 2, 3                                         09027004 T  01041002211
                             "4DECK0",0,          % 4, 5                                         09027006 T  01041002211
                             "6FORMA",0,          % 6, 7                                         09027008 T  01041002211
                             "9INTRI",0,          % 8, 9                                         09027010 T  01041002211
                             "5LISTA",0,          % 10, 11                                       09027012 T  01041002211
                             "4LISTO",0,          % 12, 13                                       09027014 T  01041002211
                             "5LISTP",0,          % 14, 15                                       09027016 T  01041002211
                             "3MCP00",0,          % 16, 17                                       09027018 T  01041002211
                             "4TAPE0",0,          % 18, 19                                       09027020 T  01041002211
                             "4NEST0",0,          % 20, 21                                       09027022 T  01041002211
                             "3NEW00",0,          % 22, 23                                       09027024 T  01041002211
                             "7NEWIN",0,          % 24, 25                                       09027026 T  01041002211
                             "40MIT0",0,          % 26, 27                                       09027028 T  01041002211
                             "1$0000",0,          % 28, 29                                       09027030 T  01041002211
                             "3PRT00",0,          % 30, 31                                       09027032 T  01041002211
                             "5PUNCH",0,          % 32, 33                                       09027034 T  01041002211
                             "5PURGE",0,          % 34, 35                                       09027036 T  01041002211
                             "4SEGS0",0,          % 36, 37                                       09027038 T  01041002211
                             "3SEQ00",0,          % 38, 39                                       09027040 T  01041002211
                             "6SEQER",0,          % 40, 41                                       09027042 T  01041002211
                             "6SINGL",0,          % 42, 43                                       09027044 T  01041002211
                             "5STUFF",0,          % 44, 45                                       09027046 T  01041002211
                             "4VOID0",0,          % 46, 47                                       09027048 T  01041002211
                             "5VOIDT",0,          % 48, 49                                       09027050 T  01041002211
                             "4BEND0",0,          % 50, 51                                       09027052 T  01041002211
                             "4XREF0",0,          % 52, 53                                       09027054 T  01041002211
                             "7INCLU",0,          % 54,55                          %107-         09027056 C  01041002211
                            "8CODEF",0,           % 56,57                          %106-         09027058 C  01041002211
        0;                                                                                      09027100 T  01041002211
                                                                                      105 IS   59 LONG, NEXT SEG  104
                        LISTOG:=LISTER:=BOOLEAN(1-ERRORCOUNT,[46:1]);                           09028000 T  01041002211
                    OPTIONS[13] := REAL(LISTER);                                                09028005 T  01041002513
        COMMENT     LISTOG IS NOT SET BY DEFAULT ON TIMESHARING;                                09028010 T  01041002712
                        NOHEADING := TRUE;                                                      09028050 T  01041002712
                        BUILDLINE,[47:1]←SEQXEQTOG←REMOTOG←                                     09028100 T  01041002713
                                        BOOLEAN(ERRORCOUNT,[47:1]);                             09028150 T  01041002810
                        ERRORCOUNT := 0;                                                        09028900 T  01041003112
                        ERRMAX:=999; % MAY BE CHANGED IN DOLLARCARD.                            09028910 T  01041003113
                        BASENUM:=10000; ADDVALUE:=1000; NEWBASE:=TRUE;                          09028920 T  01041003211
        COMMENT     DEFAULT VALUES FOR "$SEQ" OPTION;                                           09028930 T  01041003411
                        LASTUSED := 4; % FOR INITIALIZATION.                                    09029000 T  01041003411
                        SGNO←1;SGAVL←2;PDNX←0;                                                  09030000 T  01041003513
                        FILENO:=DA:=1;                                                          09031000 T  01041003713
                        FILETHING ← 4095;                                                       09031100 T  01041003912
                        MAXSTACK ← 513;                                                         09032000 T  01041003913
```

```
                NEXTINFO ← LASTINFO ← LASTSEQROW×256+LASTSEQUENCE+1;                    09033000 T  01041004011
                PUTNBUMP(0);                                                           09034000 T  01041004312
                BLANKET(0,INFO[LASTSEQROW,LASTSEQUENCE]); % FOR "$ CHECK".             09034500 T  01041004410
                READACARD; % INITIALIZATION OF NCR,FCR, AND LCR, AND                   09035000 T  01041004610
                           % READS FIRST CARD INTO CARD BUFFER.                        09036000 T  01041004611
                LASTUSED := 1; % ASSUMES CARD ONLY UNTIL TOLD DIFFERENTLY.             09037000 T  01041004611
                NXTELBT←1; FAULTLEVEL←32;                                              09038000 T  01041004712
                PRTI ← PRTIMAX ← 18;                                                   09039000 T  01041004811
                MRCLEAN ← TRUE;                                                        09040000 T  01041005010
COMMENT  START FILLING TABLES NEEDED TO COMPILE A PROGRAM;                             09040100 T  01041005011
                FILL TEN[*] WITH                                                       09041000 T  01041005011
                                                                                       09042000 T  01041005112
                                                                                       09043000 T  01041005112
                                                                                       09044000 T  01041005112
                                                                                       09045000 T  01041005112
                                                                                       09046000 T  01041005112
                                                                                       09047000 T  01041005112
                                                                                       09048000 T  01041005112
                                                                                       09049000 T  01041005112
                                                                                       09050000 T  01041005112
                                                                                       09051000 T  01041005112
                                                                                       09052000 T  01041005112
                                                                                       09053000 T  01041005112
                                                                                       09054000 T  01041005112
                                                                                       09055000 T  01041005112
                                                                                       09056000 T  01041005112
                0CT1141000000000000,    0CT1131200000000000,                           09057000 T  01041005112
                                                                                  START OF SEGMENT **********  106
   0CT1121440000000000,    0CT1111750000000000,    0CT1102342000000000,                09058000 T  01041005211
   0CT1073032400000000,    0CT1063641100000000,    0CT1054611320000000,                09059000 T  01041005211
   0CT1045753604000000,    0CT1037346545000000,    0CT1011124027620000,                09060000 T  01041005211
    0CT0001351035564000,    0CT0011643245121000,    0CT0022214116345200,               09061000 T  01041005211
    0CT0032657142036440,    0CT0043432772446150,    0CT0054341571157602,               09062000 T  01041005211
    0CT0065432127413543,    0CT0076740555316473,    0CT0111053071060221,               09063000 T  01041005211
   0CT0121265707274266,    0CT0131543271153343,    0CT0142074147406234,                09064000 T  01041005211
   0CT0152513201307703,    0CT0163236041571663,    0CT0174105452130240,                09065000 T  01041005211
   0CT0205126764556310,    0CT0216354561711772,    0CT0231004771627437,                09066000 T  01041005211
   0CT0241206170175347,    0CT0251447626234641,    0CT0261761573704011,                09067000 T  01041005211
   0CT0272356132665013,    0CT0303051561442216,    0CT0313664115752661,                09068000 T  01041005211
   0CT0324641141345435,    0CT0336011371636745,    0CT0347413670206536,                09069000 T  01041005211
   0CT0361131664625027,    0CT0371360241772234,    0CT0401654312370703,                09070000 T  01041005211
   0CT0412227375067064,    0CT0422675274304701,    0CT0433454553366062,                09071000 T  01041005211
   0CT0444367706263476,    0CT0455465667740415,    0CT0467003245730521,                09072000 T  01041005211
   0CT0501060411731665,    0CT0511274514320242,    0CT0521553637404312,                09073000 T  01041005211
   0CT0532106607305375,    0CT0542530351166674,    0CT0553256443424453,                09074000 T  01041005211
   0CT0564132154331566,    0CT0575160607420123,    0CT0606414751324150,                09075000 T  01041005211
   0CT0621012014361120,    0CT0631214417455344,    0CT0641457523370635,                09076000 T  01041005211
   0CT0651773450267005,    0CT0662372362344606,    0CT0673071057035747,                09077000 T  01041005211
   0CT0703707272645341,    0CT0714671151416632,    0CT0726047403722400,                09078000 T  01041005211
   0CT0737461304707100,    0CT0751137556607072,    0CT0761367512350710,                09079000 T  01041005211
   0CT0771665435043073;                                                                09080000 T  01041005211
                                                                                  106 IS   69 LONG, NEXT SEG  104
COMMENT  THIS IS THE FILL FOR THE SECOND ROW OF INFO;                                  09081000 T  01041005211
                THE FIRST ITEMS ARE STREAM RESERVED WORDS,                             09082000 T  01041005211
                THEN ORDINARY RESERVED WORDS,                                          09083000 T  01041005211
                THEN INTRINSIC FUNCTIONS;                                              09084000 T  01041005211
 FILL INFO[1,*] WITH                                                                   09085000 T  01041005211
```

```
0CT0670000600000400, "2SI000",

0CT0700001040000402, "2DI000",
0CT0710001460000404, "2CI000",
0CT0720001630000406, "5TALLY",
0CT0730000530000410, "2DS000",
0CT0740000150000412, "4SKIP0",
0CT0750001620000414, "4JUMP0",
0CT0760000740000416, "2DB000",
0CT0770000500000420, "2SB000",
0CT1010000730000422, "2SC000",
0CT1020001160000424, "3LOC00",
0CT1030001170000426, "2DC000",
0CT1040001430000430, "5LOCAL",
0CT1050000340000432, "3LIT00",
0CT1060001036400434, "3SET00",
0CT1060001066500436, "5RESET",
0CT1060001020500440, "3WDS00",
0CT1060001357700442, "3CHR00",
0CT1060001057300444, "3ADD00",
0CT1060001617200446, "3SUB00",
0CT1060000727600450, "3ZON00",
0CT1060000417500452, "3NUM00",
0CT1060000766700454, "3OCT00",
0CT1060000176600456, "3DEC00",
0CT1004000260000460, "6TOGGL", "E0000000",
0CT0430000050000000, "5ALPHA",
0CT1330001030000000, "3AND00",
0CT0430000170000525, "5ARRAY",
0CT0660000000000000, "5BEGIN",
0CT0430000030000503, "7BOOLF", "AN000000",
0CT0470000000000000, "5CLOSF",
0CT1070000000000655, "7COMMF", "NT000000",
0CT0430000230000000, "6DEFIN", "E0000000",
0CT1360006000000000, "3DIV00",
0CT0550000000000000, "2D0000",
0CT0520000000000000, "6DOUBL", "E0000000",
0CT0430000100000000, "4DUMP0",
0CT0570000000000000, "4ELSE0",
0CT0600000000000000, "3END00",
0CT1300002030000000, "3EQV00",
0CT0360000000000644, "5FALSE",
0CT0430000210000000, "4FILE0",
0CT0610000001200000, "4FILL0",
0CT0530000000000000, "3FOR00",
0CT0430000200000554, "6FORMA", "T0000000",
0CT1100000000000000, "7FORWA", "RD000000",
0CT0640000000000604, "2G0000",
0CT0630000000000000, "2IF000",
0CT0430000130000000, "2IN000",
0CT0430000060000000, "7INTEG", "ER000000",
0CT1310000000000000, "3IMP00",
0CT0430000070000000, "5LABEL",
0CT0430000110000613, "4LIST0",
0CT0500000000000000, "4LOCK0",
0CT1360016000000000, "3MOD00",
0CT0430000140000000, "7MONIT", "OR000000",
```

```
COMMENT 256;          09086000 T  01041005312
                START OF SEGMENT *********  107
COMMENT 258;          09087000 T  01041005411
COMMENT 260;          09088000 T  01041005411
COMMENT 262;          09089000 T  01041005411
COMMENT 264;          09090000 T  01041005411
COMMENT 266;          09091000 T  01041005411
COMMENT 268;          09092000 T  01041005411
COMMENT 270;          09093000 T  01041005411
COMMENT 272;          09094000 T  01041005411
COMMENT 274;          09095000 T  01041005411
COMMENT 276;          09096000 T  01041005411
COMMENT 278;          09097000 T  01041005411
COMMENT 280;          09098000 T  01041005411
COMMENT 282;          09099000 T  01041005411
COMMENT 284;          09100000 T  01041005411
COMMENT 286;          09101000 T  01041005411
COMMENT 288;          09102000 T  01041005411
COMMENT 290;          09103000 T  01041005411
COMMENT 292;          09104000 T  01041005411
COMMENT 294;          09105000 T  01041005411
COMMENT 296;          09106000 T  01041005411
COMMENT 298;          09107000 T  01041005411
COMMENT 300;          09108000 T  01041005411
COMMENT 302;          09109000 T  01041005411
COMMENT 304;          09110000 T  01041005411
COMMENT 307;          09111000 T  01041005411
COMMENT 309;          09112000 T  01041005411
COMMENT 311;          09113000 T  01041005411
COMMENT 313;          09114000 T  01041005411
COMMENT 315;          09115000 T  01041005411
COMMENT 318;          09116000 T  01041005411
COMMENT 320;          09117000 T  01041005411
COMMENT 323;          09118000 T  01041005411
COMMENT 326;          09119000 T  01041005411
COMMENT 328;          09120000 T  01041005411
COMMENT 330;          09121000 T  01041005411
COMMENT 333;          09122000 T  01041005411
COMMENT 335;          09123000 T  01041005411
COMMENT 337;          09124000 T  01041005411
COMMENT 339;          09125000 T  01041005411
COMMENT 341;          09126000 T  01041005411
COMMENT 343;          09127000 T  01041005411
                %A    09128000 T  01041005411
COMMENT 347;          09129000 T  01041005411
COMMENT 349;          09130000 T  01041005411
COMMENT 352;          09131000 T  01041005411
COMMENT 355;          09132000 T  01041005411
COMMENT 357;          09133000 T  01041005411
COMMENT 359;          09134000 T  01041005411
COMMENT 361;          09135000 T  01041005411
COMMENT 364;          09136000 T  01041005411
COMMENT 366;          09137000 T  01041005411
COMMENT 368;          09138000 T  01041005411
COMMENT 370;          09139000 T  01041005411
COMMENT 372;          09140000 T  01041005411
COMMENT 374;          09141000 T  01041005411
```

```
0CT1250000000000000, "3NOTOO",                                  COMMENT 377;        09142000 T  01041005411
0CT1320000430000624, "2ORDOO",                                  COMMENT 379;        09143000 T  01041005411
0CT0430000120000000, "3OUTOO",                                  COMMENT 381;        09144000 T  01041005411
0CT0430000010000476, "3OWNOO",                                  COMMENT 383;        09145000 T  01041005411
0CT0430000160000463, "9PROCE", "DURE0000",                      COMMENT 385;        09146000 T  01041005411
0CT0440000000000000, "4READO",                                  COMMENT 388;        09147000 T  01041005411
0CT0430000040000000, "4REALO",                                  COMMENT 390;        09148000 T  01041005411
0CT0650000000000000, "7RELEA", "SE000000",                      COMMENT 392;        09149000 T  01041005411
0CT0510000000000000, "6REWIN", "D0000000",                      COMMENT 395;        09150000 T  01041005411
0CT0430000020000773, "4SAVEO",                                  COMMENT 398;        09151000 T  01041005411
0CT0460000000000000, "5SPACE",                                  COMMENT 400;        09152000 T  01041005411
0CT1110000000000000, "4STEPO",                                  COMMENT 402;        09153000 T  01041005411
0CT0430000220000000, "6STREA", "M0000000",                      COMMENT 404;        09154000 T  01041005411
0CT0430000150000562, "6SWITC", "H0000000",                      COMMENT 407;        09155000 T  01041005411
0CT1120000000000000, "4THENO",                                  COMMENT 410;        09156000 T  01041005411
0CT1130000000000000, "2TOOOO",                                  COMMENT 412;        09157000 T  01041005411
0CT0360000010000000, "4TRUEO",                                  COMMENT 414;        09158000 T  01041005411
0CT0560000000000000, "5UNTIL",                                  COMMENT 416;        09159000 T  01041005411
0CT1140000000000000, "5VALUE",                                  COMMENT 418;        09160000 T  01041005411
0CT0540000000000540, "5WHILE",                                  COMMENT 420;        09161000 T  01041005411
0CT1150000000000000, "4WITHO",                                  COMMENT 422;        09162000 T  01041005411
0CT0450000000000531, "5WRITE",                                  COMMENT 424;        09163000 T  01041005411
0CT0130000000140673, "3ABSOO",                0CT0000000000700000,%426              09164000 T  01041005411
0CT0130000000060000, "6ARCTA", "N0000000",    0CT0000000001600000,%429;             09165000 T  01041005411
0CT0130000000040000, "3COSOO",                0CT0000000001500000,%433;             09166000 T  01041005411
0CT0130000000360000, "6ENTIE", "R0000000",    0CT0000000001100000,%436              09167000 T  01041005411
0CT0130000000040000, "3EXPOO",                0CT0000000002000000,%440;             09168000 T  01041005411
0CT0130000000040000, "2LNOOO",                0CT0000000001700000,%443;             09169000 T  01041005411
0CT0130000000240000, "4SIGNO",                0CT0000000001000000,%446              09170000 T  01041005411
0CT0130000000040000, "3SINOO",                0CT0000000001400000,%449;             09171000 T  01041005411
0CT0130000000040515, "4SQRTO",                0CT0000000001300000,%452;             09172000 T  01041005411
0CT0130000000440000, "4TIMEO",                0CT0000000001200000,%455              09173000 T  01041005411
0CT0140000000040000, "3ZIPOO",  0,                                COMMENT 455;        09174000 T  01041005411
0CT0130000000060000, "9OUTPU", "T(W)0000",    0CT0000000000100000,%461;             09175000 T  01041005411
0CT0130000050060000, "1BLOCK", " CONTROL",    0CT0000000000200000,%465;             09176000 T  01041005411
0CT0130000000060000, "8INPUT", "(W)00000",    0CT0000000000300000,%469;             09177000 T  01041005411
0CT0000000000060000, "4SORTO", 0,             0CT0000000000400000,% 473             09178000 T  01041005411
0CT0130000000040000, "4DUMPO",                0CT0000000000500000,%477;             09179000 T  01041005411
0CT0130000000060000, "#X TO ", "THE IOOO",    0CT0000000000600000,%480;             09180000 T  01041005411
0CT0130000000060000, "1GO TO", " SOLVER ",    0CT0000000002100000,%484;             09181000 T  01041005411
0CT0130000140060000, "1ALGOL", " WRITE  ",    0CT0000000002200000,%488;             09182000 T  01041005411
0CT0130000150060000, "1ALGOL", " READ   ",    0CT0000000002300000,%492;             09183000 T  01041005411
0CT0130000160060000, "1ALGOL", " SELECT ",    0CT0000000002400000,%496;             09184000 T  01041005411
0CT0000000000040000, "5MERGE",    0CT0000000002700000,% 500                         09184100 T  01041005411
0CT0130000000560652, "6STATU", "S0000000",    0CT0000000003000000,%503              09184200 T  01041005411
0CT0130000000640000, "3MAXOO",                0CT0000000003100047,%507              09184300 T  01041005411
0CT0430000240000000, "6AUXME", "M0000000";                        %510              09185000 T  01041005411
                                                                                107 IS  257 LONG, NEXT SEG  104
                                                                                      09185100 T  01041005411
                                                                                      09186000 T  01041005411
                                                                                      09187000 T  01041005411
                                                                                      09188000 T  01041005411
                                                                                      09189000 T  01041005512
                                                                              START OF SEGMENT **********  108
                                                                                      09190000 T  01041005610
                                                                                      09191000 T  01041005610
                                                                                      09192000 T  01041005610
                                                                                      09193000 T  01041005610
```

```
 COMMENT  THIS IS THE FILL FOR STACKHEAD;
 FILL STACKHEAD[*] WITH
320,359,313,458,385,  0,337,347,383,361,379,412,  0,  0,  0,372,

355,309,  0,  0,368,  0,446,  0,  0,549,311,410,  0,  0,400,  0,
  0,  0,  0,  0,503,315,  0,  0,330,  0,  0,  0,  0,  0,  0,440,
390,449,349,  0,414,  0,  0,  0,452,  0,  0,  0,  0,  0,  0,  0,
381,328,  0,  0,  0,  0,  0,  0,  0,  0,436,  0,402,407,  0,
```

```
  0,  0,377,  0,  0,  0,416,455,355,  0,  0,357,  0,552,374,  0,
  0,  0,433,  0,418,398,343,  0,  0,326,339,  0,  0,  0,366,  0,
  0,  0,  0,  0,  0,422,  0,392,  0,  0,  0,424,  0;

 FILL SUPERSTACK[*] WITH
  0,  0,313,  0,307,  0,337,347,383,  0,379,412,  0,  0,  0,372,    %WF
335,309,  0,  0,  0,  0,  0,  0,  0,  0,  0,420,410,  0,  0,  0,  0,  %WF
  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  %WF
390,  0,364,  0,414,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  %WF
  0,328,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,402,  0,  0,  %WF
  0,  0,377,  0,510,  0,416,  0,355,  0,  0,357,  0,  0,  0,  0,
  0,  0,  0,  0,  0,398,  0,  0,  0,  0,326,339,  0,  0,  0,366,  0,  %WF
  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,424,  0;                  %WF

 COMMENT THIS IS THE FILL FOR THE SPECIAL CHARACTORS;
 FILL SPECIAL[*] WITH
  OCT1200000000200000,  COMMENT  #;   OCT0000000000100000,  COMMENT  @;
  OCT0000000000000000,                OCT1160000000120000,  COMMENT  !;
  OCT1340000450002763,  COMMENT  >;   OCT1340000250002662,  COMMENT  ≥;
  OCT1350000200000000,  COMMENT  +;   OCT0000000000000000,
  OCT1220000000060000,  COMMENT  ,;   OCT1210000000000000,  COMMENT  [;
  OCT1270000000000000,  COMMENT  &;   OCT0420000000000000,  COMMENT  (;
  OCT1340010450003571,  COMMENT  <;   OCT1260000000000000,  COMMENT  ←;
  OCT1360001000000000,  COMMENT  ×;   OCT0000000000000000,
  OCT0000000000040000,  COMMENT  $;   OCT1370000000000000,  COMMENT  *;
  OCT1350000600000000,  COMMENT  -;   OCT1240000000160000,  COMMENT  );
  OCT0620000000000000,  COMMENT  .;   OCT1340010250003470,  COMMENT  .≤;
  OCT0000000000000000,                OCT1360002000000000,  COMMENT  /;
  OCT1170000000000000,  COMMENT  ,;   OCT0000000000020000,  COMMENT  %;
  OCT1340001050002561,  COMMENT  #;   OCT1340011050002460,  COMMENT  =;
  OCT1230000000000000,  COMMENT  ];   OCT0000000000140000,  COMMENT  ";
   0,0;

 COMMENT THIS IS THE FILL FOR THE REALLY SPECIAL CHARACTERS FOR DATACOM;
FILL INFO[2,*] WITH OCT0030000120000000, "2LB000",  % THESE ENTRIES ARE
                    OCT0030000130000000, "2RB000",  % DESIGNED TO LOOK
                    OCT0030000140000000, "3GTR00",  % LIKE DEFINE
                    OCT0030000150000000, "3GEQ00",  % DECLARATIONS AT
                    OCT0030000160000000, "3EQL00",  % BLOCK LEVEL 0.
                    OCT0030000170000000, "3NEQ00",
                    OCT0030000200000000, "3LEQ00",
                    OCT0030000210000000, "3LSS00",
                    OCT0030000220000000, "5TIMES",
                    OCT0030000230000000, "5INPUT",
                    OCT0030000240000000, "2I0000",
                    OCT0030000250000000, "6SERIA","L0000000",
                    OCT0030000260000000, "6RAND0","M0000000",
                    OCT0030000270000000, "6UPDAT","E0000000",
                     OCT0030000300000000, "6OUTPU","T0000000",
                    OCT0030000310000000, "7CANTU","SE000000",
  OCT0130000000740000, "3MIN00",              OCT0000000003200000,%549
  OCT0130000001040000, "5DELAY",              OCT0000000003300000,%552
  OCT0000000000060000, "!SUPER", " MOVER  ",  OCT0000000003400000,%555
  OCT0000000000060000, "!DYNAM", "IC DIALS",  OCT0000000004000000,%559
```

```
                                                    09194000 T  0104!005610
                                                    09195000 T  0104!005610
                                                    09196000 T  0104!005610
                                          108 IS  125 LONG, NEXT SEG  104
                                                    09196100 T  0104!005610
                                                    09196200 T  0104!005611
                                        START OF SEGMENT **********  109
                                                    09196300 T  0104!005810
                                                    09196400 T  0104!005810
                                                    09196500 T  0104!005810
                                                    09196600 T  0104!005810
                                                    09196700 T  0104!005810
                                                    09196800 T  0104!005810
                                                    09196900 T  0104!005810
                                          109 IS  125 LONG, NEXT SEG  104
                                                    09197000 T  0104!005810
                                                    09198000 T  0104!005810
                                                    09199000 T  0104!005811
                                        START OF SEGMENT **********  110
                                                    09200000 T  0104!005913
                                                    09201000 T  0104!005913
                                                    09202000 T  0104!005913
                                                    09203000 T  0104!005913
                                                    09204000 T  0104!005913
                                                    09205000 T  0104!005913
                                                    09206000 T  0104!005913
                                                    09207000 T  0104!005913
                                                    09208000 T  0104!005913
                                                    09209000 T  0104!005913
                                                    09210000 T  0104!005913
                                                    09211000 T  0104!005913
                                                    09212000 T  0104!005913
                                                    09213000 T  0104!005913
                                                    09214000 T  0104!005913
                                          110 IS   32 LONG, NEXT SEG  104
                                                    09214100 T  0104!005913
                                                    09214105 T  0104!005913
                                        START OF SEGMENT **********  111
                                                    09214110 T  0104!006113
                                                    09214115 T  0104!006113
                                                    09214120 T  0104!006113
                                                    09214125 T  0104!006113
                                                    09214130 T  0104!006113
                                                    09214135 T  0104!006113
                                                    09214140 T  0104!006113
                                                    09214145 T  0104!006113
                                                    09214150 T  0104!006113
                                                    09214155 T  0104!006113
                                                    09214160 T  0104!006113
                                                    09214165 T  0104!006113
                                                    09214170 T  0104!006113
                                                    09214180 T  0104!006113
                                                    09214190 T  0104!006113
                                                    09214200 T  0104!006113
                                                    09214210 T  0104!006113
                                                    09214220 T  0104!006113
                                                    09214230 T  0104!006113
```

```
                0CT0130000000060000, "5FILE ", "ATTRBUTS", 0CT0000000015000000,%563      09214240 T  0104100611 3
                0CT0000000000040000, "5DCPWR",             0CT0000000005600000,%567      09214250 T  0104100611 3
                0CT0000000000040000, "5DCMTH",             0CT0000000005500000,%570      09214255 T  0104100611 3
                0CT0130000001140000, "5DSQRT",             0CT0000000012300000,%573      09214260 T  0104100611 3
                0CT0130000001240000, "4CEXPO",             0CT0000000010000000,%576      09214270 T  0104100611 3
                0CT0130000001340000, "3CLN00",             0CT0000000010200000,%579      09214295 T  0104100611 3
                0CT0130000001440000, "4CSIN0",             0CT0000000010600000,%582      09214300 T  0104100611 3
                0CT0130000001540000, "4CCOS0",             0CT0000000011000000,%585      09214305 T  0104100611 3
                0CT0130000001640000, "5CSQRT",             0CT0000000012400000,%588      09214310 T  0104100611 3
                0CT0130000001740000, "4DEXPO",             0CT0000000007700000,%591      09214315 T  0104100611 3
                0CT0130000002040000, "3DLN00",             0CT0000000010100000,%594      09214320 T  0104100611 3
                0CT0130000002140000, "4DSIN0",             0CT0000000010500000,%597      09214325 T  0104100611 3
                0CT0130000002240000, "4DCOS0",             0CT0000000010700000,%600      09214330 T  0104100611 3
                0CT0130000002360000, "7DARCT", "AN000000", 0CT0000000011300000,%603      09214340 T  0104100611 3
                0CT0130000002460000, "6DLOG1", "00000000", 0CT0000000010400000,%607      09214345 T  0104100611 3
                0CT0130000002560000, "8DARCT", "AN200000", 0CT0000000011500000,%611      09214350 T  0104100611 3
                0CT0130000002640000, "4DMOD0",             0CT0000000006500000,%615      09214355 T  0104100611 3
                0CT0130000002740000, "4CABS0",             0CT0000000005300000,%618      09214360 T  0104100611 3
                0CT0130000003060000, "7ARCTA", "N2000000", 0CT0000000011400000,%621      09214365 T  0104100611 3
                0CT0130000003160000, "6DROUN", "D0000000", 0CT0000000006100000,%625      09214370 T  0104100611 3
                0CT0130000000040000, "5LOG10",             0CT0000000010300000,%629      09214375 T  0104100611 3
                0CT0130000000040000, "5COTAN",             0CT0000000011200000,%632      09214380 T  0104100611 3
                0CT0130000000060000, "6ARCSI", "N0000000", 0CT0000000011600000,%635      09214385 T  0104100611 3
                0CT0130000000040000, "5ARCOS",             0CT0000000011700000,%639      09214390 T  0104100611 3
                0CT0130000000040000, "4SINH0",             0CT0000000012000000,%642      09214395 T  0104100611 3
                0CT0130000000040000, "4COSH0",             0CT0000000012100000,%645      09214400 T  0104100611 3
                0CT0130000000040000, "4TANH0",             0CT0000000012200000,%648      09214405 T  0104100611 3
                0CT0130000000040000, "3ERF00",             0CT0000000012500000,%651      09214410 T  0104100611 3
                0CT0130000000040000, "5GAMMA",             0CT0000000012600000,%654      09214415 T  0104100611 3
                0CT0130000000040000, "5LNGAM",             0CT0000000012700000,%657      09214420 T  0104100611 3
                0CT0130000000040000, "3TAN00",             0CT0000000011100007,%660      09214425 T  0104100611 3
                0CT0030000260000000, "4FASTO",                                 %663      09214426 T  0104100611 3
                0CT0030000270000000, "4SLOWO",                                 %665      09214427 T  0104100611 3
                0CT0030000240000000, "7PROTE", "CT000000",                     %667      09214428 T  0104100611 3
                0CT2000000000004050, COMMENT POWERS OF TEN ;                   %670      09214430 T  0104100611 3
                0CT0430000250000000, "5FIELD",                                 %671      09214432 C  0104100611 3
                                  0, ">SORT ", "TEMPORAR", "Y0000000", % SORTA %673      09214435 P  0104100611 3
                "       " ;          COMMENT LASTSEQUENCE,LASTSEQROW ;         %674      09214440 P  0104100611 3
                                                                                    111 IS  166 LONG, NEXT SEG  104
                 COMMENT NOW LINK THESE ENTRIES INTO STACKHEAD;                              09214500 T  0104100611 3
                FOR NEXTINFO←512 STEP 2 UNTIL 534,537 STEP 3 UNTIL 546                      09214510 T  0104100611 3
                ,567STEP 3UNTIL 603,607STEP 4UNTIL 615,618,621STEP 4UNTIL 629,632,635,      09214515 T  0104100681 0
                639 STEP 3 UNTIL 660,663 STEP 2 UNTIL 667, 671                      %117-   09214516 P  0104100861 1
                DO PUT(TAKE(NEXTINFO)&STACKHEAD[GT2←TAKE(NEXTINFO+1)MOD 125][35:35:13],     09214520 T  0104100951 2
STACK(F←2) = *TEMPORARY STORAGE*
                          LASTINFO←STACKHEAD[GT2]←NEXTINFO);                                09214530 T  0104101001 1
                 NEXTINFO ← LASTINFO ← LASTSEQROW × 256 + LASTSEQUENCE + 1;                 09214980 T  0104101061 0
                 BUILDLINE.[45:1]←TRUE ;                                                    09214985 T  0104101081 1
                 PUTNBUMP(0);                                                               09214990 T  0104101101 1
                    FILL MACRO[*] WITH                                                      09215000 T  0104101111 2
                         OCT0131,          COMMENT SFS A      00          ;                09216000 T  0104101111 3
                                                                                  START OF SEGMENT **********  112
                         OCT0116,          COMMENT SFD A      01          ;                09217000 T  0104101131 2
                         OCT0000,          COMMENT SYNTAX ERROR02         ;                09218000 T  0104101131 2
                         OCT0140,          COMMENT INC A      03          ;                09219000 T  0104101131 2
                         OCT0130,          COMMENT SRS A      04          ;                09220000 T  0104101131 2
                         OCT0117,          COMMENT SRD A      05          ;                09221000 T  0104101131 2
```

```
          0CT0000,          COMMENT SYNTAX ERROR06                ;  09222000 T  01041011312
          0CT0000,          COMMENT SYNTAX ERROR07                ;  09223000 T  01041011312
          0CT00310143,      COMMENT CRF A, SFS 008                ;  09224000 T  01041011312
          0CT00160143,      COMMENT CRF A, SFD 009                ;  09225000 T  01041011312
          0CT00470143,        COMMENT CRF A,  JFW 0  10           ;  09226000 T  01041011312
          0CT00400143,      COMMENT CRF A, INC 011                ;  09227000 T  01041011312
          0CT00300143,      COMMENT CRF A, SRS 012                ;  09228000 T  01041011312
          0CT00170143,      COMMENT CRF A, SRD 013                ;  09229000 T  01041011312
          0CT0000,          COMMENT SYNTAX ERROR14                ;  09230000 T  01041011312
          0CT0000,          COMMENT SYNTAX ERROR15                ;  09231000 T  01041011312
          0CT0153,          COMMENT RSA A       16                ;  09232000 T  01041011312
          0CT0104,          COMMENT RDA A       17                ;  09233000 T  01041011312
          0CT0150,          COMMENT RCA A       18                ;  09234000 T  01041011312
     0CT004201430042,       COMMENT SEC 0, CRF A, SEC 0 19        ;  09235000 T  01041011312
          0CT0122,          COMMENT SES A       20                ;  09236000 T  01041011312
          0CT0106,          COMMENT SED A       21                ;  09237000 T  01041011312
          0CT0000,          COMMENT SYNTAX ERROR22                ;  09238000 T  01041011312
          0CT0000,          COMMENT SYNTAX ERROR23                ;  09239000 T  01041011312
          0CT0056,          COMMENT TSA 0       24                ;  09240000 T  01041011312
          0CT0000,          COMMENT SYNTAX ERROR25                ;  09241000 T  01041011312
          0CT0000,          COMMENT SYNTAX ERROR26                ;  09242000 T  01041011312
          0CT0000,          COMMENT SYNTAX ERROR27                ;  09243000 T  01041011312
          0CT0000,          COMMENT SYNTAX ERROR28                ;  09244000 T  01041011312
          0CT0007,          COMMENT TDA 0       29                ;  09245000 T  01041011312
          0CT0000,          COMMENT SYNTAX ERROR30                ;  09246000 T  01041011312
          0CT0000,          COMMENT SYNTAX ERROR31                ;  09247000 T  01041011312
          0CT0115,          COMMENT SSA A       32                ;  09248000 T  01041011312
          0CT0114,          COMMENT SDA A       33                ;  09249000 T  01041011312
          0CT0154,          COMMENT SCA A       34                ;  09250000 T  01041011312
          0CT0141;          COMMENT STC A       35                ;  09251000 T  01041011312
                                                                      112 IS   36 LONG, NEXT SEG  104
 FILL TEXT[0,*] WITH 0,0,0,0,0,0,0,0,0,0,                             09251010 T  01041011312
                                                                    START OF SEGMENT **********  113
          "[#       ",                                               09251020 T  01041011512
          "]#       ",                                               09251030 T  01041011512
          ">#       ",                                               09251040 T  01041011512
          "≥#       ",                                               09251050 T  01041011512
          "=#       ",                                               09251060 T  01041011512
          "≠#       ",                                               09251070 T  01041011512
          "≤#       ",                                               09251080 T  01041011512
          "<#       ",                                               09251090 T  01041011512
          "×#       ",                                               09251100 T  01041011512
          "1#       ",                                               09251101 T  01041011512
          "3#       ",                                               09251102 T  01041011512
          "0#       ",                                               09251103 T  01041011512
          "1#       ",                                               09251104 T  01041011512
          "2#       ",                                               09251105 T  01041011512
          "2#       ",                                               09251106 T  01041011512
          "0#       "                                                09251107 T  01041011512
 ;                                                                    09251200 T  01041011512
                                                                      113 IS   26 LONG, NEXT SEG  104
 NEXTTEXT←26 ;                                                        09251300 T  01041011512
               DO UNTIL STEPI = BEGINV;                               09252000 T  01041011513
  BUILDLINE,[45:1]←FALSE;                                             09252050 T  01041011713
                                                                      09252100 T  01041011912
    COMMENT THE FOLLOWING IS THE FIRST CODE EXECUTED IN ANY PROGRAM,  09253000 T  01041011912
            THE OUTER BLOCK(NUMBER 1) CONSISTS OF THE FOLLOWING CODE; 09254000 T  01041011912
```

```
      LITC 0  ---     THIS PUTS A BOTTOM ON THE STACK                          09255000 T  01041011912
                      AND IS ALSO USED AS A ONE SYLLABLE                      09256000 T  01041011912
                      CHARACTER MODE PROGRAM TO CAUSE AN EXIT.                09257000 T  01041011912
                      ITS PRIMARY FUNTION IS TO CUT BACK                      09258000 T  01041011912
                      THE STACK AFTER A COMMUNICATE OPERATOR.                 09259000 T  01041011912
      MKS     ---     THIS SETS THE PROGRAM UP FOR RUNNING                    09260000 T  01041011912
                      IN SUBPROGRAM LEVEL.THIS IS TO ALLOW                    09261000 T  01041011912
                      C-RELATIVE ADDRESSING FOR CONSTANTS                     09262000 T  01041011912
                      IN THE PROGRAM STREAM                                   09263000 T  01041011912
      OPDC XXXX---    THIS ACCESSES A PROGRAM DESCRIPTOR                      09264000 T  01041011912
                      THAT GETS THE PROGRAM INTO SUBPROGRAM                   09265000 T  01041011912
                      LEVEL. XXXX IS THE FIRST AVAILABLE PRT                  09266000 T  01041011912
                      CELL.AT THE START OF COMPILATION XXXX IS                09267000 T  01041011912
                      ASSUMED TO CONTAIN A LABEL DESCRIPTOR                   09268000 T  01041011912
                      IT IS CHANGED BEFORE COMPILATION IS                     09269000 T  01041011912
                      COMPLETE TO LOOK LIKE A WORD MODE                       09270000 T  01041011912
                      PROGRAM DESCRIPTOR;                                     09271000 T  01041011912
    EMITL(0);EMITO(MKS);                                                      09272000 T  01041011912
    GT1←PROGDESCBLDR(3,0,0);                                                  09273000 T  01041012011
    GT1 := GETSPACE(TRUE,-5); % SEG.#2 DESCR.                                 09274000 T  01041012211
      INSERTCOP:=1;                                          %107-            09274100 C  01041012410
     ERRORTOG←TRUE; BLOCK(FALSE);                                             09275000 T  01041012610
COMMENT THIS CODE WILL PUT AN EXTRA CARD ON OCRDIMG TAPE                      09275100 T  01041012713
   THUS AVOIDING E.O.F. NO LABEL CONDITION WHEN PATCHING                      09275200 T  01041012713
  THE END. CARD OFF AN INPUT TAPE;                                            09275250 T  01041012713
IF NEWTOG THEN                                                                09275300 T  01041012713
BEGIN FILL LIBARRAY[*] WITH "END;END.","        ","LAST CAR",                 09275350 T  01041012811
                                                                  START OF SEGMENT ********** 114
     "D ON OCR","DING TAP","E       ", "        ","        ",                 09275400 T  01041013011
     "        ","99999999";                                                    09275450 T  01041013011
                                                                  114 IS   10 LONG, NEXT SEG  104
          WRITE(NEWTAPE,10,LIBARRAY[*])                                       09275500 T  01041013011
     END;                                                                     09275550 T  01041013313
                                                                              09275600 T  01041013512
                                                                              09275650 T  01041013512
                                                                              09275700 T  01041013512
                                                                              09275750 T  01041013512
                                                                              09275800 T  01041013512
                                                                              09275850 T  01041013512
                                                                              09275900 T  01041013512
                                                                              09275950 T  01041013512
                                                                              09276000 T  01041013512
COMMENT THE FOLLOWING CODE SEARCHES THROUGH INFO TO DETERMINE                 09277000 T  01041013512
        WHICH INTRINSICS HAVE BEEN USED.IF AN INTRINSIC HAS BEEN              09278000 T  01041013512
        USED THEN A PRT ADDRESS WILL HAVE BEEN ASSIGNED AND                   09279000 T  01041013512
        THIS INDICATES THAT A DESCRIPTOR MUST BE BUILT FOR PLACING            09280000 T  01041013512
        IN THE PRT.POWERSOFTEN IS ENTERED IN THE OBJECT PROGRAM               09281000 T  01041013512
        PRT AS AN ABSENT DATA DESCRIPTOR.IT MAY BE RECOGNIZED IN              09282000 T  01041013512
        INFO BECAUSE IT IS MINUS. THE FIRST WORD IN EACH OF THESE             09283000 T  01041013512
        ENTRIES LOOKS LIKE THE REST OF INFO EXCEPT THAT THE INCR              09284000 T  01041013512
        FIELD IS BROKEN INTO 2 PARTS. [33:2] IS USED TO ADD TO THE            09285000 T  01041013512
        INDEX OF CURRENT WORD TO LINK TO NEXT ENTRY.THE REST OF               09286000 T  01041013512
        THE INCR FIELD IS USED BY IMPFUN. THE ADDITIONAL INFO                 09287000 T  01041013512
        PORTION INDICATES AN INDEX THAT ALLOWS THE MCP TO ASSIGN              09288000 T  01041013512
        DRUM ADDRESSES TO THE INTRINSICS;                                     09289000 T  01041013512
                                                                              09290000 T  01041013512
    GT1 ← GT3 ← STARTINTRSC;                                                  09291000 T  01041013512
L1: GT1 ← GT1 + (GT2 ← INFO[GT1.LINKR,GT1.LINKC]).[33:2];                     09292000 T  01041013610
```

```
          IF GT2 ≥ 0 THEN  %                  NOT POWERS OF TEN TABLE                  09293000 T  0104!014!2
          BEGIN IF GT2.ADDRESS ≠ 0 THEN % IT WAS USED                                  09294000 T  0104!014!0
                BEGIN SGNO ← SGAVL; SGAVL ← SGAVL + 1;                                 09295000 T  0104!014!3
                      GT2 ← PROGDESCBLDR(INFO[GT1.LINKR,GT1.LINKC].[1:1]               09296000 T  0104!014!0
                           x 2 + 1, 0, GT2.ADDRESS);                                   09296100 T  0104!014!1
                      PDPRT[PDINX,[37:5],PDINX,[42:6]] ←                               09297000 T  0104!015!0
                       1 & INFO[GT1.LINKR,GT1.LINKC][13:18:15]                         09298000 T  0104!015!0
                         & SGNO[28:38:10] & 1[2:47:1];                                 09298100 T  0104!015!3
                      PDINX ← PDINX + 1;                                               09299000 T  0104!016!0
                      IF PRTOG THEN  % WRITE OUT INTRINSICS USED.                      09300000 T  0104!016!2
                      BEGIN GT3 ← GT3 + 1;                                             09300100 T  0104!016!0
                            BLANKET(14,LIN); % BLANK BUFFER.                           09300150 T  0104!016!3
                            WRTINTRSC(SGNO, INFO[GT3.LINKR,GT3.LINKC].                 09300200 T  0104!016!2
                                      B2D(GT2,[38:10]), LIN);                          09301000 T  0104!016!0
                            IF NOHEADING THEN DATIME; WRITELINE;                       09302000 T  0104!017!1
                      END                                                              09303000 T  0104!018!0
                END;                                                                   09304000 T  0104!018!0
                GT3 ← GT1 ← GT1 + INFO[GT1.LINKR,GT1.LINKC].[33:15] + 1;               09305000 T  0104!018!0
                GO TO L1;                                                              09305100 T  0104!018!1
            END;                                                                       09306000 T  0104!018!2
              L←L-1; COMMENT WIPES OUT EXTRANEOUS BFW EMITTED BY BLOCK;                09306100 T  0104!018!2
          EMITL(5);EMITO(COM);                                                         09307000 T  0104!018!1
      ENIL[0,1] ← 1023 & 99999999[10:20:28];  ENILPTR ← 1;                             09307100 T  0104!019!0
          SEGMENT((L+3) DIV 4,1,0);                                                    09308000 T  0104!019!3
  COMMENT IF THE POWERS-OF-TEN TABLE HAS BEEN USED, IT IS WRITTEN OUT                  09309000 T  0104!019!0
          AT THIS TIME AS A TYPE 2 SEGMENT;                                            09310000 T  0104!019!0
          IF GT1←GT2.ADDRESS≠0 THEN                                                    09311000 T  0104!019!0
          BEGIN SGAVL←(SGNO←SGAVL)+1;                                                  09312000 T  0104!019!3
                GT2←PROGDESCBLDR(2,0,GT2.ADDRESS);                                     09313000 T  0104!020!0
                MOVE(69,TEN,EDOC[0,0]);                                                09314000 T  0104!020!0
                BUILDLINE ← BOOLEAN(2×REAL(BUILDLINE));                                09314100 T  0104!020!2
                SEGMENT(-69, SGNO,0);                                                  09315000 T  0104!020!0
                BUILDLINE ← BUILDLINE.[46:1] ;                                         09315100 T  0104!020!3
              END;                                                                     09316000 T  0104!020!2
        BEGIN ARRAY PRT[0:7,0:127],SEGDICT[0:7,0:127];                                 09317000 T  0104!020!2
PRT(1044) = *SEGMENT DESCRIPTOR*

                                                                  START OF SEGMENT **********  115

STACK(F+3) = PRT
STACK(F+4) = SEGDICT
                    INTEGER PRTADR,SEGMNT,LINK;                                        09318000 T  0115!000!3
STACK(F+5) = PRTADR
STACK(F+6) = SEGMNT
STACK(F+7) = LINK
          COMMENT THE PRT AND SEGMENT DICTIONARY ARE NOW BUILT;                        09333000 T  0115!000!3
                                                                                       09334000 T  0115!000!3
                                                                                       09335000 T  0115!000!3
                                                                                       09336000 T  0115!000!3
                                                                                       09337000 T  0115!000!3
                                                                                       09338000 T  0115!000!3
                                                                                       09339000 T  0115!000!3
                                                                                       09340000 T  0115!000!3
                                                                                       09341000 T  0115!000!3
                                                                                       09342000 T  0115!000!3
                                                                                       09343000 T  0115!000!3
                                                                                       09344000 T  0115!000!3
                                                                                       09345000 T  0115!000!3
                                                                                       09346000 T  0115!000!3
```

```
                    FOR I←0 STEP 1 UNTIL PDINX-1 DO                                    09347000 T  0115:0005:3
                      IF (GT1←PDPRT[I,[37:5],I,[42:6]]),[38:10]=0 THEN                 09348000 T  0115:0005:3
                BEGIN PRTADR←GT1,[8:10]; SEGMNT←GT1,[28:10];                           09349000 T  0115:0009:3
                      LINK←SEGDICT[SEGMNT,[36:5],SEGMNT,[41:7]],[8:10];                09350000 T  0115:0013:3
                      MDESC(GT1,[18:10]&SEGMNT[18:33:15]                               09351000 T  0115:0016:1
                           &(IF LINK=0 THEN SEGMNT+2048 ELSE LINK)                     09352000 T  0115:0019:3
                           [6:36:12]&GT1[4:4:2]&5[1:45:3],                             09353000 T  0115:0021:2
                           PRT[PRTADR DIV 128,PRTADR MOD 128]);                        09354000 T  0115:0023:3
                      SEGDICT[SEGMNT,[36:5],SEGMNT,[41:7]],[8:10]←PRTADR;              09354100 T  0115:0026:1
                END ELSE                                                               09355000 T  0115:0029:3
                BEGIN SEGMNT←GT1,[28:10];                                              09356000 T  0115:0033:3
                      SEGDICT[SEGMNT,[36:5],SEGMNT,[41:7]]←                            09357000 T  0115:0033:3
                      SEGDICT[SEGMNT,[36:5],SEGMNT,[41:7]]&GT1[23:38:10]               09358000 T  0115:0037:2
                         & GT1[33:13:15] & GT1[4:3:1] & GT1[1:1:2];                     09359000 T  0115:0039:3
                END;                                                                   09360000 T  0115:0042:1
           COMMENT SET UP NEWINX = TOTAL SEGMENT SIZE;  NEWINX←AKKUM;                  09361000 T  0115:0046:0
           COMMENT CODE TO ADD IN CORE STORAGE REQUIREMENTS;                           09361005 T  0115:0046:1
           GTI1←0;                                                                     09361010 T  0115:0047:3
           COMMENT ADD IN ARRAYS;                                                      09361020 T  0115:0047:3
           GTI1←GTI1+( IF NOOFARRAYS =0 THEN 0 ELSE IF NOOFARRAYS ≤4                   09361030 T  0115:0048:0
                       THEN 2000  ELSE IF NOOFARRAYS ≤ 8 THEN 3500                     09361040 T  0115:0048:0
                       ELSE 5000);                                                     09361050 T  0115:0050:1
           COMMENT ADD IN SEGMENT SIZE REQUIREMENTS;                                   09361060 T  0115:0053:2
           GTI1←GTI1+ (IF NEWINX ≤ 1000 THEN NEWINX ELSE IF NEWINX ≤2000               09361070 T  0115:0055:3
                       THEN 1000  ELSE NEWINX/2);                                      09361080 T  0115:0055:3
           COMMENT ADD IN STACK AND PRT;                                               09361100 T  0115:0058:0
           GTI1←GTI1+ 512 + PRT↑MAX;                                                   09361110 T  0115:0061:2
           COMMENT ADD IN JRT;                                                         09361120 T  0115:0061:2
           GTI1←GTI1 + ( (FILENO +1)× 5);                                              09361130 T  0115:0063:2
           COMMENT ADD IN I/O BUFFER REQUIREMENTS;                                     09361140 T  0115:0063:2
      GTI1←GTI1+IOBUFFSIZE; COMMENT I/O SIZE CAL. IN P.IODEC;                          09361150 T  0115:0065:2
           COMMENT ADD SEGMENT DICT.SIZE;                                              09361160 T  0115:0065:2
           GTI1←GTI1+ SGAVL-1;                                                         09361170 T  0115:0066:1
COMMENT    ADD IN CORE ESTIMATE FOR SORT;                                              09361180 T  0115:0066:1
           GTI1:=GTI1+CORESZ;                                                          09361181 T  0115:0068:0
           COMMENT  CHECK IF TOTAL IS MORE THAN 8 MODS;                                09361182 T  0115:0068:0
                IF GTI1 ≥ 32000 THEN GTI1← 32000;                                      09361190 T  0115:0069:3
           COMMENT AT THIS POINT  GTI1 HAS THE NEEDED TOTAL CORE REQD;                 09361200 T  0115:0069:3
COMMENT WRITE OUT FILE PARAMETER BLOCK;                                                09361210 T  0115:0071:3
      GT1←MIN((IDLOC-IDLOCTEMP).[33:15]+1, 128);% AHA                                  09393000 T  0115:0071:3
           MOVE(GT1,IDARRAY[0],EDOC[0,0]);                                             09394000 T  0115:0071:3
           ZEROUT(IDARRAY[0],0,30);                                                    09395000 T  0115:0076:0
           IDARRAY[4]:=MOVEANDBLOCK(EDOC,GT1,0);                          %106-        09395500 T  0115:0078:1
           IDARRAY[5]←GT1;                                                             09396000 P  0115:0080:0
COMMENT WRITE OUT SEGMENT DICTIONARY;                                                  09397000 T  0115:0083:2
           IDARRAY[0]:=MOVEANDBLOCK(SEGDICT,SGAVL,1);                     %106-        09398000 T  0115:0084:1
           IF BUILDLINE THEN IDARRAY[0]←IDARRAY[0]&MOVEANDBLOCK                        09399000 P  0115:0084:1
                             (LDICT,SGAVL,2)[18:33:15];                    %106-        09399100 T  0115:0087:3
           IDARRAY[1]←SGAVL;                                                           09399150 P  0115:0089:2
COMMENT WRITE OUT PRT;                                                                 09400000 T  0115:0092:0
           IDARRAY[2]:=MOVEANDBLOCK(PRT,PRTIMAX,3);                       %106-        09401000 T  0115:0093:2
           IDARRAY[3]←PRTIMAX;                                                         09402000 P  0115:0093:2
COMMENT MARK FIRST EXECUTABLE SEGMENT;                                                 09403000 T  0115:0096:0
           IDARRAY[6]←1;                                                               09404000 T  0115:0097:3
COMMENT PASS NUMBER OF FILES;                                                          09405000 T  0115:0097:3
           IDARRAY[7] ← (FILENO-1)&GTI1[18:27:15];                                     09405100 T  0115:0098:1
                                                                                       09405200 T  0115:0098:1
```

```
                COMMENT WRITE DISK SEGMENT ZERO;                                                    09406000 T  0115!010!3
                        GT1:=DA; DA:=0; MOVE(30,IDARRAY[0],PRT[0,0]);                %106-          09407000 P  0115!010!3
                        GT2:=MOVEANDBLOCK(PRT,30,6); DA:=GT1;                        %106-          09407010 C  0115!010!3
                        IF CODEFILE THEN WRITE(LINE);                                %106-          09407020 C  0115!010!2
                            IF SAVETIME ≥ 0 AND ERRORCOUNT = 0THEN                                  09407050 T  0115!011!0
                        LOCK(CODE,SAVE);                                                            09407100 T  0115!011!0
                         CLOSE(CARD,RELEASE);  % RELEASE PRIMARY INPUT FILE.         %119-          09407200 C  0115!011!0
                         CLOSE(TAPE,RELEASE);  % RELEASE SECONDARY INPUT FILE.       %119-          09407300 C  0115!012!0
                         LOCK(NEWTAPE,*);  % CLOSE WITH CRUNCH.                     %119-          09407400 C  0115!012!3
                        IF LISTER OR NOT NOHEADING THEN                                             09408000 T  0115!012!3
                        BEGIN FORMAT PAN("NUMBER OF ERRORS DETECTED =",I4,".  COMPILAT"             09409000 T  0115!012!1
PRT(1045) = *SEGMENT DESCRIPTOR*
                                                                                  START OF SEGMENT **********  116
                                                                                  START OF SEGMENT **********  117
PRT(1046) = PAN
                                  ,"ION TIME =",I5," SECONDS.",X22,2A4/                             09410000 T  0117!000!0
                                  "PRT SIZE =",I4,"; TOTAL SEGMENT SIZE =",I6,                      09411000 T  0117!000!0
                                " WORDS; DISK SIZE =",I4," SEGS; NO. PGM. SEGS =",                  09412000 T  0117!000!0
                               I4/"ESTIMATED CORE STORAGE REQUIRED =",I6," WORDS.",                 09413000 T  0117!000!0
                               /"ESTIMATED AUXILIARY MEMORY REQUIRED =",I6," WORDS.",               09414000 T  0117!000!0
                               /"NUMBER OF CARD-IMAGES PROCESSED = ",F7.0);                         09414100 T  0117!000!0
                                                                                   117 IS   66 LONG, NEXT SEG  116
                    FORMAT SERR("THERE WERE ",V8," SEQUENCE ERRORS");                               09414101 T  0116!000!0
                                                                                  START OF SEGMENT **********  118
PRT(1047) = SERR
                                                                                   118 IS    9 LONG, NEXT SEG  116
                               MOVECHARACTERS(4,INFO[LASTSEQROW,LASTSEQUENCE-1],0,GT1,4);           09415000 T  0116!000!0
                               MOVECHARACTERS(4,INFO[LASTSEQROW,LASTSEQUENCE-1],4,GT2,4);           09416000 T  0116!000!2
                            IF CHECKTOG       THEN                                                  09416001 T  0116!000!0
                                WRITE(LINE[DBL] ,SERR,IF NUMSEQUENCEERRORS = 0                      09416002 T  0116!000!2
PRT(1050) = *LIST, LABEL, OR SEGMENT DESCRIPTOR*
                                  THEN "A" ELSE "I", IF NUMSEQUENCEERRORS = 0                       09416004 T  0116!001!1
                                  THEN "    NO" ELSE NUMSEQUENCEERRORS);                            09416006 T  0116!001!1
                                     WRITE(LINE[DBL],PAN,ERRORCOUNT,(TIME(1)-TIME1)/60,GT1,GT2,     09417000 T  0116!002!1
PRT(1051) = *LIST, LABEL, OR SEGMENT DESCRIPTOR*
                                       PRTIMAX,AKKUM,IF DA≤CHUNK THEN DA ELSE ((DA+CHUNK-1)         09418000 T  0116!003!2
                                       DIV CHUNK)×CHUNK,SGAVL-1,GTI1,AUXMEMREQ,CARDCOUNT);          09419000 T  0116!004!0
                        END END  END PROGRAM;                                                       09420000 T  0116!005!3
PRT(1052) = *SEGMENT DESCRIPTOR*
                                                                                   116 IS   55 LONG, NEXT SEG  115
PRT(1053) = *SEGMENT DESCRIPTOR*
                                                                                   115 IS  134 LONG, NEXT SEG  104
                                                                                   104 IS  215 LONG, NEXT SEG    3


                COMMENT THIS SECTION CONTAINS GENERATORS USED BY THE BLOCK ROUTINE;                 10000000 T  0003!083!0
                COMMENT   FORMATPHRASE COMPILES A PSEUDO CODE USED BY THE OBJECT TIME               10001000 T  0003!083!0
                            FORMATING ROUTINES TO PRODUCE DESIRED I/O.  THERE IS ONE                10002000 T  0003!083!0
                            WORD OF PSEUDO CODE PRODUCED FOR EACH EDITING PHRASE.  IN               10003000 T  0003!083!0
                            ADDITION ONE WORD IS PRODUCED FOR EACH LEFT PARENTHESIS,                10004000 T  0003!083!0
                            RIGHT PARENTHESIS, AND STROKE.  EACH SIX CHARACTORS OF                  10005000 T  0003!083!0
                            STRING ALSO PRODUCES ONE WORD.  IN ADDITION THERE IS ONE                10006000 T  0003!083!0
                            EXTRA WORD FOR EACH LEFT PARENTHESIS WITH NO REPEAT PART.               10007000 T  0003!083!0
                            THIS IS AN IMPLIED STROKE TO CONTROL END OF LINE CONDI-                 10008000 T  0003!083!0
                            TIONS.  THE WORD IS BROKEN UP INTO NINE FIELDS;                         10009000 T  0003!083!0
```

```
      S = [1:1],                                                          10010000 T  00031083010
      REPEAT = [38:10],                                                   10011000 T  00031083010
      SKIP = [32:6],                                                      10012000 T  00031083010
      CODE = [2:4],                                                       10013000 T  00031083010
      W = [6:6],                                                          10014000 T  00031083010
      W1 = [28:4], W2 = [24:4], D1 = [20:4], D2 = [16:4].                 10015000 T  00031083010
S IS A DISTINGUISHER BETWEEN EDITING PHRASES AND OTHER                    10016000 T  00031083010
TYPE WORDS.  CODE IS THE INTERNAL CODE TO DISTINGUISH                     10017000 T  00031083010
BETWEEN THE VARIOUS EDITING PHRASES OR BETWEEN THE OTHER                  10018000 T  00031083010
WORDS.  GIVEN S = 1 WE HAVE:                                              10019000 T  00031083010
 IF CODE = 0 THEN RIGHTPAREN.                                             10020000 T  00031083010
 IF CODE = 2 THEN STRING.                                                 10021000 T  00031083010
 IF CODE = 4 THEN LEFTPAREN.                                              10022000 T  00031083010
 IF CODE = 6 THEN STROKE.                                                 10023000 T  00031083010
 IF CODE = 8 THEN SCALE.                                                  10023100 T  00031083010
GIVEN S = 0 WE HAVE                                                       10024000 T  00031083010
 IF CODE = 0 THEN D.                                                      10025000 T  00031083010
 IF CODE=1 THEN T.                                                        10025010 T  00031083010
 IF CODE = 2 THEN X.                                                      10026000 T  00031083010
 IF CODE = 4 THEN A.                                                      10027000 T  00031083010
 IF CODE = 6 THEN I.                                                      10028000 T  00031083010
 IF CODE = 8 THEN F.                                                      10029000 T  00031083010
 IF CODE =10 THEN E.                                                      10030000 T  00031083010
 IF CODE = 11 THEN U.                                                     10030100 T  00031083010
 IF CODE =12 THEN O.                                                      10031000 T  00031083010
 IF CODE = 13 THEN V.                                                     10031100 T  00031083010
 IF CODE =14 THEN L.                                                      10032000 T  00031083010
 IF CODE = 15 THEN R.                                                     10032100 T  00031083010
W IS THE FIELD WIDTH.                                                     10033000 T  00031083010
FOR STRINGS [12:36] IS W CHARACTORS OF ALPHA, RIGHT                       10034000 T  00031083010
 ADJUSTED. THE REST OF THE FIELDS ARE MEANINGLESS.                        10035000 T  00031083010
 REPEAT IS THE REPEAT FIELD - FOR LEFTPARENS WITH NO                      10036000 T  00031083010
 REPEAT FIELD, REPEAT = 0.  FOR RIGHTPARENS, REPEAT TELLS                 10037000 T  00031083010
 HOW MANY WORDS BACK THE CORRESPONDING LEFTPAREN IS.                      10038000 T  00031083010
 IMPLIED STROKES ARE DISTINGUISHED FROM VISIBLE STROKES BY                10039000 T  00031083010
 A NON-ZERO REPEAT FIELDS.                                                10040000 T  00031083010
   THE DESCRIPTION OF W1,W2, D1, AND D2 APPLIES ONLY TO                   10041000 T  00031083010
 FORMATING TYPES.  FOR THE PURPOSES OF DESCRIPTION LET                    10042000 T  00031083010
 D BE THE DECIMAL PART.  W IS, OF COURSE, THE WIDTH.                      10043000 T  00031083010
   THEN FOR D, W1=W2=D1=D2=SKIP=0.                                        10044000 T  00031083010
FOR X, W = SKIP = WIDTH MOD 64 AND W1 = WIDTH DIV 64.                     10045000 T  00031083010
       W2 = D1 = D2 =0.                                                   10046000 T  00031083010
   FOR T, W=(WIDTH-1) MOD 64,  W1=(WIDTH-1) DIV 64, AND                   10046010 T  00031083010
       W2=D1=D2=0.                                                        10046020 T  00031083010
   FOR A, W1 = W, SKIP = 0 IF W < 6.  OTHERWISE                           10047000 T  00031083010
       W1 = 6, SKIP = W-6.  W2=D1=D2=0.                                   10048000 T  00031083010
   FOR I: SKIP = IF W > 16 THEN W-16 ELSE 0.                              10049000 T  00031083010
      IF W > 8 THEN W1 = 8, W2 = W-SKIP-8.                                10050000 T  00031083010
      IF W < 8 THEN W1 = W, W2 = 0.  ALWAYS D1=D2=0.                      10051000 T  00031083010
   FOR F  IF D < 8 THEN D1 = D, D2=0.                                     10052000 T  00031083010
          IF D > 8 THEN D1 = 8, D2=D-8.                                   10053000 T  00031083010
          IF D >16 THEN ERROR.                                            10054000 T  00031083010
          IF W-D-1 > 16 THEN SKIP = W-D-17, OTHERWISE                     10055000 T  00031083010
            SKIP=0.                                                       10056000 T  00031083010
          IF W-D-1 > 8 THEN W1=8, W2=W-D-1-SKIP-8.                        10057000 T  00031083010
          IF W-D-1 < 8 THEN W1=W-D-1,W2=0.                                10058000 T  00031083010
   FOR E  D1 AND D2 ARE CALCULATED AS IN F EXCEPT THAT WE                 10059000 T  00031083010
      D+1 FOR D.  SKIP = W-D-6. W1=W2=0.                                  10060000 T  00031083010
```

```
                        FOR 0,  W1=W2=D1=D2=SKIP=0,                                                    10061000 T  0003!083!0
                        FOR L,  W2=D1=D2=0,  IF W > 5 THEN W1=5 ELSE W1 = W,                           10062000 T  0003!083!0
                                SKIP = W-W1,                                                           10063000 T  0003!083!0
                        FOR U! SKIP = W1 = W2 = D1 = D2 = 0,                                           10063100 T  0003!083!0
                        FOR B! SEE U-PHRASE DESCRIPTION,                                               10063110 T  0003!083!0
                        FOR R! SEE ABOVE F-PHRASE DESCRIPTION,                                         10063200 T  0003!083!0
                        FOR V! SKIP = W1 = W2 = UNSET, D1,D2 AS IN ABOVE                               10063300 T  0003!083!0
                               F-PHRASE DESCRIPTION,                                                   10063400 T  0003!083!0
                      FORMATPHRASE USES RECURSION TO DO ANALYSIS OF SYNTAX.  THE                       10064000 T  0003!083!0
                      WORDS ARE GENERATED AND PLACED DIRECTLY INTO THE CODE                            10065000 T  0003!083!0
                      BUFFER.  FORMATPHRASE IS A BOOLEAN PROCEDURE WHICH REPORTS                       10066000 T  0003!083!0
                      IF IT NOTICES AN ERROR;                                                          10067000 T  0003!083!0
            PROCEDURE WHIPOUT(W); VALUE W; REAL W;                                                     10068000 T  0003!083!0
PRT(105A) = WHIPOUT
                      BEGIN                                                                            10069000 T  0003!083!0
                                                                                                       10070000 T  0003!083!0
                      MOVE(1,W,EDOC[F,[38:3],F,[41:7]]);                                               10071000 T  0003!083!0
                            IF DEBUGTOG                                                                10072000 T  0003!083!1
                              THEN BEGIN                                                               10073000 T  0003!083!1
                                DEBUGWORD(B?D(F),W,LIN);                                               10074000 T  0003!083!3
                                WRITELINE  END;                                                        10075000 T  0003!083!3
                          IF (F←F+1) > 1024 THEN FLAG(307);                                            10076000 T  0003!084!0
                                                                                                       10077000 T  0003!085!2
                                                                                                       10078000 T  0003!085!2
                                                                                                       10079000 T  0003!085!2
                                                                                                       10080000 T  0003!085!2
                                                                                                       10081000 T  0003!085!2
                          END WHIPOUT;                                                                 10082000 T  0003!085!2
```

```
            BOOLEAN PROCEDURE FORMATPHRASE;                                                    10083000 T  0003!085!2
                    BEGIN                                                                      10084000 T  0003!085!2
                         LABEL EL,EX,EXIT,L1,L2,L3;                                            10085000 T  0003!085!2
                                                                                    START OF SEGMENT **********  119
            PROCEDURE EMITFORMAT(S,CODE,REPEAT,SKIP,W,W1,W2,D1,D2);                            10086000 T  0119!000!0
PRT(1055) = EMITFORMAT
                      VALUE S,CODE,REPEAT,SKIP,W,W1,W2,D1,D2 ;                                 10087000 T  0119!000!0
                         REAL CODE,REPEAT,SKIP,W,W1,W2,D1,D2 ;                                 10088000 T  0119!000!0
                         BOOLEAN S;                                                            10089000 T  0119!000!0
                      BEGIN IF W > 63 THEN FLAG(163);                                          10090000 T  0119!000!0
                        W ← REPEAT & W       [ 6:42:6]                                       10091000 T  0119!000!0
                                   & SKIP    [32:42:6]                                       10092000 T  0119!000!2
                                   & W1      [28:44:4]                                       10093000 T  0119!000!3
                                   & W2      [24:44:4]                                       10094000 T  0119!000!4
                                   & D1      [20:44:4]                                       10095000 T  0119!000!5
                                   & D2      [16:44:4]                                       10096000 T  0119!000!6
                                   & CODE    [ 2:44:4]                                       10097000 T  0119!000!7
                                   & REAL(S) [ 1:47:1];                                      10098000 T  0119!000!8
                        WHIPOUT(W) END FMITFORMAT;                                           10099000 T  0119!000!9
                                                                                             10099000 T  0119!001!1
```

```
            STREAM PROCEDURE PACKALPHA(PLACE,LETTER,CTR);                                      10100000 T  0119!001!2
PRT(1056) = PACKALPHA
```

```
                        VALUE LETTER,CTR;                                                   10101000 T  0119:0012:0
                        BEGIN DI ← PLACE; DS ← LIT "B";                                     10102000 T  0119:0012:0
                              SI ← LOC CTR; SI ← SI+7; DS ← CHR;                            10103000 T  0119:0012:1
                              SI ← PLACE; SI ← SI+3; DS ← 5 CHR;                            10104000 T  0119:0013:3
                              SI ← LOC LETTER; SI ← SI+7; DS ← CHR END PACKALPHA;           10105000 T  0119:0014:0


                        INTEGER REPEAT,SKIP,W,W1,W2,D1,D2,CODE; BOOLEAN S;                  10106000 T  0119:0015:2

STACK(F+3) = REPEAT
STACK(F+4) = SKIP
STACK(F+5) = W
STACK(F+6) = W1
STACK(F+7) = W2
STACK(F+10) = D1
STACK(F+11) = D2
STACK(F+12) = CODE
STACK(F+13) = S

                        DEFINE RRIGHT  = 0#,                                                10107000 T  0119:0015:2
                               RLEFT   = 4#,                                                10108000 T  0119:0015:2
                               RSTROKE = 6#;                                                10109000 T  0119:0015:2
                        DEFINE RSCALE = 8 #,  RU = 11 #,  RV = 13 #,  RR = 15 # ;           10109500 T  0119:0015:2
                        DEFINE RD =  0#, RX =  2#, RA =  4#, RI =  6#,                      10110000 T  0119:0015:2
                               RT=1 #,                                                      10110010 T  0119:0015:2
                               RF =  8#, RE = 10#, RO = 12#, RL = 14#;                      10111000 T  0119:0015:2
                    IF ELCLASS < 0 THEN BEGIN REPEAT ← -ELCLASS;NEXTENT;                    10112000 T  0119:0015:2
                         IF ELCLASS="," OR ELCLASS=")" THEN GO EX END                       10112100 T  0119:0018:1
                        ELSE BEGIN REPEAT:=REAL(ELCLASS#"<");                               10113000 T  0119:0021:2
                             IF ELCLASS="*" THEN BEGIN REPEAT,[12:1]←1;                     10113100 T  0119:0023:2
                                                       NEXTENT;                             10113200 T  0119:0026:0
                        END                          END;                                   10113300 T  0119:0026:1
                    IF ELCLASS="(" OR ELCLASS="<"                                           10114000 T  0119:0026:1
                      THEN BEGIN                                                            10115000 T  0119:0027:3
                        SKIP ← F;                                                           10116000 T  0119:0028:1
                        EMITFORMAT(TRUE,RLEFT,REPEAT,1,0,0,0,0,0);                          10117000 T  0119:0029:3
                        DO BEGIN NEXTENT;                                                   10118000 T  0119:0032:1
EL:                       IF FORMATPHRASE THEN GO TO EX END                                 10119000 T  0119:0033:3
                        UNTIL ELCLASS # ",";                                                10120000 T  0119:0035:2
                        WHILE ELCLASS = "/"                                                 10121000 T  0119:0036:1
                          DO BEGIN EMITFORMAT(TRUE,RSTROKE,0,1,0,0,0,0,0);                  10122000 T  0119:0037:2
                             NEXTENT END;                                                   10123000 T  0119:0041:2
                        IF ELCLASS  #  ")" AND ELCLASS  #  ">"                             10124000 T  0119:0042:0
                           THEN GO TO EL;                                                   10124100 T  0119:0043:2
                        IF LASTELCLASS = "," THEN GO TO EX;                                 10124200 T  0119:0044:1
                        IF REPEAT = 0 THEN                                                  10125000 T  0119:0046:0
                             EMITFORMAT(TRUE,RSTROKE,1,0,0,0,0,0,0);                        10126000 T  0119:0046:1
                        REPEAT←F-SKIP; F←SKIP;                                              10127000 T  0119:0050:0
                        WHIPOUT(EDOC[F,[38:3],F,[41:7]]&REPEAT[28:38:10]);                  10127100 T  0119:0052:0
                        F←SKIP+REPEAT; S←TRUE; CODE←RRIGHT END                              10127200 T  0119:0056:0
                    ELSE IF ELCLASS = "0"                                                   10128000 T  0119:0059:2
                       THEN BEGIN CODE←RO; W←8 END                                          10129000 T  0119:0059:3
                    ELSE IF ELCLASS = "D"                                                   10130000 T  0119:0062:0
                       THEN BEGIN CODE←RD; W←8 END                                          10131000 T  0119:0063:2
                    ELSE IF ELCLASS = "," THEN GO TO L2                                     10132000 T  0119:0065:3
                    ELSE IF ELCLASS = "/" THEN GO TO EXIT                                   10133000 T  0119:0066:1
                    ELSE IF ELCLASS=")" OR ELCLASS=">" THEN                                 10134000 T  0119:0068:0
                       IF LASTELCLASS="," THEN GO EX ELSE GO EXIT                           10134100 T  0119:0070:0
```

```
ELSE IF ELCLASS = "S" THEN
      BEGIN
           NEXTENT;
           W ← IF ELCLASS = "=" THEN 1 ELSE 0;
           IF ELCLASS="+" OR ELCLASS="-" THEN NEXTENT;
           IF ELCLASS="*" THEN REPEAT.[12:1]←1 ELSE
           IF ELCLASS > 0 THEN BEGIN ERR(136);
                                     GO TO EXIT
                               END
                          ELSE REPEAT ← - ELCLASS;
           EMITFORMAT(TRUE,RSCALE,REPEAT,0,W,0,0,0,0);
           GO TO L2
      END
 ELSE IF ELCLASS = "'"
    THEN BEGIN
       IF REPEAT ≠ 1 THEN FLAG(136);
       CODE ← 100;
       DO BEGIN
         SKIP ← 1;
         DO BEGIN RESULT ← 5; COUNT ← 0; SCANNER;
           IF ELCLASS ← ACCUM[1].[18:6] = CODE
             THEN BEGIN
               IF SKIP ≠ 1 THEN WHIPOUT(W);
               GO TO L2 END;
           CODE ← "'";
           PACKALPHA(W,ELCLASS,SKIP);
         END UNTIL SKIP ← SKIP+1 = 7;
         WHIPOUT(W)
       END UNTIL FALSE END
ELSE BEGIN CODE←ELCLASS;
IF CODE = "U" OR CODE = "B" THEN
      BEGIN %%% ALL OF COMPILER CODE TO HANDLE U-PHRASE.
      NEXTENT ;
      SKIP ← 0 ;
      IF ELCLASS = "*" OR ELCLASS ≤ 0 THEN
           BEGIN %%% PHRASE IS AT LEAST UW OR U*.
           IF ELCLASS = "*" THEN REPEAT.[13:1] ← 1
           ELSE W ← -ELCLASS ;
           NEXTENT ;
           IF ELCLASS = "." THEN
                BEGIN %%% PHRASE IS AT LEAST UW. OR U*..
                NEXTENT ;
                IF ELCLASS = "*" OR ELCLASS ≤ 0 THEN
                     BEGIN %%% PHRASE IS UW<OR>*.D<OR>*.
                     IF ELCLASS = "*" THEN REPEAT.[14:1]←1
                     ELSE SKIP ← -ELCLASS ;
                     NEXTENT ;
                     END
                ELSE GO TO EX
                END
           END
      ELSE W ←-63 ; %%% PHRASE IS U.
      EMITFORMAT(FALSE,RU,REPEAT,SKIP,W,REAL(CODE="B"),
                 REAL(W<0),0,0) ;
      GO TO EXIT ;
      END OF U PHRASE HANDLER ;
IF GETINT THEN BEGIN W←11;  REPEAT.[13:1]←1 END
```

```
10134500 T  0119:0072:1
10134510 T  0119:0073:3
10134520 T  0119:0074:0
10134530 T  0119:0074:1
10134540 T  0119:0077:3
10134545 T  0119:0080:0
10134550 T  0119:0083:2
10134560 T  0119:0085:3
10134570 T  0119:0086:0
10134580 T  0119:0086:0
10134590 T  0119:0087:3
10134600 T  0119:0090:1
10134610 T  0119:0091:2
10135000 T  0119:0091:2
10136000 T  0119:0092:0
10136500 T  0119:0093:2
10137000 T  0119:0095:2
10138000 T  0119:0095:3
10139000 T  0119:0096:0
10140000 T  0119:0096:1
10141000 T  0119:0099:2
10142000 T  0119:0100:1
10143000 T  0119:0101:3
10144000 T  0119:0103:3
10145000 T  0119:0104:0
10146000 T  0119:0105:2
10147000 T  0119:0106:1
10148000 T  0119:0108:1
10149000 T  0119:0109:2
10150000 T  0119:0110:1
10150100 T  0119:0111:3
10150110 T  0119:0113:3
10150120 T  0119:0114:0
10150125 T  0119:0114:1
10150130 T  0119:0115:2
10150135 T  0119:0117:2
10150140 T  0119:0117:3
10150145 T  0119:0119:2
10150150 T  0119:0122:0
10150155 T  0119:0122:1
10150160 T  0119:0123:2
10150165 T  0119:0123:3
10150170 T  0119:0124:0
10150175 T  0119:0126:0
10150185 T  0119:0126:1
10150190 T  0119:0128:0
10150195 T  0119:0131:2
10150200 T  0119:0131:3
10150205 T  0119:0131:3
10150210 T  0119:0131:3
10150215 T  0119:0131:3
10150220 T  0119:0131:3
10150225 T  0119:0133:2
10150230 T  0119:0135:2
10150260 T  0119:0137:2
10150270 T  0119:0137:3
10150280 T  0119:0137:3
```

```
 ELSE ELCLASS := -(W := ELCLASS);                                                10150290 T  0119:014:2
    IF CODE = "I"                                                                10151000 T  0119:014:2
     THEN BEGIN                                                                  10152000 T  0119:014:2
      SKIP ← DIVIDE(W,W1,W2); CODE ← RI END                                       10153000 T  0119:014:0
    ELSE IF CODE = "F"                                                           10154000 T  0119:014:1
     THEN BEGIN CODE ← RF; GO TO L1 END                                          10155000 T  0119:014:3
ELSE IF CODE = "R" THEN BEGIN CODE ← RR; GO TO L1 END                            10155500 T  0119:014:3
    ELSE IF CODE = "E"                                                           10156000 T  0119:015:1
     THEN BEGIN CODE ← RE; D1←1;                                                 10157000 T  0119:015:3
L1:   NEXTENT;                                                                   10158000 T  0119:015:0
      IF ELCLASS≠"." THEN GO EX;                                                 10159000 T  0119:015:1
      IF GETINT THEN BEGIN ELCLASS←3; REPEAT.[14:1]←1 END;                       10159100 T  0119:015:3
      IF DIVIDE(ELCLASS+D1,D1,D2) > 0 THEN GO TO EX;                             10160000 T  0119:016:2
     IF CODE = RF OR CODE = RR THEN                                              10161000 T  0119:016:0
        SKIP ← DIVIDE(W-ELCLASS-1,W1,W2)                                         10161500 T  0119:016:3
        ELSE IF SKIP ← W-ELCLASS-6 < 0 THEN GO TO EX END                         10162000 T  0119:016:0
    ELSE IF CODE = "X"                                                           10163000 T  0119:017:0
     THEN BEGIN CODE ← RX; W1 ← W.[38:4];                                        10164000 T  0119:017:2
      SKIP ← W ← W.[42:6] END                                                    10165000 T  0119:017:0
      ELSE IF CODE="T" THEN IF W←ABS(W)-1<0 THEN FLAG(136)                       10165500 T  0119:017:3
      ELSE BEGIN CODE←RT; W1←W.[38:4]; W←W.[42:6] END                            10165505 T  0119:018:0
    ELSE IF CODE = "A"                                                           10166000 T  0119:018:1
     THEN BEGIN CODE ← RA; W1 ←6; GO TO L3 END                                   10167000 T  0119:018:2
    ELSE IF CODE="V" THEN                                                        10167100 T  0119:019:0
     BEGIN CODE ← RV ;                                                           10167200 T  0119:019:3
           COUNT←ACCUM[1]←0;                                                     10167300 T  0119:019:1
           IF EXAMIN(NCR)=" " THEN                                               10167400 T  0119:019:1
              BEGIN RESULT←7; SCANNER END;                                       10167500 T  0119:019:0
           IF EXAMIN(NCR)="," THEN                                               10167600 T  0119:019:0
            BEGIN NEXTENT;                                                       10167700 T  0119:019:3
                  IF GETINT THEN REPEAT.[14:1]←1 ELSE                            10167800 T  0119:020:1
                  GT1←DIVIDE(ELCLASS,D1,D2);                                     10167900 T  0119:020:3
              ELCLASS :=-ELCLASS;                                                10167910 T  0119:020:3
         END;   END ELSE IF CODE="L"                                             10168000 T  0119:020:1
           THEN BEGIN CODE ← RL; W1 ← 5;                                         10169000 T  0119:020:3
L3:          IF W<W1 THEN W1←W; SKIP ← W-W1 END ELSE GO EX END;                  10170000 T  0119:021:0
       EMITFORMAT(S,CODE,REPEAT,SKIP,W,W1,W2,D1,D2);                             10171000 T  0119:021:2
L2:    NEXTENT; GO TO EXIT;                                                      10172000 T  0119:021:0
EX:    FORMATPHRASE ← TRUE; ERR(136);                                            10173000 T  0119:021:0
EXIT:  END FORMATPHRASE;                                                         10174000 T  0119:021:3
                                                                  119 IS  225 LONG, NEXT SEG    3


COMMENT  GETINT DOES A CALL ON NEXTENT AND CHECKS TO SEE IF AN INTEGER           10175000 T  0003:085:2
         WAS THE RESULT: IF NOT ERROR - OTHERWISE MAKE SIGN PLUS;                10176000 T  0003:085:2
BOOLEAN PROCEDURE GETINT;                                                        10177000 T  0003:085:2
         BEGIN NEXTENT; IF ELCLASS ← - ELCLASS < 0 THEN                          10178000 T  0003:085:2
              IF ELCLASS=-("**") THEN GETINT←TRUE ELSE                           10178100 T  0003:085:2
              BEGIN FLAG(137); ELCLASS ← 0 END                                   10179000 T  0003:085:3
         END GETINT;                                                             10180000 T  0003:085:3


COMMENT  DIVIDE PARTIONS THE PARAMETER NUMBER INTO THREE PARTS.  THE              10181000 T  0003:086:0
         RESULT IS PASSED BACK THROUGH P1,P2, AND THE FUNCTION                   10182000 T  0003:086:0
```

```
                IDENTIFIER.  SEE CODE FOR DETAILS;                                  10183000 T  0003:086210
INTEGER PROCEDURE DIVIDE(NUMBER,P1,P2);                                             10184000 T  0003:086210
                VALUE NUMBER; INTEGER P1,P2,NUMBER;                                 10185000 T  0003:086210
                BEGIN                                                               10186000 T  0003:086210
                  IF NUMBER < 0 THEN BEGIN FLAG(138); ERRORTOG←TRUE;                10187000 T  0003:086210
                     NUMBER ← 0 END;                                                10188000 T  0003:086513
                     P1 ← IF NUMBER < 8 THEN NUMBER ELSE 8;                         10189000 T  0003:086611
                     NUMBER ← NUMBER-P1;                                            10190000 T  0003:086913
                     P2 ← IF NUMBER < 8 THEN NUMBER ELSE 8;                         10191000 T  0003:087011
                     DIVIDE ← NUMBER-P2 END DIVIDE;                                 10192000 T  0003:087313


COMMENT  DEFINEGEN  PACKS THE ALPHA FOR A DEFINE INTO INFO.  DEFINEGEN              10193000 T  0003:087712
                PRINCIPALLY ELIMINATES BLANKS AND COMMENTS.  PUTOGETHER              10194000 T  0003:087712
                ACTUALLY DOES THE MECHANICAL WORK OF PROCESSING THE                 10195000 T  0003:087712
                CHARACTORS FROM ACCUM TO INFO.  THE FINAL TRANSFER IS DONE          10196000 T  0003:087712
                IN PACKINFO.  THE OTHER FUNCTION SERVED BY DEFINEGEN IS             10197000 T  0003:087712
                THAT OF COUNTING DEFINES AND CROSSHATCHES IN ORDER TO               10198000 T  0003:087712
                ALLOW NESTED DEFINES.                                               10199000 T  0003:087712
                     A WORD ON THE OVERALL PLAN OF ATTACK ON HANDLEING               10200000 T  0003:087712
                DEFINES:                                                            10201000 T  0003:087712
                     THERE ARE FOUR PLACES THAT THERE IS CODE WRITTEN               10202000 T  0003:087712
                EXPLICITLY FOR THE DEFINE.                                          10203000 T  0003:087712
                     FIRST IS DEFINEGEN WHICH LOADS THE ALPHA INTO INFO.            10204000 T  0003:087712
                     SECOND IS IN THE TABLE ROUTINE AFTER A DEFINED ID IS           10205000 T  0003:087712
                NOTICED.  HERE A SETUP IS PERFORMED SO THAT THE SCANNER             10206000 T  0003:087712
                WILL SCAN INFO.  SINCE INFO (UNLIKE I/O BUFFERS) IS NOT             10207000 T  0003:087712
                A SAVE ARRAY.  WE CAN NOT DIRECTLY SCAN INFO.  INSTEAD WE           10208000 T  0003:087712
                FOOL READACARD SO THAT THE ALPHA IS FETCHED FROM INFO AND           10209000 T  0003:087712
                PLACED INTO A SMALL SAVE ARRAY (DEFINEARRAY) INSTEAD OF             10210000 T  0003:087712
                BEING FETCHED FROM AN I/O DEVICE.  NATURALLY WE MUST HAVE           10211000 T  0003:087712
                NESTING WHICH IS OBTAINED BY USING DEFINEARRAY AS A SMALL           10212000 T  0003:087712
                STACK.  THE QUANTITIES SAVED ARE LCR,NCR, AND LASTUSED.             10213000 T  0003:087712
                LASTUSED DOUBLES AS A DEVICE FOR DIRECTING THE FLOW OF              10214000 T  0003:087712
                INFORMATION FROM I/O GEAR AND FROM INFO DURING ANALYSIS OF          10215000 T  0003:087712
                DEFINES.  THIS STACKING IS DONE HERE BY THE TABLE ROUTINE.          10216000 T  0003:087712
                     THIRD IS READACARD WHICH HAS NOW BEEN FAIRLY WELL              10217000 T  0003:087712
                DESCRIBED.  THE ONLY ADDITIONAL COMMENT NECESSARY IS THAT           10218000 T  0003:087712
                LASTUSED SERVES AS AN INDEX TO FETCH NEXT WORD OF ALPHA             10219000 T  0003:087712
                FROM INFO. READACARD FETCHES ONLY ONE WORD AT A TIME.               10220000 T  0003:087712
                     FOURTH IS ALSO IN THE TABLE ROUTINE.  IT IS AT THE             10221000 T  0003:087712
                PLACE THAT A CROSSHATCH IS NOTICED.  IT CAUSES AN UNSETUP           10222000 T  0003:087712
                SO THAT THE SCANNER WILL STOP SCANNING THAT PART OF INFO            10223000 T  0003:087712
                AND RESUME ITS PREVIOUS TASKS.  A DISTINCTION IS MADE               10224000 T  0003:087712
                BETWEEN CROSSHATCHES SCANNED AFTER A DEFINE DECLARATION             10225000 T  0003:087712
                AND THOSE SCANNED DURING THE RECALL PROCESS.  THE LATER             10226000 T  0003:087712
                ONLY CAUSES AN UNSETUP;                                             10227000 T  0003:087712
          PROCEDURE DEFINEGEN(MACRO,J); VALUE MACRO,J; BOOLEAN MACRO; REAL J;       10228000 T  0003:087712
PRT(1057) = DEFINEGEN
                BEGIN                                                               10229000 T  0003:087712
                   OWN INTEGER CHARCOUNT, REMCOUNT;                                 10230000 T  0003:087712
                                                                  START OF SEGMENT **********  120
PRT(1060) = CHARCOUNT
PRT(1061) = REMCOUNT
                COMMENT CHARCOUNT CONTAINS NUMBER OFCHARACTORS OF THE DEFINE THAT WE  10231000 T  0120:000010
```

```
                        HAVE PUT INTO INFO,  REMCOUNT CONTAINS NUMBER OF CHARACT-          10232000 T  0120:000:0
                        ORS REMAINING IN THIS ROW OF INFO;                                 10233000 T  0120:000:0
              PROCEDURE PUTOGETHER(CHAR); REAL CHAR;                                       10234000 T  0120:000:0
PRT(1062) = PUTOGETHER
                    BEGIN                                                                  10235000 T  0120:000:0
              STREAM PROCEDURE PACKINFO(INFO,ISKIP,COUNT,ASKIP,ACCUM);                     10236000 T  0120:000:0
                                                                          START OF SEGMENT **********  121
PRT(1063) = PACKINFO
                        VALUE ISKIP,COUNT,ASKIP;                                           10237000 T  0121:000:0
                        BEGIN DI ← INFO; DI ← DI+ISKIP;                                    10238000 T  0121:000:0
                              SI ← ACCUM;SI ← SI+ASKIP; SI ← SI+3;                         10239000 T  0121:000:1
                              DS ← COUNT CHR END PACKINFO;                                 10240000 T  0121:000:3



                       INTEGER COUNT,SKIPCOUNT;                                          10241000 T  0121:000:1
STACK(F+2) = COUNT
STACK(F+3) = SKIPCOUNT
                        IF (COUNT ← CHAR.[12:6]) + CHARCOUNT > 2047                        10242000 T  0121:000:1
                          THEN BEGIN FLAG(142);  TB1← TRUE  END                            10243000 T  0121:000:2
                          ELSE BEGIN                                                       10244000 T  0121:000:3
                             IF COUNT > REMCOUNT                                           10245000 T  0121:000:2
                               THEN BEGIN                                                  10246000 T  0121:000:2
                                  SKIPCOUNT ← COUNT-(COUNT←REMCOUNT);                      10247000 T  0121:001:1
                                  REMCOUNT ← 2048 END                                      10248000 T  0121:001:1
                               ELSE REMCOUNT ← REMCOUNT-COUNT;                             10249000 T  0121:001:3
                             GT1 ← CHARCOUNT DIV 8 + NEXTTEXT;                             10250000 T  0121:001:1
                             PACKINFO(TEXT[GT1.LINKR,GT1.LINKC], CHARCOUNT,[45:3],         10251000 T  0121:001:1
                             COUNT,0,CHAR);                                                10252000 T  0121:002:2
                             IF SKIPCOUNT ≠ 0 THEN                                         10253000 T  0121:002:3
                                PACKINFO(TEXT[NEXTTEXT,LINKR+1,0],0,SKIPCOUNT,             10254000 T  0121:002:0
                                  COUNT,CHAR);                                             10255000 T  0121:002:3
                             CHARCOUNT ← CHARCOUNT+SKIPCOUNT+COUNT END                     10256000 T  0121:002:1
                    END PUTOGETHER;                                                        10257000 T  0121:003:2
                                                                          121 IS   34 LONG, NEXT SEG  120



                       INTEGER LASTRESULT;                                               10258000 T  0120:000:0
STACK(F+2) = LASTRESULT
                         REAL K,N,ELCLASS;                                               10258100 T  0120:000:0
STACK(F+3) = K
STACK(F+4) = N
STACK(F+5) = ELCLASS
                         DEFINE I=NXTELBT#;                                              10258200 T  0120:000:0
                         LABEL FINAL,PACKIN;                                             10258300 T  0120:000:0
                         LABEL BACK,SKSC,EXIT;                                           10259000 T  0120:000:0
                         REAL DINFO;                                                     10259200 T  0120:000:0
STACK(F+6) = DINFO
                         BOOLEAN TSSTREAMTOG;                  % 1289                     10259400 T  0120:000:0
STACK(F+7) = TSSTREAMTOG
                         DINFO ← J.[18:15];                                              10259600 T  0120:000:0
                         J ← J.[33:15];                                                  10259700 T  0120:000:2
                         TB1← FALSE;                                                     10260000 T  0120:000:1
                         TSSTREAMTOG← STREAMTOG;               % 1289                     10260050 T  0120:000:2
                 STREAMTOG←TRUE;                                                         10260100 T  0120:000:0
```

```
                    CHARCOUNT ← 0;                                                  10261000 T  0120↑000411
                    DEFINECTR ← 1; LASTRESULT ← 2;                                  10262000 T  0120↑000513
                    REMCOUNT ← (256-NEXTTEXT.LINKC)×8;                              10263000 T  0120↑000712
                    K←0;                                                            10263200 T  0120↑000913
BACK:               STOPDEFINE←TRUE;                                                10263300 T  0120↑001011
                    ELCLASS←TABLE(NXTELBT);                                         10263400 T  0120↑001113
SKSC:               NXTELBT←NXTELBT-1;                                              10263500 T  0120↑001312
                    IF MACRO THEN                                                   10263600 T  0120↑001410
                    BEGIN IF ELCLASS=COMMA THEN                                     10263700 T  0120↑001411
                          IF K=0 THEN                                               10263800 T  0120↑001513
FINAL:                    BEGIN PUTOGETHER("1#0000"); GO TO EXIT END                10263900 T  0120↑001712
                          ELSE GO PACKIN;                                           10264000 T  0120↑002112
                          IF ELCLASS=LEFTPAREN OR ELCLASS=LFTBRKET THEN             10264100 T  0120↑002112
                          BEGIN K←K+1; GO TO PACKIN END;                            10264200 T  0120↑002211
                          IF ELCLASS=RTPAREN OR ELCLASS=RTBRKET THEN                10264300 T  0120↑002512
                          IF K←K-1<0 THEN GO FINAL ELSE GO PACKIN;                  10264400 T  0120↑002611
                          IF ELCLASS=SEMICOLON THEN                                 10264410 T  0120↑003010
                              BEGIN FLAG(142); GO TO FINAL END ELSE GO PACKIN       10264420 T  0120↑003011
                    END;                                                            10264500 T  0120↑003211
                          IF RESULT = 1 THEN IF J ≠ 0 THEN                          10264600 T  0120↑003211
                          FOR N ← 1 STEP 1 UNTIL J DO                               10264650 T  0120↑003411
                          BEGIN                                                     10264700 T  0120↑003610
                              IF EQUAL(ACCUM[1],[12:6]+3, ACCUM[1],                 10264750 T  0120↑003610
                                  DEFINFO[(N-1)×10]) THEN                           10264760 T  0120↑003811
                              BEGIN                                                 10264800 T  0120↑004011
                                  DEFINEPARAM(DINFO+1, N);                          10264810 T  0120↑004112
                                  GO PACKIN;                                        10264820 T  0120↑004211
                              END;                                                  10264830 T  0120↑004312
                          END;                                                      10264900 T  0120↑004312
PACKIN:                                                                             10264910 T  0120↑004512
                          IF RESULT = 4                                             10265000 T  0120↑004610
                            THEN BEGIN                                              10266000 T  0120↑004610
   COMMENT  INSERT " MARKS = 2130706432 IS DECIMAL FOR 1"0000;                      10267000 T  0120↑004712
                              PUTOGETHER(2130706432);                               10268000 T  0120↑004712
                              PUTOGETHER(ACCUM[1]);                                 10269000 T  0120↑004810
                              PUTOGETHER(2130706432) END                            10270000 T  0120↑004913
                          ELSE BEGIN                                                10271000 T  0120↑005011
                            IF BOOLEAN(RESULT) AND BOOLEAN(LASTRESULT)              10272000 T  0120↑005210
                              THEN PUTOGETHER("1 0000"); COMMENT INSERT BLANK;      10273000 T  0120↑005210
                            PUTOGETHER(ACCUM[1]) END;                               10274000 T  0120↑005410
                          IF TB1 THEN GO TO EXIT;                                   10275000 T  0120↑005513
                          LASTRESULT ← RESULT;                                      10276000 T  0120↑005611
                          IF MACRO THEN GO BACK;                                    10276500 T  0120↑005712
                          IF ELCLASS=DECLARATORS AND ELBAT[I].ADDRESS = DEFINEV     10277000 T  0120↑005810
                            THEN BEGIN DEFINECTR ← DEFINECTR+1; GO BACK END;        10278000 T  0120↑006010
                          IF ELCLASS ≠ CROSSHATCH THEN GO BACK;                     10279000 T  0120↑006410
                          IF DEFINECTR ≠ 1                                          10280000 T  0120↑006512
                            THEN BEGIN STOPDEFINE ← TRUE;                           10281000 T  0120↑006513
                              IF ELCLASS←TABLE(I)≠COMMA THEN                        10282000 T  0120↑006712
                              DEFINECTR←DEFINECTR-1; GO SKSC END;                   10283000 T  0120↑006912
EXIT:               DEFINECTR := 0; STREAMTOG←TSSTREAMTOG;          % 1289        10284000 T  0120↑007112
                    NEXTTEXT ← (CHARCOUNT+7) DIV 8 + NEXTTEXT;                      10285000 T  0120↑007313
                END  DEFINEGEN;                                                     10286000 T  0120↑007610
                                                                        120 IS   80 LONG, NEXT SEG    3
```

```
                    COMMENT  LISTELEMENT IS RESPONSIBLE FOR THE GENERATION OF CODE FOR LIST      10287000 T  0003:087:2
                             ELEMENTS;                                                          10288000 T  0003:087:2
                    PROCEDURE LISTELEMENT;                                                      10289000 T  0003:087:2
                            BEGIN                                                               10290000 T  0003:087:2
                                REAL T1,T2,T3;                                                  10291000 T  0003:087:2
                                                                                  START OF SEGMENT **********  122
STACK(F+2) = T1
STACK(F+3) = T2
STACK(F+4) = T3
                                LABEL BOOFINISH,STORE,LRTS;                                     10292000 T  0122:000:0
                                DIALA ← DIALB ← 0;                                              10293000 T  0122:000:0
                                IF ELCLASS = FORV THEN FORSTMT  COMMENT FORCLAUSE;              10294000 T  0122:000:2
                                ELSE IF ELCLASS = LFTBRKET                                      10295000 T  0122:000:1
                                 THEN BEGIN   COMMENT  GROUP OF LIST ELEMENTS;                  10296000 T  0122:000:3
                                     DO BEGIN STEPIT; LISTELEMENT END UNTIL ELCLASS#COMMA;      10297000 T  0122:000:4
                                     IF ELCLASS = RTBRKET THEN STEPIT ELSE ERR(158) END         10298000 T  0122:000:6
                                ELSE BEGIN COMMENT THE MEAT OF THE MATTER:                      10299000 T  0122:001:0
                                                  VARIABLES AND EXPRESSIONS;                    10300000 T  0122:001:0
                                  L ← (T1←L)+1;  COMMENT SAVE L FOR LATER FIXUP;                10301000 T  0122:001:1
                                  EMITPAIR(LSTRTN,STD);  COMMENT  PREPARE LSTRTN FOR            10302000 T  0122:001:2
                                                                  NEXT TIME AROUND;             10303000 T  0122:001:3
                                  IF(GT1 ← TABLE(I+1) = COMMA                                   10304000 T  0122:001:3
                                       OR GT1 = RTPAREN                                         10305000 T  0122:001:5
                                       OR GT1 = RTBRKET)                                        10306000 T  0122:001:6
                                    AND ELCLASS ≥ BOOID AND ELCLASS ≤ INTID                     10307000 T  0122:001:7
                                   THEN BEGIN COMMENT SIMPLE VARIABLES;                         10308000 T  0122:001:9
                                        CHECKER(ELBAT[I]);                                      10308100 T  0122:002:0
                                       EMITN(ELBAT[I].ADDRESS); STEPIT END                      10309000 T  0122:002:1
                                  ELSE BEGIN IF ELCLASS ≥ BOOARRAYID                            10310000 T  0122:002:3
                                             AND ELCLASS ≤ INTARRAYID                           10311000 T  0122:002:4
                                     THEN BEGIN COMMENT   IS EITHER A SUBCRIPTED VARIABLE       10312000 T  0122:002:1
                                                  OR THE BEGINNING OF AN EXPRESSION.  THIS      10313000 T  0122:002:6
                                                  SITUATION IS VERY SIMILAR TO THAT IN          10314000 T  0122:002:6
                                                  ACTUALPARAPART (SEE COMMENTS THERE FOR         10315000 T  0122:002:6
                                                  FURTHER DETAILS);                             10316000 T  0122:002:6
                                       T2 ← FL; T3 ← ELCLASS; VARIABLE(T2);                     10317000 T  0122:002:6
                                  IF TABLE(I-2)=FACTOP AND TABLE(I-1)=RTBRKET THEN ERR(157);    10318000 T  0122:002:8
                                       IF ELCLASS = COMMA OR                                    10319000 T  0122:003:2
                                          ELCLASS = RTPAREN OR                                  10320000 T  0122:003:4
                                          ELCLASS = RTBRKET THEN                                10321000 T  0122:003:5
                                           IF T2 = 0 THEN GO TO STORE ELSE GO TO LRTS;           10322000 T  0122:003:6
                                       IF T3 = BOOARRAYID THEN GO TO BOOFINISH;                 10323000 T  0122:003:8
                                       SIMPARITH;                                               10324000 T  0122:003:9
                                       IF ELCLASS = RELOP THEN BEGIN RELATION;                  10325000 T  0122:004:0
BOOFINISH:                               SIMPBOO END END                                        10326000 T  0122:004:3
                                  ELSE IF EXPRSS = DTYPE THEN ERR(156);                         10327000 T  0122:004:1
STORE:                            EMITPAIR(JUNK,STD); EMITN(JUNK) END;                          10328000 T  0122:004:5
LRTS:                             EMITO(RTS); CONSTANTCLEAN;                                    10329000 T  0122:004:7
                                  T2 ← L; L ← T1; EMITNUM(T2-LSTR); L←T2 END END LSTELMT;       10330000 T  0122:004:9
                                                                  122 IS   55 LONG, NEXT SEG    3


                    COMMENT   LISTGEN COMPILES ALL THE CODE FOR A LIST.  LISTGEN CALLS          10331000 T  0003:087:2
                              LISTELEMENT WHICH IS RESPONSIBLE FOR EACH INDIVIDUAL              10332000 T  0003:087:2
                              LIST ELEMENT.  LIST ELEMENT ALSO TAKES CARE TO GENERATE           10333000 T  0003:087:2
                              CODE WHICH UPDATES LSTRTN AFTER EACH CALL ON THE LIST.            10334000 T  0003:087:2
```

```
                        LISTGEN GENERATES THE CHANGING OF LSTRTN TO -1, THE END          10335000 T  0003!087712
                        FLAG FOR A LIST, THE CODE TO JUMP AROUND THE LIST,               10336000 T  0003!087712
                        THE INITIAL JUMP OF THE LIST, THE OBTAINING OF A PRT CELL        10337000 T  0003!087712
                        FOR THE LIST, THE OBTAINING OF AN ACCIDENTAL PROGRAM             10338000 T  0003!087712
                        DESCRIPTOR, THE STUFFING OF F INTO THIS DESCRIPTOR.              10339000 T  0003!087712
                        LISTGEN EXPECTS I TO POINT AT FIRST LIST ELEMENT AND             10340000 T  0003!087712
                        LEAVES I POINTING AT FIRST ITEM BEYOND RIGHTPAREN.  THE          10341000 T  0003!087712
                        VALUE RETURNED BY LISTGEN IS THE LOCATION OF THE                 10342000 T  0003!087712
                        ACCIDENTAL ENTRY DESCRIPTOR IN THE PRT;                          10343000 T  0003!087712
              REAL PROCEDURE LISTGEN;                                                    10344000 T  0003!087712
                   BEGIN                                                                 10345000 T  0003!087712
                        INTEGER JUMPLACE,LISTPLACE;                                      10346000 T  0003!087712
                                                                        START OF SEGMENT **********  123
STACK(F+3) = JUMPLACE
STACK(F+4) = LISTPLACE
                        JUMPLACE ← BAE;                                                  10347000 T  0123!000010
                        LISTGEN ← LISTPLACE ← PROGDESCBLDR(0,L,0);                       10348000 T  0123!000112
                 COMMENT BUILDS ACCIDENTAL ENTRY FOR LIST;                               10349000 T  0123!000312
                        EMITV(LSTRTN); EMITO(BFW); LSTR ← L;                             10350000 T  0123!000312
                 COMMENT INITIAL JUMP OF A LIST;                                         10351000 T  0123!000513
                        LISTMODE ← TRUE;                                                 10352000 T  0123!000513
                 COMMENT CAUSES FORSTMT TO RECOGNIZE THAT WE ARE COMPILING LISTS;        10353000 T  0123!000610
                        I←I-1;                                                           10354000 T  0123!000610
                        DO BEGIN                                                         10355000 T  0123!000713
                             STEPIT;                                                     10356000 T  0123!000810
                             LISTELEMENT                                                 10357000 T  0123!000811
                        END UNTIL ELCLASS ≠ COMMA;                                       10358000 T  0123!000811
                        EMITL(1); EMITO(CHS);                                            10359000 T  0123!001010
                        EMITPAIR(LSTRTN,SND);                                            10360000 T  0123!001113
                        EMITO(RTS);                                                      10361000 T  0123!001211
                 COMMENT SET END FLAG OF -1;                                             10362000 T  0123!001313
                        CONSTANTCLEAN;                                                   10363000 T  0123!001313
                        DIALA ← DIALB ← 0;                                               10364000 T  0123!001410
                        LISTMODE ← FALSE;                                                10365000 T  0123!001512
                       ADJUST;                                                           10365100 T  0123!001610
                        EMITB(BFW,JUMPLACE,L);                                           10366000 T  0123!001611
                        STUFFF(LISTPLACE);                                               10367000 T  0123!001713
                        IF ELCLASS ≠ RTPAREN THEN ERR(104) ELSE STEPIT                   10368000 T  0123!001811
                   END  LISTGEN;                                                         10369000 T  0123!002112
                                                                    123 IS   24 LONG, NEXT SEG    3


              BOOLEAN PROCEDURE MERRIMAC;                                                10370000 T  0003!087712
PRT(1064) = MERRIMAC
                   BEGIN COMMENT THIS TIME THE MERRIMAC WILL HANDLE THE MONITOR,         10371000 T  0003!087712
                                  03 JULY 1963                                           10372000 T  0003!087712
                     THERE ARE SIX TYPES OF MONITOR LIST ELEMENTS, THEY ARE              10373000 T  0003!087712
                   LABELS, SWITCHES, SIMPLE VARIABLES, SUBSCRIPTED VARIABLES,            10374000 T  0003!087712
                   ARRAYS, AND FUNCTION DESIGNATORS.                                     10375000 T  0003!087712
                     WITH ONE EXCEPTION, THE MERRIMAC ROUTINES ONLY FUNCTION             10376000 T  0003!087712
                   IS TO SAVE INFORMATION SO THAT OTHER ROUTINES, SUCH AS THE            10377000 T  0003!087712
                   VARIABLE ROUTINE, CAN GENERATE THE ACTUAL CODE THAT CALLS             10378000 T  0003!087712
                   THE PRINTI ROUTINE AT OBJECT TIME.  THE ONE EXCEPTION IS              10379000 T  0003!087712
                   THE CASE OF A SUBSCRIPTED VARIABLE WITH AN EXPRESSION FOR             10380000 T  0003!087712
                   A SUBSCRIPT.  THE CODE FOR THE EXPRESSION IS GENERATED, AN            10381000 T  0003!087712
```

ACCIDENTAL ENTRY PROGRAM DESCRIPTOR IS CREATED, AND THE
ADDRESS OF THE DESCRIPTOR IS REMEMBERED.
  THE PRINTI ROUTINE IS AN INTRINSIC WHICH PRINTS THE
INFORMATION IT RECEIVES ACCORDING TO A SPECIFIED FORMAT
FOR BOTH MONITORING AND DUMPING. THE FOLLOWING CHART
EXPLAINS THE VARIOUS ACTIONS TAKEN BY THE PRINTI ROUTINE
AND THE PARAMETERS THAT MUST BE PASSED FOR THE FIVE
POSSIBLE CALLS ON PRINTI.

ID IS DEFINED TO MEAN THE FIRST SEVEN CHARACTERS OF THE IDENTIFIER TO BE PRINTED.

N IS DEFINED TO MEAN THE NUMBER OF DIMENSIONS OF AN ARRAY OR SUBSCRIPTED VARIABLE.

V IS DEFINED TO MEAN THE VALUE TO BE PRINTED.

S1---SN IS DEFINED TO MEAN THE SUBSCRIPT TO BE PRINTED.

S1*---SN* IS DEFINED TO MEAN THE SUBSCRIPT TO BE MONITORED. PRINTI COMPARES SN* TO SN AND PRINTS ONLY IF THEY ARE EQUAL.

FORMAT TYPE	MONITOR	DUMP
0	LABELS SWITCHES	
1	SIMPLE VARIABLES FUNCTION	LABELS SIMPLE VARIABLES
2	ARRAYS	SUBSCRIPTED VARS
3	SUBSCRIPTED VARS	
4		ARRAYS

FORMAT TYPE	PRINTOUT
0	ID
1	ID=V
2	ID[S1---SN]=V
3	ID[S1---SN]=V
4	ID=V1---VN

THE FORMAT THAT V IS PRINTED IN WILL BE DETERMINED BY
THE TYPE OF V. THE FOLLOWING CONVENTIONS APPLY FOR
PASSING THE TYPEV TO PRINTI.

TYPE	TYPEV
BOOLEAN	0
REAL	1
ALPHA	2
INTEGER	3

```
POWERSOFTEN IS A TABLE OF POWERS OF TEN THAT PRINTI
AND OTHER ROUTINES USE FOR CONVERSION PURPOSES.
FORMAT TYPE   ACTUAL PARAMETERS TO PRINTI
-----------   --------------------------
     0        <ID,CHARI,FILE,0>
 ---------    -------------------------
     1        (V,TYPEV,POWERSOFTEN,ID,CHARI,FILE,1)
 ---------    -------------------------
     2        (S1---SN,V,N,TYPEV,POWERSOFTEN,ID,CHARI,
               FILE,2)
 ---------    -------------------------
     3        (S1*---SN*,S1---SN,V,N,TYPEV,POWERSOFTEN
               ,ID,CHARI,FILE,3)
 ---------    -------------------------
     4        (DESCRIPTOR FOR THE ARRAY,N,TYPEV,
               POWERSOFTEN,ID,CHARI,FILE,4)
**********   ***************************
  SINCE THE RESTRICTION EXISTS THAT THE SCOPE OF THE
MONITOR FOR A LABEL OR SWITCH MUST BE THE SAME AS
THE SCOPE OF THE LABEL OR SWITCH, THE INFORMATION
THAT IS GATHERED BY THE MONITOR IS STORED IN THE
ORIGIONAL ENTRY IN INFO.  IN THE CASES OF VARIABLES,
ARRAYS, AND FUNCTION DESIGNATORS,THE MONITORS SCOPE
MAY BE DIFFERENT THAN THE SCOPE OF THE ITEM BEING
MONITORED, THEREFORE, A NEW ENTRY IS MADE IN INFO
WITH THE CURRENT LEVEL COUNTER AND THE ADDITIONAL
MONITORING INFORMATION.
*********FORMAT OF INFO FOR MONITORED ITEMS**********
ALL MONITORED ITEMS- MONITOR BIT [1:1] IN THE ELBAT
   WORD WILL BE SET.
SIMPLE VARIABLES- A NEW ENTRY IS MADE IN INFO WITH
      ONE EXTRA WORD WHICH CONTAINS THE ADDRESS OF
      THE MONITOR FILE IN [37:11]. I WILL HAVE A
      DEFINE SVARMONFILE = [37:11]#.
ARRAYS- A NEW ENTRY IS MADE IN INFO WITH THE SAME
      NUMBER OF WORDS AS THE ORIGIONAL ENTRY.  THE
      MONITOR FILE IS REMEMBERED IN [27:11] OF THE
      FIRST WORD OF ADDITIONAL INFO.  I WILL HAVE A
      DEFINE ARRAYMONFILE = [27:11]#.
SUBSCRIPTED VARIABLES- THE TECHNIQUE FOR HANDLING
            SUBSCRIPTED VARIABLES IS IDENTICLE TO THE
            TECHNIQUE FOR ARRAYS EXCEPT THAT EACH WORD
            OF INFO CONTAINING LOWER BOUND INFORMATION
            ALSO CONTAINS MONITOR INFORMATION.  EITHER
            A LITERAL OR AN ADDRESS WILL BE CONTAINED
            IN BITS [12:11].  IN [11:1] IS A BIT THAT
            DESIGNATES WHETHER AN OPDC OR A LITC
            SHOULD BE GENERATED USING [12:11].  IF THE
            BIT IS 1 THEN AN OPDC WILL BE GENERATED.
            ELSE A LITC.  IF AN OPDC IS GENERATED IT
            MAY BE ON A SIMPLE VARIABLE, OR ON AN
            ACCIDENTAL ENTRY PROGRAM DESCRIPTOR.  THE
            PURPOSE OF THE LITC OR OPDC IS TO PASS
            S1* TO THE PRINTI ROUTINE.
LABELS- THE FIRST WORD OF ADDITIONAL INFO CONTAINS
      THE ADDRESS OF THE FILE DESCRIPTOR IN THE
      ORIGIONAL ENTRY IN BITS [13:11].  I WILL HAVE A
```

```
10439000 T  00031087712
10440000 T  00031087712
10441000 T  00031087712
10442000 T  00031087712
10443000 T  00031087712
10444000 T  00031087712
10445000 T  00031087712
10446000 T  00031087712
10447000 T  00031087712
10448000 T  00031087712
10449000 T  00031087712
10450000 T  00031087712
10451000 T  00031087712
10452000 T  00031087712
10453000 T  00031087712
10454000 T  00031087712
10455000 T  00031087712
10456000 T  00031087712
10457000 T  00031087712
10458000 T  00031087712
10459000 T  00031087712
10460000 T  00031087712
10461000 T  00031087712
10462000 T  00031087712
10463000 T  00031087712
10464000 T  00031087712
10465000 T  00031087712
10466000 T  00031087712
10467000 T  00031087712
10468000 T  00031087712
10469000 T  00031087712
10470000 T  00031087712
10471000 T  00031087712
10472000 T  00031087712
10473000 T  00031087712
10474000 T  00031087712
10475000 T  00031087712
10476000 T  00031087712
10477000 T  00031087712
10478000 T  00031087712
10479000 T  00031087712
10480000 T  00031087712
10481000 T  00031087712
10482000 T  00031087712
10483000 T  00031087712
10484000 T  00031087712
10485000 T  00031087712
10486000 T  00031087712
10487000 T  00031087712
10488000 T  00031087712
10489000 T  00031087712
10490000 T  00031087712
10491000 T  00031087712
10492000 T  00031087712
10493000 T  00031087712
10494000 T  00031087712
10495000 T  00031087712
```

```
        DEFINE LABLMONFILE = [13:11]#,                                  10496000 T  0003:087712
    SWITCHES- THE MONITOR IS THE SAME AS THAT FOR LABELS,               10497000 T  0003:087712
        I WILL HAVE A DEFINE SWITMONFILE = [13:11]#,                    10498000 T  0003:087712
    FUNCTION DESIGNATORS- A NEW ENTRY IS MADE IN INFO                   10499000 T  0003:087712
        WITH THE SAME NUMBER OF WORDS AS THE                            10500000 T  0003:087712
        ORIGIONAL ENTRY,  THE MONITOR FILE IS                           10501000 T  0003:087712
        REMEMBERED IN [27:11] OF THE FIRST WORD OF                      10502000 T  0003:087712
        ADDITIONAL INFO,  I WILL HAVE A DEFINE                          10503000 T  0003:087712
        FUNCMONFILE = [27:11]#;                                         10504000 T  0003:087712
DEFINE FILEIDENT = RR7#;  COMMENT FILEIDENT CONTAINS THE                10505000 T  0003:087712
                                                          START OF SEGMENT ********** 124
                          ADDRESS OF THE MONITOR FILE;                  10506000 T  0124:000010
DEFINE SUBSCRIPT = RR1#;  COMMENT SUBSCRIPT IS USED TO                  10507000 T  0124:000010
                          SAVE THE ADDRESS OR VALUE OF A                10508000 T  0124:000010
                          SUBSCRIPT,  ONE ADDITIONAL BIT IS             10509000 T  0124:000010
                          USED TO TELL WHETHER TO EMIT AN               10510000 T  0124:000010
                          OPDC OR A LITC ON THIS ADDRESS OR             10511000 T  0124:000010
                          VALUE;                                        10512000 T  0124:000010
DEFINE NODIM = RR2#;  COMMENT NODIM CONTAINS THE NUMBER OF              10513000 T  0124:000010
                      DIMENSIONS OF AN ARRAY OR SUBSCRIPTED             10514000 T  0124:000010
                      VARIABLE APPEARING IN A MONITOR LIST;             10515000 T  0124:000010
DEFINE INC = RR3#;  COMMENT INC CONTAINS THE LINK TO                    10516000 T  0124:000010
                    ADDITIONAL INFO AND IS USED WHEN MAKING             10517000 T  0124:000010
                    A NEW ENTRY IN INFO FOR ARRAYS;                     10518000 T  0124:000010
DEFINE ELBATWORD = RR4#;  COMMENT ELBATWORD CONTAINS THE                10519000 T  0124:000010
                          ELBAT WORD FOR A MONITOR LIST                 10520000 T  0124:000010
                          ELEMENT;                                      10521000 T  0124:000010
DEFINE OPLIT = RR4#;      COMMENT OPLIT IS USED FOR MARKING             10522000 T  0124:000010
                        SUBSCRIPTED VARIABLES TO TELL ME                10523000 T  0124:000010
                        WHETHER TO EMIT AN OPDC OR A LITC,              10524000 T  0124:000010
                        0 IS USED FOR OPDC, 1 FOR LITC;                 10525000 T  0124:000010
DEFINE TESTVARB = RR5#;  COMMENT TESTVARB CONTAINS A LINK               10526000 T  0124:000010
                         POINTING AT THE END OF ADDITIONAL              10527000 T  0124:000010
                         INFO AND IS USED TO TELL WHEN TO               10528000 T  0124:000010
                         STOP MOVING INFO FOR THE NEW ENTRY             10529000 T  0124:000010
                         FOR MONITORED ARRAYS;                          10530000 T  0124:000010
DEFINE NXTINFOTEMP = RR6#;  COMMENT NXTINFOTEMP CONTAINS A              10531000 T  0124:000010
                            LINK POINTING AT THE FIRST                  10532000 T  0124:000010
                            ADDITIONAL WORD OF INFO FOR                 10533000 T  0124:000010
                            MONITORED ARRAYS;                           10534000 T  0124:000010
DEFINE INSERTFILE = 27:37:11#;  COMMENT INSERTFILE IS THE               10535000 T  0124:000010
                                CONCATENATE DEFINE FOR                  10536000 T  0124:000010
                                STUFFING THE MONITOR FILE               10537000 T  0124:000010
                                ADDRESS INTO THE FIRST                  10538000 T  0124:000010
                                ADDITIONAL INFO WORD FOR                10539000 T  0124:000010
                                ARRAYS AND FUNCTIONS;                   10540000 T  0124:000010
DEFINE NOPARPART = NODIMPART#;  COMMENT NOPARPART IS A                  10541000 T  0124:000010
                                PARTIAL WORD DESIGNATOR [40             10542000 T  0124:000010
                                :8] USED TO EXTRACT THE                 10543000 T  0124:000010
                                NUMBER OF PARAMETERS FOR A              10544000 T  0124:000010
                                GIVEN PROCEDURE FROM INFO;              10545000 T  0124:000010
DEFINE NOPAR = NODIM#;  COMMENT NOPAR CONTAINS THE NUMBER               10546000 T  0124:000010
                        OF PARAMETERS FOR A FUNCTION                    10547000 T  0124:000010
                        DESIGNATOR APPEARING IN A MONITOR               10548000 T  0124:000010
                        LIST;                                           10549000 T  0124:000010
LABEL START;  COMMENT WHEN START IS REACHED, I MUST BE                  10550000 T  0124:000010
              POINTING AT THE FILE IDENTIFIER IN THE                    10551000 T  0124:000010
              MONITOR DECLARATION;                                      10552000 T  0124:000010
```

```
                LABEL MARKMONITORED;  COMMENT  THE CODE AT MARKMONITORED
                                      TURNS ON THE MONITOR BIT OF THE ELBAT
                                      WORD IN THE MONITOR LIST AND STORES
                                      IT IN ACCUM[0] FOR THE E ROUTINE;
                LABEL STORESUBS;  COMMENT STORESUBS IS THE CODE THAT
                                  REMEMBERS ALL THAT IS NECESSARY ABOUT
                                  EACH SUBSCRIPT EXPRESSION;
                LABEL CHKCOMMA;  COMMENT CHKCOMMA REQUIRES THAT I BE
                                 POINTING THE LAST LOGICAL QUANTITY OF THE
                                 MONITOR LIST ELEMENT THAT HAS JUST BEEN
                                 PROCESSED;
                LABEL EXIT;  COMMENT EXIT EXITS THE MERRIMAC PROCEDURE;
          START:IF ELCLASS≠FILEID THEN
                BEGIN IF Q="5INDEX" OR Q="4FLAGO" OR Q="6INTOV" OR Q=
                     "6EXPOV" OR Q="4ZEROO"THEN MERRIMAC←TRUE ELSE
                          ERR(400);  GO EXIT;
                END
                COMMENT ERROR 400 IS MISSING FILE ID IN MONITOR DEC;
                     CHECKER(ELBAT[I]);
                FILEIDENT←ELBAT[I].ADDRESS;              I←I+1;
                IF CHECK(LEFTPAREN,40)
                THEN GO TO EXIT;
                COMMENT ERROR 401 IS MISSING LEFT PARENTHSIS IN MONITOR;
MARKMONITORED:STEPIT;            ACCUM[0]←-ABS(ELBAT[I]);
                IF RANGE(BOOID,INTID)
                THEN BEGIN COMMENT THIS CODE HANDLES SIMPLE VARIABLES;
                          E;              PUTNBUMP(FILEIDENT);
                          GO CHKCOMMA;
                     END;
          IF RANGE(BOOARRAYID,INTARRAYID)
          THEN BEGIN COMMENT THIS CODE HANDLES ARRAYS AND
                     SUBSCRIPTED VARIABLES;
                     E;                  NXTINFOTEMP←NEXTINFO;
                     PUTNBUMP(NODIM←TAKEFRST&FILEIDENT[INSERTFILE]);
                     TESTVARB←(NODIM←NODIM, NODIMPART )+(INC←(
                     ELBATWORD←ELBAT[I]).LINK←ELBATWORD.INCR);
                     DO PUTNBUMP(TAKE(INC←INC+1))
                     UNTIL INC ≥ TESTVARB;
                     IF TABLE(I+1) ≠ LFTBRKET
                     THEN GO CHKCOMMA;
                     TESTVARB←NODIM←NXTINFOTEMP;
                     STEPIT;
          STORESUBS:IF(RR3←TABLE(I+2) = COMMA OR RR3 = RTBRKET) AND
                      STEPI ≠ NONLITNO
                     THEN BEGIN COMMENT THIS IS THE SIMPLE CASE OF
                                SUBSCRIPTED VARIABLES,  EITHER A LITC
                                OR AN OPDC ON A VARIABLE IS ALL THAT
                                IS NEEDED TO CALL THE SUBSCRIPT;
                                SUBSCRIPT←ELBAT[I].ADDRESS;
                                OPLIT←0;
                                IF NOT RANGE( INTRNSICPROCID,INTID)
                                THEN IF CHECK(LITNO,402)
                                     THEN GO TO EXIT
                                     ELSE COMMENT MARK FOR LITC;
                                          OPLIT←1;
                                      COMMENT ERROR 402 IS BAD
                                     SUBSCRIPT IN MONITOR DECLARATION;
```

```
            STEPIT;                                                    10606000 T  01241005:3
        END                                                            10607000 T  01241005:6:0
    ELSE BEGIN COMMENT THIS IS THE SPECIAL CASE OF                     10608000 T  01241005:6:0
            SUBSCRIPTED VARIABLES.  CODE FOR THIS                      10609000 T  01241005:6:1
            SUBSCRIPT EXPRESSION MUST BE GENERATED                     10610000 T  01241005:6:1
            AND JUMPED AROUND, AN ACCIDENTAL ENTRY                     10611000 T  01241005:6:1
            PROGRAM DESCRIPTOR CREATED AND THE                         10612000 T  01241005:6:1
            ADDRESS SAVED IN SUBSCRIPT.  SUBSCRIPT                     10613000 T  01241005:6:1
            MUST BE MARKED FOR AN OPDC;                                10614000 T  01241005:6:1
            JUMPCHKNX;  SUBSCRIPT=PROGDESCBLDR(                        10615000 T  01241005:6:1
            ADES,L,0);  AEXP;        EMITO(RTS);                       10616000 T  01241005:7:2
            JUMPCHKX;                                                  10616500 T  01241006:0
            OPLIT=0;                                                   10617000 T  01241006:1
            IF MODE > 0                                                10618000 T  01241006:1:2
            THEN BEGIN COMMENT STUFF F AT THIS                         10619000 T  01241006:1:3
                    POINT IF MODE > 0;                                 10620000 T  01241006:2:1
                    STUFFF(SUBSCRIPT);EMITPAIR(                        10621000 T  01241006:2:1
                    SUBSCRIPT,STD);                                    10622000 T  01241006:3:3
                END;                                                   10623000 T  01241006:4:0
        END;                                                           10624000 T  01241006:4:0
    PUT(TAKE(NXTINFOTEMP=NXTINFOTEMP+1) &                              10625000 T  01241006:4:0
    SUBSCRIPT[12:37:11] & OPLIT[11:47:01],                             10626000 T  01241006:6:0
    NXTINFOTEMP);                                                      10627000 T  01241006:8:0
    IF ELCLASS = COMMA                                                 10628000 T  01241006:8:1
    THEN GO TO STORESUBS;                                              10629000 T  01241006:9:2
    IF CHECK(RTBRKET,403)                                              10630000 T  01241007:0:0
    THEN GO TO EXIT;                                                   10631000 T  01241007:0:1
     COMMENT ERROR 403 IS IMPROPER SUBSCRIPT                           10632000 T  01241007:1:3
    EXPRESSION DELIMITER IN MONITOR LIST ELEMENT;                      10633000 T  01241007:1:3
                    IF NXTINFOTEMP # TESTVARB                          10634000 T  01241007:1:3
                    THEN BEGIN COMMENT ERROR 404 MONITOR LIST          10635000 T  01241007:2:0
                            ELEMENT HAS IMPROPER NUMBER OF             10636000 T  01241007:3:2
                            SUBSCRIPTS;                                10637000 T  01241007:3:2
                            I=I-1;       ERROR(404); GO TO EXIT;       10638000 T  01241007:3:2
                        END;                                           10639000 T  01241007:5:3
                    GO CHKCOMMA;                                       10640000 T  01241007:5:3
                END;                                                   10641000 T  01241007:6:0
            IF ELCLASS = LABELID OR ELCLASS = SWITCHID                 10642000 T  01241007:6:0
            THEN BEGIN COMMENT THIS CODE HANDLES LABELS AND SWITCHES;   10643000 T  01241007:7:2
                    IF(ELBATWORD=ELBAT[I]).LVL # LEVEL                 10644000 T  01241007:8:0
                    THEN BEGIN COMMENT ERROR 405 MEANS LABEL OR        10645000 T  01241007:9:3
                            SWITCH MONITORED AT IMPROPER LEVEL;        10646000 T  01241008:0:1
                            ERROR(405); GO TO EXIT;                    10647000 T  01241008:0:1
                        END;                                           10648000 T  01241008:2:0
                    PUT(TAKEFRST & FILEIDENT[13:37:11],GIT(ELBAT[I])   10649000 T  01241008:2:0
                    );                                                 10650000 T  01241008:4:0
                    PUT(TAKE(ELBATWORD)&(0=ABS(ELBATWORD))[1:1:34],    10651000 T  01241008:5:2
                        ELBATWORD); GO CHKCOMMA;                       10652000 T  01241008:7:3
                END;                                                   10653000 T  01241008:8:1
            IF RANGE(BOOPROCID,INTPROCID)                              10654000 T  01241008:8:1
            THEN BEGIN COMMENT THIS CODE HANDLES FUNCTIONS;            10655000 T  01241008:9:2
                E ;%                                                   10656000 T  01241009:0:0
    IF LEVEL=(RR2=ELBAT[I]).LVL THEN                                   10656010 T  01241009:0:1
       BEGIN                                                           10656011 T  01241009:2:1
       %%% COPY FORWARD BIT FROM ELBAT[I] INFO ENTRY INTO MONITOR"S INFO  10656012 T  01241009:3:2
       %%% ENTRY, AND THEN TURN OFF THE ELBAT[I] INFO ENTRY"S FORWARD BIT.  10656013 T  01241009:3:2
       PUT(TAKE(LASTINFO+1) & TAKE(RR2.LINK+1)[1:1:1],LASTINFO+1) ;    10656014 T  01241009:3:2
```

```
PUT(ABS(TAKE(RR2,LINK+1)),RR2,LINK+1) ;                                  10656015 T  0124:009810
END ;                                                                    10656016 T  0124:010210
              PUTNBUMP(NOPAR + TAKEFRST &                                10656030 T  0124:010210
                    FILEIDENT[INSERTFILE]);          TESTVARB+(NOPAR     10657000 T  0124:010211
                    +NOPAR, NOPARPART )+(INC+(ELBATWORD+ELBAT[I]),       10658000 T  0124:010411
                    LINK+ELBATWORD,INCR);                                10659000 T  0124:010611
                    DO PUTNBUMP(TAKE(INC+INC+1))                         10660000 T  0124:010913
                    UNTIL INC ≥ TESTVARB;                                10661000 T  0124:011113
                    GO CHKCOMMA;                                         10662000 T  0124:011313
               END;                                                      10663000 T  0124:011410
          ERROR(406);         GO TO EXIT;                                10664000 T  0124:011410
           COMMENT ERROR 406 IS IMPROPER MONITOR LIST ELEMENT;           10665000 T  0124:011512
CHKCOMMA: IF STEPI = COMMA                                               10666000 T  0124:011512
          THEN GO MARKMONITORED;                                         10667000 T  0124:011611
          IF CHECK(RTPAREN,407)                                          10668000 T  0124:011713
          THEN GO TO EXIT;                                               10669000 T  0124:011810
           COMMENT ERROR 407 IS MISSING RIGHT PARENTHESIS IN MONITOR     10670000 T  0124:011912
                 DECLARATION;                                            10671000 T  0124:011912
          IF STEPI = SEMICOLON                                           10672000 T  0124:011912
          THEN GO TO EXIT;                                               10673000 T  0124:011913
          IF CHECK(COMMA,408)                                            10674000 T  0124:012011
          THEN GO TO EXIT;                                               10675000 T  0124:012112
           COMMENT ERROR 408 MEANS IMPROPER MONITOR DECLARATION          10676000 T  0124:012211
          DELIMITER;                                                     10677000 T  0124:012211
          STEPIT;             GO TO START;                               10678000 T  0124:012211
     EXIT:;                                                              10679000 T  0124:012313
     END MERRIMAC;                                                       10680000 T  0124:012410
                                                                  124 IS  127 LONG, NEXT SEG    3
```

```
                    PROCEDURE DMUP;                                      10681000 T  0003:087712
PRT(1065) = DMUP
                BEGIN COMMENT                15 JULY 1963                10682000 T  0003:087712
                  THERE ARE FOUR TYPES OF DUMP  LIST ELEMENTS, THERE     10683000 T  0003:087712
                ARE LABELS, SIMPLE VARIABLES, SUBSCRIPTED VARIABLES, AND 10684000 T  0003:087712
                ARRAYS.                                                  10685000 T  0003:087712
                  THE DMUP ROUTINE GENERATES CODE AND SAVES INFORMATION. 10686000 T  0003:087712
                THE INFORMATION THAT IS SAVED IS OF TWO TYPES.  FOR EASE 10687000 T  0003:087712
                OF REFERENCE I WOULD LIKE TO DEFINE THE DUMP LABEL OUTSIDE 10688000 T  0003:087712
                THE PARENTHESES AS THE DUMPOR, AND ANY LABEL APPEARING AS 10689000 T  0003:087712
                A DUMP LIST ELEMENT A DUMPEE.  BOTH DUMPORS AND DUMPEES  10690000 T  0003:087712
                HAVE A COUNTER ASSOCIATED WITH THEM WHICH IS INCREMENTED  10691000 T  0003:087712
                BY ONE EACH TIME THE LABEL IS PASSED.  THE ADDRESS OF THIS 10692000 T  0003:087712
                COUNTER IS KEPT IN BITS [21:11] OF THE FIRST ADDITIONAL  10693000 T  0003:087712
                WORD OF INFO.  THE ADDRESS OF THE PROGRAM DESCRIPTOR FOR  10694000 T  0003:087712
                THE CODE GENERATED BY DMUP IS KEPT IN BITS [24:11] OF THE 10695000 T  0003:087712
                FIRST ADDITIONAL WORD OF INFO FOR THE DUMPOR.            10696000 T  0003:087712
                  THE CODE THAT IS GENERATED IS OF TWO TYPES.  CODE TO   10697000 T  0003:087712
                INITIALIZE THE COUNTERS MENTIONED ABOVE IS EXECUTED UPON 10698000 T  0003:087712
                ENTRY TO THE BLOCK CONTAINING THE DUMP DECLARATION.  THE 10699000 T  0003:087712
                OTHER TYPE CODE IS ONLY EXECUTED WHEN THE DUMPOR IS PASSED 10700000 T  0003:087712
                .  THIS CODE THEN COMPARES THE DUMPORS COUNTER WITH THE   10701000 T  0003:087712
                DUMP INDICATOR.  IF THEY ARE NOT EQUAL IT JUMPS TO EXIT. 10702000 T  0003:087712
                IF THEY ARE EQUAL IT THEN PROCEEDS TO CALL PRINTI ONCE   10703000 T  0003:087712
                FOR EACH DUMP LIST ELEMENT.  FOR A DESCRIPTION OF PRINTI 10704000 T  0003:087712
```

```
SEE THE COMMENTS FOR THE MERRIMAC ROUTINE;                                10705000 T  0003;0877;2
LABEL START;  COMMENT WHEN START IS REACHED, I MUST BE                    10706000 T  0003;0877;2
                                                          START OF SEGMENT ********** 125
          POINTING AT THE FILE INDENTIFIER IN THE DUMP                    10707000 T  0125;0000;0
          DECLARATION;                                                    10708000 T  0125;0000;0
LABEL EXIT;  COMMENT EXIT APPEARS AT THE END OF THE DMUP                  10709000 T  0125;0000;0
          ROUTINE,  NO STATMENTS ARE EXECUTED AFTER IT                    10710000 T  0125;0000;0
          IS REACHED;                                                     10711000 T  0125;0000;0
DEFINE FILEIDENT = RR1#;  COMMENT FILEIDENT CONTAINS THE                  10712000 T  0125;0000;0
                          ADDRESS OF THE MONITOR FILE;                    10713000 T  0125;0000;0
LABEL STARTCALL;  COMMENT THE CODE AT STARTCALL GENERATES                 10714000 T  0125;0000;0
                  CODE TO CALL THE PRINTI ROUTINE,  WHEN                  10715000 T  0125;0000;0
                  STARTCALL IS REACHED, I MUST BE POINTING                10716000 T  0125;0000;0
                  AT THE CHARACTER IMMEDIATELY BEFORE THE                 10717000 T  0125;0000;0
                  DUMP LIST ELEMENT TO BE PASSED TO PRINTI;               10718000 T  0125;0000;0
DEFINE NODIM = RR2#;  COMMENT NODIM CONTAINS THE NUMBER OF                10719000 T  0125;0000;0
                      DIMENSIONS OF AN ARRAY OR A                         10720000 T  0125;0000;0
                      SUBSCRIPTED VARIABLE APPEARING IN A                 10721000 T  0125;0000;0
                      DUMP LIST;                                          10722000 T  0125;0000;0
DEFINE LEXIT = RR3#;  COMMENT LEXIT CONTAINS THE PROGRAM                  10723000 T  0125;0000;0
                      COUNTER SETTING AT WHICH CODE IS                    10724000 T  0125;0000;0
                      GENERATED TO EXIT THE ROUTINE EMITTED               10725000 T  0125;0000;0
                      BY DMUP;                                            10726000 T  0125;0000;0
DEFINE DUMPETEMP = RR2#;  COMMENT DUMPETEMP HOLDS THE                     10727000 T  0125;0000;0
                          LOCATION OF THE COUNTER                         10728000 T  0125;0000;0
                          ASSOCIATED WITH THIS LABEL IF                   10729000 T  0125;0000;0
                          SPACE HAS BEEN ASSIGNED FOR IT;                 10730000 T  0125;0000;0
DEFINE DIMCTR = RR3#;  COMMENT DIMCTR IS INITIALIZED TO                   10731000 T  0125;0000;0
                       NODIM,  IT IS THEN COUNTED DOWN TO                 10732000 T  0125;0000;0
                       ZERO AS SUBSCRIPT CODE IS GENERATED;               10733000 T  0125;0000;0
LABEL PASSN;  COMMENT THE CODE AT PASSN PASSES N (THE                     10734000 T  0125;0000;0
              NUMBER OF DIMENSIONS) TO THE PRINTI ROUTINE;                10735000 T  0125;0000;0
LABEL SUBSLOOP;  COMMENT THE CODE AT SUBSLOOP PASSES                      10736000 T  0125;0000;0
                 SUBSCRIPTS TO PRINTI;                                    10737000 T  0125;0000;0
ARRAY LABELCTR[0:100];  COMMENT LABELCTR IS AN ARRAY THAT                 10738000 T  0125;0000;0

STACK(F+2) = LABELCTR                 HOLDS THE ADDRESSES OF ALL LABEL    10739000 T  0125;0001;3
                                      COUNTERS FOR LABELS APPEARING IN    10740000 T  0125;0001;3
                                      THIS DUMP DECLARATION,  IT IS       10741000 T  0125;0001;3
                                      NECESSARY TO RETAIN THIS            10742000 T  0125;0001;3
                                      INFORMATION SO THAT CODE MAY BE     10743000 T  0125;0001;3
                                      GENERATED AT THE END OF THE         10744000 T  0125;0001;3
                                      DECLARATION TO INITIALIZE THE       10745000 T  0125;0001;3
                                      COUNTERS;                           10746000 T  0125;0001;3
DEFINE LABELCTRINX = RR4#;  COMMENT LABELCTRINX IS THE                    10747000 T  0125;0001;3
                            VARIABLE USED TO INDEX INTO THE               10748000 T  0125;0001;3
                            LABELCTR ARRAY;                               10749000 T  0125;0001;3
DEFINE DUMPE =  2;37;11#;  COMMENT DUMPE IS THE                           10750000 T  0125;0001;3
                           CONCATENATE DEFINE FOR INSERTING               10751000 T  0125;0001;3
                           THE COUNTER ASSOCIATED WITH THIS               10752000 T  0125;0001;3
                           LABEL INTO THE FIRST ADDITIONAL                10753000 T  0125;0001;3
                           WORD OF INFO;                                  10754000 T  0125;0001;3
DEFINE LWRBND = RR5#;  COMMENT LWRBND CONTAINS THE LOWER                  10755000 T  0125;0001;3
                       BOUND FOR MONITORED SUBSCRIPTED                    10756000 T  0125;0001;3
                       VARIABLES;                                         10757000 T  0125;0001;3
DEFINE FORMATTYPE = RR5#;  COMMENT FORMATTYPE IS THE                      10758000 T  0125;0001;3
                           FORMAT TYPE REFERRED TO IN THE                 10759000 T  0125;0001;3
                           COMMENTS FOR THE MERRIMAC                      10760000 T  0125;0001;3
```

```
                                    ROUTINE DESCRIBING PRINTI;
DEFINE FINALL = RR5#;  COMMENT FINALL IS A TEMPORARY CELL
                      USED TO HOLD L WHILE THE DUMP
                      INDICATOR TEST CODE IS BEING
                      GENERATED;
DEFINE TESTLOC = RR6#;  COMMENT TESTLOC CONTAINS THE
                       LOCATION OF THE CODE THAT MUST BE
                       GENERATED TO MAKE THE TEST TO
                       DETERMINE WHETHER OR NOT DUMPING
                       SHOULD OCCUR;
DEFINE DUMPR = 24:37:11#;  COMMENT DUMPR IS THE
                          CONCATENATE DEFINE USED TO
                          INSERT THE ADDRESS OF THE
                          PROGRAM DESCRIPTOR FOR THE CODE
                          GENERATED FROM THE DUMP
                          DECLARATION;
DEFINE DUMPLOC = RR7#;  COMMENT DUMPLOC CONTAINS THE
                       ADDRESS OF THE PROGRAM DESCRIPTOR
                       THAT DESCRIBES THE CODE GENERATED
                       BY DMUP;
DEFINE ELBATWORD = RR8#;  COMMENT ELBATWORD CONTAINS THE
                         ELBAT WORD FOR THE DUMP LIST
                         ELEMENT CURRENTLY BEING OPERATED
                         ON;
LABEL CALLPRINTI;  COMMENT CALLPRINTI FINISHES THE CALL
                  ON PRINTI,  IT GENERATES THE CODE TO
                  PASS TYPEV, POWERSOFTEN, ID, CHARI,
                  FILE, AND FORMAT TYPE;
DEFINE SUBSCTR = RR9#;  COMMENT SUBSCTR CONTAINS THE
                       DIMENSION NUMBER THAT IS CURRENTLY
                       BEING WORKED ON;
START:IF CHECK(FILEID,409)
      THEN GO TO EXIT;
      COMMENT ERROR 409 MEANS MISSING FILE ID IN DUMP DEC;
            CHECKER(ELBAT[I]);
      FILEIDENT←ELBAT[I].ADDRESS;          STEPIT;
      IF CHECK(LEFTPAREN,410)
      THEN GO TO EXIT;
      COMMENT ERROR 410 MEANS MISSING LEFT PAREN IN DUMP DEC;
      JUMPCHKNX;          ADJUST;          DUMPLOC←PROGDESCBLDR
      (ADES,L,0);         TESTLOC←L;       L←L+3;
      LABELCTRINX←-1;     EMITO(NOP);      BUMPL;
STARTCALL:EMITO(MKS);     STEPIT;          ELBATWORD←-ABS(ELBAT
      [I]);
      IF RANGE(BOOARRAYID,INTARRAYID)
      THEN BEGIN COMMENT THIS CODE HANDLES ARRAYS AND
                 SUBSCRIPTED VARIABLES;
                 NODIM←DIMCTR←TAKEFRST,NODIMPART;
                 IF STEPI = LFTBRKET
                 THEN BEGIN COMMENT THIS CODE HANDLES SUBSCRIPTED
                            VARIABLES;
                            STEPIT;     AEXP;       EMITO(DUP);
                            SUBSCTR←1;
                            IF(LWRBND←TAKE(GIT(ELBATWORD)+SUBSCTR)
                              ).[35:13] ≠ 0
                            THEN BEGIN COMMENT SUBTRACT OFF THE
                                       LOWER BOUND BEFORE INDEXING;
```

```
10761000 T  0125:000:3
10762000 T  0125:000:3
10763000 T  0125:000:3
10764000 T  0125:000:3
10765000 T  0125:000:3
10766000 T  0125:000:3
10767000 T  0125:000:3
10768000 T  0125:000:3
10769000 T  0125:000:3
10770000 T  0125:000:3
10771000 T  0125:000:3
10772000 T  0125:000:3
10773000 T  0125:000:3
10774000 T  0125:000:3
10775000 T  0125:000:3
10776000 T  0125:000:3
10777000 T  0125:000:3
10778000 T  0125:000:3
10779000 T  0125:000:3
10780000 T  0125:000:3
10781000 T  0125:000:3
10782000 T  0125:000:3
10783000 T  0125:000:3
10784000 T  0125:000:3
10785000 T  0125:000:3
10786000 T  0125:000:3
10787000 T  0125:000:3
10788000 T  0125:000:3
10789000 T  0125:000:3
10790000 T  0125:000:3
10791000 T  0125:000:3
10792000 T  0125:000:3
10793000 T  0125:000:1
10794000 T  0125:000:3
10795000 T  0125:000:3
10796000 T  0125:000:1
10797000 T  0125:000:1
10798000 T  0125:000:2
10799000 T  0125:000:1
10800000 T  0125:000:1
10801000 T  0125:000:3
10802000 T  0125:001:2
10803000 T  0125:001:0
10804000 T  0125:001:0
10805000 T  0125:001:3
10806000 T  0125:002:0
10807000 T  0125:002:2
10808000 T  0125:002:2
10809000 T  0125:002:2
10810000 T  0125:002:3
10811000 T  0125:002:1
10812000 T  0125:002:1
10813000 T  0125:002:1
10814000 T  0125:002:2
10815000 T  0125:002:0
10816000 T  0125:003:0
10817000 T  0125:003:2
```

```
                    IF LWRBND.[46:2] = 0                                10818000 T  0125:0031:2
                    THEN EMIT(LWRBND)                                   10819000 T  0125:0031:3
                    ELSE EMITV(LWRBND.[35:11]);                         10820000 T  0125:0033:2
                    EMIT(LWRBND.[23:12]);                               10821000 T  0125:0035:2
               END;                                                     10822000 T  0125:0036:1
          IF DIMCTR=SUBSCTR = 0                                         10823000 T  0125:0036:1
          THEN BEGIN COMMENT PASS SUBSCRIPT,                            10824000 T  0125:0037:2
                    VALUE,N;                                            10825000 T  0125:0038:0
                    EMITV(ELBATWORD.ADDRESS);                           10826000 T  0125:0038:0
             PASSN:EMITL(NODIM);                                        10827000 T  0125:0039:3
                    IF CHECK(RTBRKET,411)                               10828000 T  0125:0040:1
                    THEN GO TO EXIT;                                    10829000 T  0125:0041:2
                     COMMENT ERROR 411 MEANS                            10830000 T  0125:0042:1
                    DUMP LIST ELEMENT HAS WRONG                         10831000 T  0125:0042:1
                    NUMBER OF SUBSCRIPTS;                               10832000 T  0125:0042:1
                    FORMATTYPE←2;  GO CALLPRINTI                        10833000 T  0125:0042:1
               END;                                                     10834000 T  0125:0043:2
          EMITN(ELBATWORD.ADDRESS);                                     10835000 T  0125:0043:3
SUBSLOOP:EMITO(LOD);  STEPIT;      AEXP;                              10836000 T  0125:0045:2
          EMITL(JUNK); EMITO(SND);                                      10837000 T  0125:0046:1
          SUBSCTR←SUBSCTR+1;                                            10838000 T  0125:0048:0
          IF(LWRBND←TAKE(GIT(ELBATWORD)+SUBSCTR)                        10839000 T  0125:0049:3
            ).[35:13] ≠ 0                                               10840000 T  0125:0050:1
          THEN BEGIN COMMENT SUBTRACT OFF THE                           10841000 T  0125:0052:0
                    LOWER BOUND BEFORE INDEXING;                        10842000 T  0125:0053:2
                    IF LWRBND.[46:2] = 0                                10843000 T  0125:0053:2
                    THEN EMIT(LWRBND)                                   10844000 T  0125:0054:0
                    ELSE EMITV(LWRBND.[35:11]);                         10845000 T  0125:0055:2
                    EMIT(LWRBND.[23:12]);                               10846000 T  0125:0057:3
                    END;                                                10847000 T  0125:0058:1
               IF DIMCTR=SUBSCTR = 0                                    10848000 T  0125:0058:1
               THEN BEGIN COMMENT EMIT COC;                             10849000 T  0125:0059:3
                         EMITO(COC);       EMITV(JUNK                  10850000 T  0125:0060:1
                         );       EMITO(XCH);                           10851000 T  0125:0061:3
                         GO PASSN;                                      10852000 T  0125:0062:1
                    END;                                                10853000 T  0125:0063:2
               EMITO(CDC);  EMITV(JUNK);EMITO(XCH);                     10854000 T  0125:0063:2
               IF CHECK(COMMA,412)                                      10855000 T  0125:0065:3
               THEN GO TO EXIT                                          10856000 T  0125:0066:0
               ELSE GO TO SUBSLOOP;                                     10857000 T  0125:0066:1
                COMMENT ERROR 412 MEANS DUMP LIST                       10858000 T  0125:0067:3
               ELEMENT HAS WRONG NUMBER OF SUBSCRIPTS                   10859000 T  0125:0067:3
               ;                                                        10860000 T  0125:0067:3
          END;                                                          10861000 T  0125:0067:3
     COMMENT THIS CODE HANDLES ARRAYS;                                  10862000 T  0125:0067:3
     IF ELCLASS ≠ COMMA AND ELCLASS ≠ RTPAREN                           10863000 T  0125:0067:3
     THEN BEGIN COMMENT ERROR 413 MEANS IMPROPER                        10864000 T  0125:0068:1
               ARRAY DUMP LIST ELEMENT;                                 10865000 T  0125:0069:3
               ERR(413);    GO TO EXIT;                                 10866000 T  0125:0069:3
          END;                                                          10867000 T  0125:0071:2
     EMITPAIR(ELBATWORD.ADDRESS.LOD);EMITL(NODIM);                      10868000 T  0125:0071:2
     FORMATTYPE←4;   I←I-1;              GO CALLPRINTI;                 10869000 T  0125:0073:2
END;                                                                    10870000 T  0125:0075:3
     FORMATTYPE←1;                                                      10871000 T  0125:0075:3
IF RANGE(BOOID,INTID)                                                   10872000 T  0125:0076:1
THEN BEGIN COMMENT THIS CODE HANDLES SIMPLE VARIABLES;                  10873000 T  0125:0077:2
     EMITV(ELBATWORD.ADDRESS);          GO CALLPRINTI;                  10874000 T  0125:0078:0
END;                                                                    10875000 T  0125:0079:3
```

```
                IF CHECK(LABELID,414)                                                   10876000 T  0125:007:3
                THEN GO TO EXIT;                                                        10877000 T  0125:008:0
                 COMMENT ERROR 414 MEANS ILLEGAL DUMP LIST ELEMENT,  THIS               10878000 T  0125:008:3
                CODE HANDLES LABELS;                                                    10879000 T  0125:008:3
                PUT(TAKEFRST & (LABELCTR[LABELCTRINX←LABELCTRINX+1]←                    10880000 T  0125:008:3
                    IF DUMPETEMP←TAKEFRST.DUMPEE = 0                                    10881000 T  0125:008:3
                    THEN GETSPACE(FALSE,-7) % LABEL DESCRIPTOR,                         10882000 T  0125:008:2
                    ELSE DUMPETEMP)[DUMPE],GIT(ELBATWORD));                             10883000 T  0125:008:2
                    EMITV(LABELCTR[                                                     10884000 T  0125:009:1
                LABELCTRINX]);      PUT(TAKE(ELBATWORD) & ELBATWORD[1:1:34]             10885000 T  0125:009:1
                ,ELBATWORD);                                                            10886000 T  0125:009:1
                EMITL(3);           IF FALSE THEN                                       10887000 T  0125:009:0
CALLPRINT:      EMITL(PASSTYPE(ELBATWORD));         EMITPAIR(GNAT(                      10888000 T  0125:009:2
                POWERSOFTEN),LOD); PASSALPHA(ELBATWORD);                                10889000 T  0125:009:3
                EMITPAIR(GNAT(CHARI),LOD);          PASSMONFILE(                        10890000 T  0125:009:3
            FILEIDENT);                                                    %109-        10891000 P  0125:010:2
            EMITNUM(FORMATTYPE&CARDNUMBER[1:4:44]);                        %109-        10891100 C  0125:010:3
            EMITV(GNAT(PRINTI));                                           %109-        10891200 C  0125:010:3
                IF STEPI = COMMA                                                        10892000 T  0125:010:1
                THEN BEGIN COMMENT GO AROUND ONE MORE TIME;                             10893000 T  0125:010:2
                        IF LABELCTRINX = 100                                            10894000 T  0125:010:0
                        THEN BEGIN COMMENT ERROR 415 MEANS LABELCTR IS                  10895000 T  0125:010:1
                                ABOUT TO OVERFLOW WITH LABEL                            10896000 T  0125:010:3
                                INFORMATION;                                            10897000 T  0125:010:3
                                ERR(415);    GO TO EXIT;                                10898000 T  0125:010:3
                             END;                                                       10899000 T  0125:010:1
                        GO STARTCALL;                                                   10900000 T  0125:010:1
                    END;                                                                10901000 T  0125:010:2
                IF CHECK(RTPAREN,416)                                                   10902000 T  0125:010:2
                THEN GO TO EXIT;                                                        10903000 T  0125:010:3
                 COMMENT ERROR 416 MEANS ILLEGAL DUMP LIST ELEMENT                      10904000 T  0125:011:2
                DELIMITER;                                                              10905000 T  0125:011:2
                LEXIT←L;            EMITL(0);           EMITO(RTS);                     10906000 T  0125:011:2
                JUMPCHKX;           STEPIT;                                             10907000 T  0125:011:2
                IF CHECK(LABELID,417)                                                   10908000 T  0125:011:0
                THEN GO TO EXIT;                                                        10909000 T  0125:011:1
                 COMMENT ERROR 417 MEANS MISSING DUMP LABEL;                            10910000 T  0125:011:0
                PUT(TAKE(ELBATWORD←-ABS(ELBAT[I])) & ELBATWORD[1:1:34],                 10911000 T  0125:011:0
                    ELBATWORD);                                                         10912000 T  0125:011:2
                IF NOT LOCAL(ELBATWORD) THEN FLAG(417);                                 10912100 T  0125:011:3
                PUT(TAKEFRST & (LABELCTR[LABELCTRINX←LABELCTRINX+1]←                    10913000 T  0125:012:0
                    IF DUMPETEMP←TAKEFRST.DUMPEE = 0                                    10914000 T  0125:012:0
                    THEN DUMPETEMP:=GETSPACE(FALSE,-7) % LABEL DESCR,                   10915000 T  0125:012:3
                    ELSE DUMPETEMP)[DUMPE],GIT(ELBATWORD));                             10916000 T  0125:012:3
                    EMITL(0);                                                           10917000 T  0125:013:3
                DO BEGIN COMMENT THIS CODE INITIALIZES THE LABEL COUNTERS;               10918000 T  0125:013:0
                     EMITPAIR(LABELCTR[LABELCTRINX],SND)                                10919000 T  0125:013:2
                  END                                                                   10920000 T  0125:013:3
                UNTIL  LABELCTRINX←LABELCTRINX-1 < 0;                                   10921000 T  0125:013:0
                L←L-1;              EMITO(STD);         STEPIT;                         10922000 T  0125:013:1
                IF CHECK(COLON,418)                                                     10923000 T  0125:013:2
                THEN GO TO EXIT;                                                        10924000 T  0125:013:3
                 COMMENT ERROR 418 MEANS MISSING COLON IN DUMP DEC;                     10925000 T  0125:014:1
                FINALL←L;           L←TESTLOC;          STEPIT;                         10926000 T  0125:014:1
                IF (GT1 ← TABLE(I) ≠ NONLITNO AND GT1 ≠ LITNO                           10926500 T  0125:014:1
                   AND GT1 < REALID AND GT1 > INTID) OR (GT1 ← TABLE(I+1)               10926510 T  0125:014:1
                   ≠ COMMA AND GT1 ≠ SEMICOLON)                                         10926520 T  0125:014:0
```

```
                THEN BEGIN COMMENT ERROR 465-DUMP INDICATOR MUST BE          10926530 T  0125!015!0
                                UNSIGNED INTEGER OR SIMPLE VARIABLE;         10926540 T  0125!015!3
                           FLAG(465); GO TO EXIT;                            10926550 T  0125!015!3
                     END;                                                    10926560 T  0125!015!1
              PRIMARY;          EMITV(DUMPETEMP);                          10927000 T  0125!015!1
              EMITO(EQL);       EMITB(BFC,TESTLOC+6,LEXIT);                10928000 T  0125!015!0
              L←FINALL;         PUT(TAKE(GIT(ELBAT[I-3])) & DUMPLOC[       10929000 T  0125!015!1
              DUMPR],GIT(ELBAT[I-3]));                                     10930000 T  0125!015!3
              IF ELCLASS = COMMA                                           10931000 T  0125!016!0
              THEN BEGIN COMMENT GO AROUND ONE MORE TIME;                  10932000 T  0125!016!1
                        STEPIT;         GO TO START;                       10933000 T  0125!016!3
                 END;                                                      10934000 T  0125!016!1
              IF CHECK(SEMICOLON,419)                                      10935000 T  0125!016!1
              THEN;                                                        10936000 T  0125!016!2
               COMMENT ERROR 419 MEANS IMPROPER DUMP DEC DELIMITER;        10937000 T  0125!016!0
EXIT:;                                                                     10938000 T  0125!016!0
END DMUP;                                                                  10939000 T  0125!016!0
                                                          125 IS  170 LONG, NEXT SEG    3


COMMENT  CODE FOR SWITCHES IS COMPILED FROM TWO PLACES - IN SWITCHGEN       10940000 T  0003!087!2
         AND IN PURGE.  COMPLEX SWITCHES (I.E. SWITCHES CONTAINING         10941000 T  0003!087!2
         OTHER THAN LOCAL LABELS) ARE COMPILED HERE.  SIMPLE               10942000 T  0003!087!2
         SWITCHES ARE COMPILED AT PURGE TIME.  THIS IS FOR REASONS         10943000 T  0003!087!2
         OF EFFICIENCY.  IF A SWITCH IS ONLY CALLED ONE THE CODE           10944000 T  0003!087!2
         IS QUITE A BIT BETTER.  AFTER SWITCHGEN GOTOG IS TRUE IF          10945000 T  0003!087!2
         A COMMUNICATE MUST BE USED.  THE BLOCK ROUTINE MARKS SUCH         10946000 T  0003!087!2
         SWITCHES FORMAL.  THIS IS, OF COURSE, A FICTION.  FOR             10947000 T  0003!087!2
         SIMPLE SWITCHES SWITCHGEN LEAVES THE INDEX TO INFO IN EDOC        10948000 T  0003!087!2
         SO THAT PURGE CAN FIND THE LABELS.  IT SHOULD BE NOTED            10949000 T  0003!087!2
         THAT A SWITCH EXPECTS THE SWITCH INDEX TO BE FOUND IN             10950000 T  0003!087!2
         JUNK.  THE RESULT RETURNED BY SWITCHGEN IS WHETHER OR NOT         10951000 T  0003!087!2
         TO STUFF F INTO A SWITCH DESCRIPTOR, SINCE A SWITCH DE-           10952000 T  0003!087!2
         SCRIPTOR IS AN ACCIDENTAL ENTRY DESCRIPTOR;                       10953000 T  0003!087!2
      BOOLEAN PROCEDURE SWITCHGEN(BEFORE,PD);                %113-        10954000 P  0003!087!2
PRT(1066) = SWITCHGEN
          VALUE BEFORE; BOOLEAN BEFORE; REAL PD;             %113-        10954100 C  0003!087!2
            BEGIN                                                          10955000 T  0003!087!2
              LABEL LX,EXIT,BEF;                                           10956000 T  0003!087!2
                                                    START OF SEGMENT ********** 126
              REAL K,N,T1,TL;                                              10957000 T  0126!000!0
STACK(F+3) = K
STACK(F+4) = N
STACK(F+5) = T1
STACK(F+6) = TL
              TL ← L;                                                      10958000 T  0126!000!0
              EMIT(0); EMITV(JUNK); EMITO(GEQ); EMITV(JUNK);               10959000 T  0126!000!1
              L ← L+1; EMITO(GTR); EMITO(LOR); EMITV(JUNK);                10960000 T  0126!000!3
              EMITO(DUP); EMITO(ADD); COMMENT  WE HAVE GENERATED TEST      10961000 T  0126!000!2
                                        AND PREPARATION FOR SWITCH-JUMP;   10962000 T  0126!000!1
              GOTOG ← FALSE; COMMENT IF WE COMPILE JUMP OUT WE KNOW;        10963000 T  0126!000!1
              IF BEFORE THEN BEGIN STEPIT; GO TO BEF END;                  10964000 T  0126!000!3
         LX:  IF STEPI = LABELID AND ELBAT[I].LVL = LEVEL                  10965000 T  0126!001!2
                THEN BEGIN                                                 10966000 T  0126!001!0
                   INFO[0,N] ← ELBAT[I];                                   10967000 T  0126!001!2
                   IF N ← N+1 = 256                                        10968000 T  0126!001!3
```

```
                    THEN BEGIN ERR(147); GO TO EXIT END;                    10969000 T  0126:0018:1
                  IF STEP1 = COMMA THEN GO TO LX;                           10970000 T  0126:0021:2
                  EMIT0(BFC); L ← BUMPL; N ← N-1;                           10971000 T  0126:0022:1
                  FOR K ← 0 STEP 1 UNTIL N                                  10972000 T  0126:0026:0
                    DO BEGIN COMMENT  SAVE LINKS TO LABELS IN EDOC;         10973000 T  0126:0026:0
                     EMIT((GT1←INFO[0,K]),[35:1]);                          10974000 T  0126:0027:2
                     EMIT(GT1) END;                                         10975000 T  0126:0029:3
                  SWITCHGEN ← FALSE END                                     10976000 T  0126:0032:1
                ELSE BEGIN                                                  10977000 T  0126:0033:3
BEF:              L ← BUMPL; N ← N-1;                                       10978000 T  0126:0034:0
                  PUT(TAKE(LASTINFO)&(PD:=PROGDESCBLDR(ADES,TL,PD))         10978500 C  0126:0037:2
                       [16:37:11],LASTINFO); % GET PRT LOC AND SAVE         10978600 C  0126:0039:2
                  FOR K ← 0 STEP 1 UNTIL N                                  10979000 T  0126:0041:2
                    DO BEGIN COMMENT EMIT CODE FOR SIMPLE LABELS SEEN;      10980000 T  0126:0041:2
                     ADJUST; T1 ← L;                                        10981000 T  0126:0042:0
                     EMITL(GNAT(GT1←INFO[0,K]));                            10982000 T  0126:0043:2
                     GENGO(GT1);                                            10983000 T  0126:0046:0
                     INFO[0,K] ← T1;                                        10984000 T  0126:0046:1
                     EMIT0(RTS);                                            10985000 T  0126:0048:1
                     CONSTANTCLEAN END;                                     10986000 T  0126:0049:3
                  I ← I-1; N ← N+1;                                         10987000 T  0126:0052:0
                  DO BEGIN ADJUST;                                          10988000 T  0126:0054:1
                    STEPIT; INFO[0,N] ← L;                                  10989000 T  0126:0055:3
                    IF N ← N+1 = 256                                        10990000 T  0126:0058:0
                      THEN BEGIN ERR(147); GO TO EXIT END;                  10991000 T  0126:0059:2
                    DEXP; EMIT0(RTS) END                                    10992000 T  0126:0061:3
                    UNTIL ELCLASS ≠ COMMA; ADJUST;                          10993000 T  0126:0062:1
                    EMITB(BFW,TL+12,L); EMIT0(BFC);                         10994000 T  0126:0064:1
                    EMIT(0); EMIT0(RTS); N ← N-1;                           10995000 T  0126:0067:2
                    FOR K ← 0 STEP 1 UNTIL N                                10996000 T  0126:0069:3
                      DO EMITB(BBW,BUMPL,INFO[0,K]);                        10997000 T  0126:0069:3
                    SWITCHGEN ← TRUE END;                                   10998000 T  0126:0076:1
                  T1 ← L;                                                   10999000 T  0126:0077:2
                  L ← TL+4;                                                 11000000 T  0126:0078:0
                  EMITL(N+1);                                               11001000 T  0126:0079:2
                  L ← T1;                                                   12000000 T  0126:0080:1
              EXIT: END SWITCHGEN;                                          12001000 T  0126:0081:2
                                                                      126 IS    86 LONG, NEXT SEG    3


                PROCEDURE DBLSTMT;                                          12002000 T  0003:0877:2
                      BEGIN                                                 12003000 T  0003:0877:2
                      REAL S,T;                                             12004000 T  0003:0877:2
                                                                      START OF SEGMENT **********  127
STACK(F+2) = S
STACK(F+3) = T
                      BOOLEAN B ;                                           12004100 T  0127:0000:0
STACK(F+4) = B
                      LABEL L1,L2,L3,L4,EXIT ;                              12005000 T  0127:0000:0
                      S←0;                                                  12006000 T  0127:0000:0
                      IF STEPI≠LEFTPAREN THEN ERR(281)                      12007000 T  0127:0000:1
                      ELSE                                                  12008000 T  0127:0002:1
                  L1: BEGIN                                                 12009000 T  0127:0003:2
                        IF STEPI=COMMA THEN                                 12010000 T  0127:0004:0
                        BEGIN                                               12011000 T  0127:0005:2
```

```
                    DPTOG←TRUE;                                                    12012000 T  0127:0005:3
                    IF STEPI=ADOP THEN STEPIT;                                     12013000 T  0127:0006:0
                    EMITNUM(NLO);                                                  12014000 T  0127:0008:0
                    EMITNUM(IF ELBAT[I-1].ADDRESS =SUB THEN -NHI ELSE NHI);        12015000 T  0127:0009:2
                    DPTOG←FALSE;                                                   12016000 T  0127:0013:2
                    STEPIT;                                                        12017000 T  0127:0014:0
                         GO TO L2;                                                 12018000 T  0127:0014:1
                    END;                                                           12019000 T  0127:0015:2
                    IF TABLE(I+1)=COMMA THEN                                       12020000 T  0127:0015:2
                         BEGIN                                                     12021000 T  0127:0016:1
                         IF ELCLASS=ADOP  OR ELCLASS=MULOP THEN                    12022000 T  0127:0017:2
                              BEGIN                                                12023000 T  0127:0019:2
                              EMITO(ELBAT[I],ADDRESS+1);                           12024000 T  0127:0019:3
L4:                           IF (S←S-1)≤0 THEN FLAG(282);  STEPIT ;               12025000 T  0127:0021:3
                              GO TO L3;                                            12026000 T  0127:0025:3
                              END;                                                 12027000 T  0127:0026:0
                         IF ELCLASS=ASSIGNOP THEN                                  12028000 T  0127:0026:0
                              BEGIN                                                12029000 T  0127:0026:1
                              IF S←S-1<0 THEN FLAG(285);  T←0;  STEPIT ;           12030000 T  0127:0027:2
                              DO                                                   12031000 T  0127:0031:3
                              BEGIN                                                12032000 T  0127:0032:0
                          IF ELCLASS  ≠COMMA THEN BEGIN ERR(284);GO EXIT END;      12033000 T  0127:0032:0
                              STEPIT;                                              12034000 T  0127:0034:1
                              B←ELCLASS=INTID OR ELCLASS=INTARRAYID                12034100 T  0127:0035:2
                                              OR ELCLASS=INTPROCID ;               12034110 T  0127:0036:0
                              IF ELCLASS≤INTID AND ELCLASS≥REALID THEN             12035000 T  0127:0038:0
                              BEGIN  EMITN(ELBAT[I],ADDRESS); STEPIT END           12036000 T  0127:0040:0
                         ELSE IF ELCLASS≤INTPROCID AND ELCLASS≥REALPROCID THEN     12036100 T  0127:0042:1
                               IF ELBAT[I].LINK ≠ PROINFO.LINK THEN FLAG(211)      12036200 T  0127:0044:0
                              ELSE BEGIN EMITL(514); STEPIT END                    12036300 T  0127:0048:0
                         ELSE IF ELCLASS>INTARRAYID OR ELCLASS<REALARRAYID         12036400 T  0127:0050:0
                              THEN ERR(286)                                        12036500 T  0127:0051:3
                                                   ELSE VARIABLE(FL);              12037000 T  0127:0053:2
                              EMITO(IF B THEN ISD ELSE STD) END UNTIL T←T+1=2;      12038000 T  0127:0055:2
                              IF ELCLASS≠RTPAREN THEN GO L3 ;                      12039000 T  0127:0058:1
                              IF S≠0 THEN FLAG(283)                                12039100 T  0127:0060:1
                              ELSE BEGIN STEPIT; GO EXIT END ;                     12040000 T  0127:0062:0
                              END;                                                 12041000 T  0127:0064:0
                         IF ELCLASS=FACTOP THEN                                    12041100 T  0127:0064:0
                              BEGIN                                                12041110 T  0127:0065:2
                              EMITO(MKS);  EMITL(4);  EMITV(GNAT(POWERALL)) ;      12041130 T  0127:0065:3
                              EMITO(DEL);  EMITO(DEL);  GO L4 ;                    12041140 T  0127:0068:0
                              END ;                                                12041160 T  0127:0070:0
                         IF ELCLASS≤INTID AND ELCLASS≥BOOID THEN                   12042000 T  0127:0070:0
                              BEGIN                                                12043000 T  0127:0072:0
                              CHECKER(T←ELBAT[I]);                                 12044000 T  0127:0072:1
                              STEPIT;STEPIT;                                       12045000 T  0127:0074:0
                              AEXP;                                                12046000 T  0127:0075:2
                              EMITV(T.ADDRESS);                                    12047000 T  0127:0075:3
                              GO TO L2;                                            12048000 T  0127:0076:1
                              END;                                                 12049000 T  0127:0077:2
                         END    ;                                                  12050000 T  0127:0077:2
                              AEXP;                                                12051000 T  0127:0077:2
                              IF ELCLASS≠COMMA THEN BEGIN ERR(284);GO EXIT         12052000 T  0127:0077:3
                                                                     END;          12053000 T  0127:0080:0
                              STEPIT; AEXP; EMITO(XCH);                            12054000 T  0127:0080:0
L2:                      S←S+1;                                                   12055000 T  0127:0082:0
L3:                      IF ELCLASS≠COMMA THEN BEGIN ERR(284);GO TO EXIT END;     12056000 T  0127:0083:2
```

```
                    GO TO L1;                                                                    12057000 T  0127:0086:1
               EXIT:END                                                                          12058000 T  0127:0087:2
                    END  DBLSTMT;                                                               12059000 T  0127:0087:2
                                                                                              127 IS   90 LONG, NEXT SEG    3



          PROCEDURE CMPLXSTMT ;                                                                 12060000 T  0003:0877:2
               BEGIN                                                                            12060100 T  0003:0877:2
               REAL S,T ;                                                                       12060200 T  0003:0877:2
                                                                                          START OF SEGMENT **********  128
STACK(F+2) = S
STACK(F+3) = T
               BOOLEAN B ;                                                                      12060250 T  0128:0000:0
STACK(F+4) = B
               LABEL L1,L2,L3,L4,L5,EXIT,ERROR ;                                                12060300 T  0128:0000:0
               DEFINE ERRX(ERRX1) = BEGIN T←ERRX1; GO ERROR END #;                              12060400 T  0128:0000:0
               IF STEPI≠LEFTPAREN THEN ERRX(381) ;                                              12060500 T  0128:0000:0
          L1:  STEPIT ;                                                                         12060550 T  0128:0002:1
               IF TABLE(I+1)=COMMA THEN                                                         12060600 T  0128:0003:3
                    BEGIN                                                                       12060700 T  0128:0005:2
                    IF ELCLASS=ADOP THEN                                                        12060800 T  0128:0005:3
                         BEGIN                                                                  12060900 T  0128:0006:1
                         T←ELBAT[I],ADDRESS ;                                                   12061000 T  0128:0007:2
                         EMITPAIR(9,STD); EMITO(XCH); EMITV(9); EMITO(T) ;                      12061100 T  0128:0008:1
                         EMITPAIR(9,STD); EMITO(T); EMITV(9) ;                                  12061200 T  0128:0011:3
          L4:            IF S←S-1<1 THEN FLAG(382); STEPIT; GO L1 ;                             12061300 T  0128:0014:0
                         END ;                                                                  12061400 T  0128:0019:2
                    IF ELCLASS=MULOP THEN                                                       12061500 T  0128:0019:2
                         BEGIN                                                                  12061600 T  0128:0019:3
                         EMITO(MKS) ;                                                           12061700 T  0128:0020:0
                         EMITL(IF ELBAT[I],ADDRESS=MUL THEN 26 ELSE 35) ;                       12061725 T  0128:0021:2
                         EMITV(GNAT(SPECIALMATH)) ;                                             12061750 T  0128:0024:1
          L5:            EMITO(DEL); EMITO(DEL); GO L4 ;                                        12061800 T  0128:0025:3
                         END ;                                                                  12061900 T  0128:0028:0
                    IF ELCLASS=ASSIGNOP THEN                                                    12062000 T  0128:0028:0
                         BEGIN                                                                  12062100 T  0128:0028:1
                         IF S←S-1<0 THEN FLAG(385); T←0; STEPIT ;                               12062200 T  0128:0029:2
                         B←ELCLASS=INTID OR ELCLASS=INTPROCID                                   12062250 T  0128:0033:3
                                         OR ELCLASS=INTARRAYID ;                                12062255 T  0128:0034:1
                         DO   BEGIN                                                             12062300 T  0128:0036:1
                              IF ELCLASS≠COMMA THEN ERRX(384); STEPIT ;                         12062400 T  0128:0037:2
                              IF ELCLASS>BOOID AND ELCLASS<BOOARRAYID THEN                      12062500 T  0128:0040:0
                                   BEGIN EMITN(ELBAT[I],ADDRESS); STEPIT END                    12062600 T  0128:0041:3
                              ELSE IF ELCLASS>BOOPROCID AND ELCLASS<BOOID THEN                  12062700 T  0128:0044:0
                                        IF ELBAT[I],LINK≠PROINFO,LINK THEN FLAG(211)            12062800 T  0128:0046:1
                                        ELSE BEGIN EMITL(514); STEPIT END                       12062900 T  0128:0049:3
                                   ELSE IF ELCLASS<LABELID AND ELCLASS>BOOARRAYID               12063000 T  0128:0052:0
                                             THEN VARIABLE(FL)                                  12063100 T  0128:0053:3
                                        ELSE ERRX(386) ;                                        12063200 T  0128:0055:2
                              EMITO(IF B THEN ISD ELSE STD) ;                                   12063300 T  0128:0057:2
                              END                                                               12063400 T  0128:0059:3
                         UNTIL T←T+1=2 ;                                                        12063500 T  0128:0059:3
                         IF ELCLASS≠RTPAREN THEN GO L3 ;                                        12063600 T  0128:0061:3
                         IF S≠0 THEN FLAG(383) ELSE BEGIN STEPIT; GO EXIT END ;                 12063610 T  0128:0063:2
                         END ;                                                                  12063700 T  0128:0066:1
```

```
                IF ELCLASS=FACTOP THEN                                                  12063800 T  0128:0066:1
                     BEGIN                                                              12063900 T  0128:0067:2
                     EMITO(MKS); EMITL(8); EMITV(GNAT(POWERALL)); GO L5 ;               12064000 T  0128:0067:3
                     END ;                                                              12064100 T  0128:0071:2
                IF ELCLASS>BOOID AND ELCLASS<BOOARRAYID THEN                            12064200 T  0128:0071:2
                     BEGIN                                                              12064300 T  0128:0072:1
                     CHECKER(T←ELBAT[I]); STEPIT; STEPIT; AEXP ;                        12064400 T  0128:0073:2
                     EMITV(T,ADDRESS); GO L2 ;                                          12064500 T  0128:0076:0
                     END ;                                                              12064600 T  0128:0078:0
                END ;                                                                   12064700 T  0128:0078:0
           AEXP; IF ELCLASS≠COMMA THEN ERRX(384); STEPIT; AEXP; EMITO(XCH);             12064800 T  0128:0078:0
      L2: S←S+1 ;                                                                       12064900 T  0128:0082:1
      L3: IF ELCLASS=COMMA THEN GO L1; T←384 ;                                          12065000 T  0128:0084:0
      ERROR:                                                                            12065100 T  0128:0087:2
           ERR(T) ;                                                                     12065200 T  0128:0087:2
      EXIT:                                                                             12065300 T  0128:0087:3
           END OF CMPLXSTMT ;                                                           12065400 T  0128:0088:0
                                                                          128 IS   91 LONG, NEXT SEG    3


 REAL PROCEDURE FIXDEFINEINFO(T); VALUE T; REAL T;                                      12101000 T  0003:0877:2
      BEGIN REAL K,S,P,J,EL;                                                            12102000 T  0003:0877:2
                                                                        START OF SEGMENT **********  129

STACK(F+3) = K
STACK(F+4) = S
STACK(F+5) = P
STACK(F+6) = J
STACK(F+7) = EL
           MACROID←TRUE;                                                                12107000 T  0129:0000:0
           P ← GIT(FIXDEFINEINFO ← T);                                                  12108000 T  0129:0000:1
           STOPDEFINE←TRUE;                                                             12111000 T  0129:0002:1
           FL←TABLE(NXTELBT);                                                           12112000 T  0129:0003:2
           NXTELBT←NXTELBT-1;                                                           12113000 T  0129:0004:1
           IF EL≠LEFTPAREN AND EL≠LFTBRKET THEN                                         12114000 T  0129:0005:3
                FLAG(175) % [ OR ( EXPECTED.                                            12115000 T  0129:0007:3
           ELSE DO BEGIN J←J+1;                                                         12116000 T  0129:0008:1
                TEXT[NEXTTEXT.LINKR,NEXTTEXT.LINKC] := TAKE(P); %122-                   12117000 C  0129:0011:2
                NEXTTEXT := NEXTTEXT + 1;                       %122-                   12118000 C  0129:0014:1
                PUT(TAKE(P) & NEXTTEXT[11:32:16] &                                      12121000 T  0129:0016:0
                     DEFSTACKHEAD[35:35:13], P);                                        12122000 T  0129:0018:0
                DEFSTACKHEAD ← P.LINK;                                                  12123000 T  0129:0019:3
                P ← GIT(K←P);                                                           12123500 T  0129:0020:1
                DEFINEGEN(TRUE,0);                                                      12124000 T  0129:0022:1
           END UNTIL EL ← ELBAT[NXTELBT].CLASS ≠ COMMA OR K = P;                        12125000 T  0129:0023:3
           IF EL≠RTPAREN AND EL≠RTBRKET OR K≠P THEN                                     12126000 T  0129:0027:2
              FLAG(174);%INCORRECT # OF PARAMS IN DEFINE INVOCATION.                    12126100 T  0129:0030:0
           MACROID←FALSE;                                                               12127000 T  0129:0031:2
      END;                                                                              12128000 T  0129:0032:0
                                                                          129 IS   36 LONG, NEXT SEG    3


 PROCEDURE DEFINEPARAM(DINFO, N);                                                       12150000 T  0003:0877:2
 COMMENT  DEFINEPARAM GENERATES EVERYTHING (EXCEPT THE ELBAT                            12150100 T  0003:0877:2
          WORD) FOR AN INFO TABLE ENTRY FOR A PARAMETER OF A                            12150200 T  0003:0877:2
```

```
                    PARAMETRIC DEFINE.                                                                  12150300 T  0003:087?2
                ;                                                                                       12150400 T  0003:087?2
                    VALUE DINFO, N; INTEGER DINFO, N;                                                   12151000 T  0003:087?2
                BEGIN                                                                                   12152000 T  0003:087?2
                    INTEGER J;                                                                          12153000 T  0003:087?2
                                                                                                  START OF SEGMENT *********  130
STACK(F+2) = J
                STREAM PROCEDURE MAKEPARAM(INF, ACC, C, Q, N); VALUE C, Q, N;                           12154000 T  0130:000?0
PRT(1067) = MAKEPARAM
                BEGIN                                                                                   12155000 T  0130:000?0
                    DI ← ACC; DI ← DI+3;                                                                12156000 T  0130:000?0
                    SI ← INF; SI ← SI+3; DS ← C CHR;                                                    12157000 T  0130:0000?1
                    SI ← LOC N; SI ← SI+7; DS ← 1 CHR;                                                  12158000 T  0130:0001?3
                    SI ← LOC Q; SI ← SI+6; DS ← 1 CHR;                                                  12159000 T  0130:0002?0
                    DI ← ACC; DI ← DI+2; DS ← 1 CHR;                                                    12160000 T  0130:0003?2
                END MAKEPARAM;                                                                          12161000 T  0130:0003?3



                    ACCUM[1] ← 0;                                                                       12161500 T  0130:0004?0
                    MAKEPARAM(INFO[DINFO,LINKR,DINFO,LINKC], ACCUM[1],                                  12162000 T  0130:0005?2
                        J ← INFO[DINFO,LINKR,DINFO,LINKC].[12:6], 770+J, N);                            12163000 T  0130:0008?0
                    SCRAM ← ACCUM[1] MOD 125;                                                           12164000 T  0130:0013?2
                    COUNT ← J+2;                                                                        12165000 T  0130:0014?1
                END DEFINEPARAM;                                                                        12166000 T  0130:0015?3
                                                                                                  130 IS   18 LONG, NEXT SEG    3
```

```
                COMMENT THIS SECTION CONTAINS THE PROCEDURES USED BY THE BLOCK ROUTINE;                 13000000 T  0003:087?2
                PROCEDURE IODEC(IOT);                                                                   13001000 T  0003:087?2
PRT(1070) = IODEC
                    VALUE IOT;                                                                          13002000 T  0003:087?2
                    INTEGER IOT;                                                                        13003000 T  0003:087?2
                 BEGIN                                                                                  13004000 T  0003:087?2
                   STREAM PROCEDURE GET7(ACCUM,RESULT);                                                 13005000 T  0003:087?2
                                                                                                  START OF SEGMENT *********  131
PRT(1071) = GET7
                  BEGIN                                                                                 13006000 T  0131:000?0
                    LOCAL T,T1; LABEL EXIT;                                                             13007000 T  0131:000?0
                    SI←ACCUM; SI←SI+2; DI← LOC T; DI←DI+7;                                              13008000 T  0131:000?0
                    DS← CHR; T1←SI; SI←LOC T; SI←SI+7;                                                  13009000 T  0131:0001?2
                    DI← RESULT; DS← 8 LIT "0       "; DI←DI-7;                                          13010000 T  0131:0002?0
                    IF SC≥"0"                                                                           13011000 T  0131:0003?3
                       THEN IF  SC<"8"                                                                  13012000 T  0131:0004?0
                                THEN                                                                    13013000 T  0131:0005?2
                                  BEGIN                                                                 13014000 T  0131:0005?2
                                    SI←T1; DS← T CHR;GO TO EXIT                                         13015000 T  0131:0005?2
                                  END;                                                                  13016000 T  0131:0006?1
                    SI←T1;DS ← 7 CHR;                                                                   13017000 T  0131:0006?1
                    EXIT;                                                                               13018000 T  0131:0007?2
                  END;                                                                                  13019000 T  0131:0007?2
```

```
                    STREAM PROCEDURE ENTERID(IDLOC,FILENO,TYPE,MULFID,FILID,FILID1,N);                13020000 T  0131:0008:0
PRT(1072) = ENTERID
                       VALUE FILENO,TYPE,MULFID,FILID,N;                                            13021000 T  0131:0008:0
                     BEGIN                                                                          13022000 T  0131:0008:0
                       DI←IDLOC; DI←DI+5;DI←DC;                                                     13023000 T  0131:0009:2
                       SI←LOC FILENO; SI←SI+6;                                                      13024000 T  0131:0009:3
                       DS← 2 CHR; SI←LOC TYPE;                                                      13025000 T  0131:0010:0
                       SI← SI+7; DS← CHR;                                                           13026000 T  0131:0010:1
                       SI← LOC MULFID;SI←SI+1;                                                      13027000 T  0131:0011:2
                       DS← 7 CHR;                                                                   13028000 T  0131:0011:3
                       SI← LOC FILID;SI←SI+1;                                                       13029000 T  0131:0012:0
                       DS← 7 CHR;                                                                   13030000 T  0131:0012:1
                       SI← LOC N; SI←SI+7;                                                          13031000 T  0131:0012:1
                       DS← CHR;                                                                     13032000 T  0131:0013:2
                       SI← FILID1; SI←SI+3;                                                         13033000 T  0131:0013:3
                       DS← N CHR; N←DI;                                                             13034000 T  0131:0014:0
                       DI← IDLOC; DI←DI+5;                                                          13035000 T  0131:0014:1
                       SI← LOC N; SI←SI+5;                                                          13036000 T  0131:0015:2
                       DS← 3 CHR                                                                    13037000 T  0131:0015:3
                     END;                                                                           13038000 T  0131:0016:0


                     REAL  MULFID,FILID,TYPE,M,K;                                                   13039000 T  0131:0016:0
STACK(F+2) = MULFID
STACK(F+3) = FILID
STACK(F+4) = TYPE
STACK(F+5) = M
STACK(F+6) = K
                     DEFINE CALL5=EMITL(0); EMITL(IOT); EMITL(8); EMITV(5) # ;                      13039400 T  0131:0016:0
                     REAL SAVADDRSF ;                                                               13039450 T  0131:0016:0
STACK(F+7) = SAVADDRSF

                          REAL ACCUM1;                                                              13039500 T  0131:0016:0
STACK(F+10) = ACCUM1
                     INTEGER CURRENT, IOTEMP, IOTEMPO;                                              13039550 T  0131:0016:0
STACK(F+11) = CURRENT
STACK(F+12) = IOTEMP
STACK(F+13) = IOTEMPO
                LABEL START;                                                                        13040000 T  0131:0016:0
                JUMP(CHKX;                                                                          13041000 T  0131:0016:0
                IF G←GTA1[J←J-1]=FILEV                                                              13042000 T  0131:0017:3
                   THEN                                                                             13043000 T  0131:0019:3
                    BEGIN                                                                           13044000 T  0131:0020:0
                    IF G←GTA1[J←J-1]=SWITCHV                                                        13045000 T  0131:0020:1
                      THEN                                                                          13046000 T  0131:0022:1
                           BEGIN                                                                    13047000 T  0131:0023:2
                                STOPENTRY←NOT SPECTOG;                                              13048000 T  0131:0023:3
                                ENTRY(SUPERFILEID);                                                 13049000 T  0131:0024:1
                                IF SPECTOG THEN GO TO START;                                        13050000 T  0131:0025:2
                                IF ELCLASS≠ASSIGNOP                                                 13051000 T  0131:0026:0
                                  THEN FLAG(34);                                                    13052000 T  0131:0026:1
                                EMITO(MKS);                                                         13053000 T  0131:0028:0
                                CHECKDISJOINT(ADDRSF);                                              13054000 T  0131:0029:2
                                G←L;L←L+1;                                                          13055000 T  0131:0029:3
                                EMITL(1);                                                           13056000 T  0131:0031:3
                                EMITL(1);                                                           13057000 T  0131:0032:1
                           EMITL(1);                                                                13058000 T  0131:0033:2
```

```
      EMITV(5) ;                                                                                                    13059000 T  0131:003:0
                                                                                                                    13060000 T  0131:003:1
           J←-1; STOPENTRY←FALSE;                                                                                   13061000 T  0131:003:1
           DO                                                                                                       13062000 T  0131:003:1
              BEGIN                                                                                                 13063000 T  0131:003:2
                    IF STEPI ≠ FILFID%                                                                              13064000 T  0131:003:2
                 THEN FLAG(35);                                                                                     13065000 T  0131:003:3
                   PASSFILE;                                                                                        13066000 T  0131:003:2
                   EMITL(J←J+1);                                                                                    13067000 T  0131:003:3
                   EMITN(ADDRSF);                                                                                   13068000 T  0131:004:3
                   EMITO(STD);                                                                                      13069000 T  0131:004:0
              END                                                                                                   13070000 T  0131:004:2
           UNTIL ELCLASS≠COMMA;                                                                                     13071000 T  0131:004:2
           GT2←L;L←G; EMITL(J+1); L←GT2; GO TO START                                                                13072000 T  0131:004:0
      END;                                                                                                          13073000 T  0131:004:0
  I←I-1;M←1;                                                                                                        13074000 T  0131:004:0
IF G=ALFAV                                                                                                          13075000 T  0131:005:0
      THEN M←0                                                                                                      13076000 T  0131:005:1
   ELSE J←J+1;K←J; STOPENTRY←NOT SPECTOG;                                                                           13077000 T  0131:005:3
 DO                                                                                                                 13078000 T  0131:005:3
  BEGIN                                                                                                             13079000 T  0131:005:0
    STOPDEFINE := TRUE ;                                                                                            13079500 T  0131:005:0
 STEPIT; J:=K;P2:=P3:=P4:=FALSE;GTA1[0]:=0; GET7(ACCUM[1],FILID);                                                    13080000 T  0131:005:1
    MULFID←0;TYPE←2; ENTER(FILEID);                                                                                 13081000 T  0131:006:1
    SAVADDRSF←ADDRSF ;                                                                                              13081500 T  0131:006:2
    IF SPECTOG THEN GO TO START;                                                                                    13082000 T  0131:006:0
       EMITO(MKS);EMITL(0);EMITL(0);                                                                                13082500 T  0131:006:2
    IF ELCLASS=LITNO                                                                                                13083000 T  0131:006:2
     THEN                                                                                                           13084000 T  0131:006:3
       BEGIN                                                                                                        13085000 T  0131:007:0
         TYPE←ELBAT[I],ADDRESS;                                                                                     13086000 T  0131:007:1
         STEPIT                                                                                                     13087000 T  0131:007:0
       END;                                                                                                         13088000 T  0131:007:0
       IF ELCLASS≤IDMAX THEN                                                                                        13088010 T  0131:007:1
           BEGIN                                                                                                    13088020 T  0131:007:2
% TO VOID A CARD                                                                                                    13088025 T  0131:007:3
                IF ACCUM1←ACCUM[1]="5PRINT" THEN TYPE←1 ELSE                                                        13088030 T  0131:007:3
                IF ACCUM1="6REMOT" THEN TYPE ← 19 ELSE                                                              13088035 T  0131:007:1
                IF ACCUM1="5PUNCH" THEN TYPE←0 ELSE                                                                 13088040 T  0131:008:0
                IF ACCUM1="4DISK0" THEN BEGIN STOPDEFINE←TRUE ;                                                     13088050 T  0131:008:0
  TYPE:=12; IF STEPI ≤ IDMAX THEN BEGIN IF ACCUM1 := ACCUM[1] ≠                                                     13088060 T  0131:008:0
    "6SERIA" THEN BEGIN IF ACCUM1 = "6RANDO" THEN                                                                   13088070 T  0131:009:2
    TYPE:=TYPE+2 ELSE IF ACCUM1 = "6UPDAT" THEN                                                                     13088075 T  0131:009:2
    TYPE:=TYPE+1 ELSE                                                                                               13088080 T  0131:009:3
    IF ACCUM1="7PROTE" THEN TYPE←26 ELSE                                                                            13088082 T  0131:010:3
  FLAG(43);END;STEPIT;            END;  IF ELCLASS=LFTBRKET THEN                                                    13088085 T  0131:010:2
  BEGIN STEPIT;L←L-2;AEXP;IF ELCLASS=COLON THEN BEGIN                                                               13088090 T  0131:010:2
       STEPIT; AEXP END ELSE FLAG(30);                                                                              13088100 T  0131:011:2
        IF ELCLASS≠RTBRKET THEN FLAG(44);END ELSE I←I-1;                                                            13088105 T  0131:011:2
                    END;                                                                                            13088110 T  0131:011:2
          STEPIT                                                                                                    13088120 T  0131:011:2
          END;                                                                                                      13088130 T  0131:011:2
     IF ELCLASS=STRNGCON THEN                                                                                       13089000 T  0131:011:3
        BEGIN                                                                                                       13090000 T  0131:012:0
          GET7(ACCUM[1],G);                                                                                         13091000 T  0131:012:1
          IF STEPI=STRNGCON                                                                                         13092000 T  0131:012:1
            THEN                                                                                                    13093000 T  0131:012:2
```

```
                    BEGIN                                                                                    13094000 T  0131!0123!3
                      GET7(ACCUM[1],FILID);                                                                  13095000 T  0131!0124!0
                  STEPIT;                                                                                    13096000 T  0131!0126!0
                      MULFID←G                                                                               13097000 T  0131!0126!1
                    END                                                                                      13098000 T  0131!0126!1
                  ELSE                                                                                       13099000 T  0131!0127!3
                    FILID←G;                                                                                 13100000 T  0131!0127!3
          END;                                                                                               13101000 T  0131!0128!1
IF MKABS(IDARRAY[127]) = IDLOC.[33:15] < (26 + KOUNT).[41:4]                                                 13101100 T  0131!0128!1
THEN FLAG(040) ELSE                                                                                          13101200 T  0131!0132!1
ENTERID(IDLOC,FILENO          ,TYPE,MULFID,FILID,INFO[(LASTINFO+1),                                          13102000 T  0131!0134!0
       LINKR,(LASTINFO+1),LINKC],KOUNT);                                                                     13103000 T  0131!0136!1
IF ELCLASS≠LEFTPAREN                                                                                         13104000 T  0131!0140!0
  THEN FLAG(26);                                                                                             13105000 T  0131!0140!0
      ARRAYFLAG←BOOLEAN(3) ;                                                                                 13106000 T  0131!0142!0
                                                                                                             13107000 T  0131!0142!1
          EMITL(REAL(NOT(P2 OR P3)),[46:2]);                                                                 13108000 T  0131!0142!1
              EMITL(FILENO);FILENO←FILENO+1;                                                                 13109000 T  0131!0144!1
      CHECKDISJOINT(TAKE(LASTINFO),ADDRESS);                                                                 13110000 T  0131!0146!1
      STEPIT;AEXP;EMITL(M);                                                                                  13111000 T  0131!0148!1
      COMMENT GUESS AT THE NO. OF BUFFERS DECLARED;                                                          13112000 T  0131!0150!0
          IF (IOTEMP←GET(L-2)).[46:2] = 0 THEN CURRENT ← IOTEMP                                              13113000 T  0131!0150!0
                        DIV 4  ELSE  CURRENT←2;                                                              13114000 T  0131!0153!3
                                                                                                             13115000 T  0131!0156!0
                                                                                                             13116000 T  0131!0156!0
                                                                                                             13117000 T  0131!0156!0
                                                                                                             13118000 T  0131!0156!0
                                                                                                             13119000 T  0131!0156!0
                                                                                                             13120000 T  0131!0156!0
                                                                                                             13121000 T  0131!0156!0
                                                                                                             13122000 T  0131!0156!0
                                                                                                             13123000 T  0131!0156!0
                                                                                                             13124000 T  0131!0156!0
                                                                                                             13125000 T  0131!0156!0
                                                                                                             13126000 T  0131!0156!0
                                                                                                             13127000 T  0131!0156!0
                                                                                                             13128000 T  0131!0156!0
                                                                                                             13129000 T  0131!0156!0
                                                                                                             13130000 T  0131!0156!0
                                                                                                             13131000 T  0131!0156!0
                                                                                                             13132000 T  0131!0156!0
                                                                                                             13133000 T  0131!0156!0
                                                                                                             13134000 T  0131!0156!0
                                                                                                             13135000 T  0131!0156!0
                                                                                                             13136000 T  0131!0156!0
                                                                                                             13137000 T  0131!0156!0
                                                                                                             13138000 T  0131!0156!0
                                                                                                             13139000 T  0131!0156!0
                                                                                                             13140000 T  0131!0156!0
                                                                                                             13141000 T  0131!0156!0
                                                                                                             13142000 T  0131!0156!0
                                                                                                             13143000 T  0131!0156!0
                                                                                                             13144000 T  0131!0156!0
    IF ELCLASS ≠ COMMA THEN BEGIN FLAG(27); GO TO START END                                                   13145000 T  0131!0156!0
FLSE                                                                                                         13146000 T  0131!0158!1
  BEGIN                                                                                                      13147000 T  0131!0158!1
    STEPIT; AEXP;                                                                                            13148000 T  0131!0159!2
                                                                                                             13148100 T  0131!0160!0
```

```
IF (IOTEMP←GET(L-1)).[46:2] = 0 THEN IOTEMP←IOTEMP DIV 4                13148200 T  0131:0160:0
                ELSE IOTEMP← 256;                                       13148300 T  0131:0163:3
IF ELCLASS≠COMMA                                                        13149000 T  0131:0165:3
  THEN                                                                  13150000 T  0131:0166:0
     BEGIN                                                              13151000 T  0131:0166:1
       EMITL(0);                                                        13152000 T  0131:0167:2
       CALL5 ;                                                          13153000 T  0131:0167:3
     END                                                                13154000 T  0131:0170:1
  ELSE                                                                  13155000 T  0131:0170:1
     BEGIN                                                              13156000 T  0131:0170:1
     IF GT1←FILEATTRIBUTEINDX(FALSE)=0 THEN                             13157000 T  0131:0171:2
          BEGIN  AEXP;                                                  13157010 T  0131:0173:2
                 IF (IOTEMP0:=GET(L-1)).[46:2] = 0 THEN                 13157020 T  0131:0174:0
                 IF IOTEMP0 DIV 4 > IOTEMP THEN                         13157030 T  0131:0177:2
                    IOTEMP:=IOTEMP0 DIV 4; CALL5; END                   13157040 T  0131:0178:1
     ELSE BEGIN                                                         13158000 T  0131:0183:3
          EMITL(0); CALL5; EMITO(MKS); EMITL(5) ;                       13159000 T  0131:0184:0
          EMITN(SAVADDRSF); GT1←FILEATTRIBUTEHANDLER(FIO);              13160000 T  0131:0189:2
          END ;                                                         13161000 T  0131:0190:1
     WHILE ELCLASS=COMMA DO                                             13162000 T  0131:0191:2
          IF GT1←FILEATTRIBUTEINDX(TRUE)=0 THEN                         13163000 T  0131:0193:2
               BEGIN ERR(291); GO START END                             13164000 T  0131:0195:2
          ELSE BEGIN                                                    13165000 T  0131:0197:2
               EMITO(MKS); EMITL(5); EMITN(SAVADDRSF) ;                 13166000 T  0131:0197:3
               GT1←FILEATTRIBUTEHANDLER(FIO) ;                          13167000 T  0131:0199:3
               END ;                                                    13168000 T  0131:0201:2
                  END ;                                                 13169000 T  0131:0201:3
             END ;                                                      13170000 T  0131:0201:3
          ARRAYFLAG←FALSE ;                                             13181000 T  0131:0201:3
       IF ELCLASS≠RTPAREN THEN FLAG(29);                                13182000 T  0131:0202:0
          COMMENT TOTAL UP THE BUFFER REQ. PER FILE DECLARATION;         13183000 T  0131:0204:0
              IOBUFFSIZE←IOBUFFSIZE + 50 + ( CURRENT × IOTEMP);         13184000 T  0131:0204:0
% VOID                                                                  13185000 T  0131:0206:1
% VOID                                                                  13186000 T  0131:0206:1
          END                                                           13187000 T  0131:0206:1
      UNTIL STEPI≠COMMA;                                                13188000 T  0131:0206:1
     STOPENTRY←FALSE;                                                   13189000 T  0131:0208:0
     END ELSE                                                           13190000 T  0131:0208:1
       BEGIN                                                            13191000 T  0131:0208:1
         IF G≠FORMATV THEN FLAG(33) ELSE                                13192000 T  0131:0209:2
      IF SPECTOG THEN ENTRY(FRMTID+REAL(GTA1[J-1]=SWITCHV))ELSE FORMATGEN 13193000 T  0131:0211:2
      END;                                                              13194000 T  0131:0215:2
   START;                                                               13195000 T  0131:0216:0
   END;                                                                 13196000 T  0131:0216:0
                                                                  131 IS  221 LONG, NEXT SEG    3


                PROCEDURE HANDLESWLIST;                                 13196300 T  0003:0877:2
PRT(1073) = HANDLESWLIST
                BEGIN                                                   13196310 T  0003:0877:2
                     LABEL  OVER;                                       13196320 T  0003:0877:2
                                                                START OF SEGMENT **********  132
                     JUMPCHKX;                                          13196330 T  0132:0000:0
                     STOPENTRY← NOT SPECTOG;                            13196340 T  0132:0000:0
                                                                        13196350 T  0132:0000:1
```

```
                ENTRY(SUPERLISTID);                                          13196360 T  0132:0001:3
                IF SPECTOG THEN GO TO OVER;                                  13196370 T  0132:0002:0
                IF ELCLASS ≠ ASSIGNOP THEN FLAG(41);                         13196380 T  0132:0003:2
                   COMMENT MISSING ←;                                        13196390 T  0132:0005:2
                EMITO(MKS);                                                  13196400 T  0132:0005:2
                CHECKDISJOINT(ADDRSF);                                       13196410 T  0132:0006:0
                G←L; L←L+1;                                                  13196420 T  0132:0006:1
                EMITL(1);                                                    13196430 T  0132:0008:1
                EMITL(1);                                                    13196440 T  0132:0009:3
          EMITL(1);                                                          13196450 T  0132:0010:0
          EMITV(5); COMMENT CREATE AN ARRAY TO HOLD                          13196460 T  0132:0011:2
                          LIST DESCRIPTORS FOR SWITCH LIST;                  13196470 T  0132:0011:3
COMMENT USED TO USE  EMITN(XITR),  DOESN"T ANYMORE;                          13196480 T  0132:0011:3
          J←-1; STOPENTRY ← FALSE;                                           13196490 T  0132:0011:3
     DO                                                                      13196500 T  0132:0013:3
          BEGIN                                                              13196510 T  0132:0014:0
                IF STEP1 ≠ LISTID AND ELCLASS ≠ SUPERLISTID                  13196520 T  0132:0014:0
                THEN BEGIN ERR(42); GO TO OVER END;                          13196530 T  0132:0015:2
                PASSLIST;                                                    13196540 T  0132:0017:3
                EMITL(J←J+1);                                                13196550 T  0132:0018:0
                EMITN(ADDRSF);                                               13196560 T  0132:0020:0
                EMITO(STD);    COMMENT STORE LIST DESC IN ARRAY;             13196570 T  0132:0020:1
          END                                                                13196580 T  0132:0021:3
                UNTIL  ELCLASS ≠ COMMA;                                      13196590 T  0132:0021:3
   GT2←L; L←G; EMITL(J+1); L←GT2;                                            13196600 T  0132:0022:1
 OVER;   END OF  HANDLESWLIST;                                               13196610 T  0132:0026:0
                                                                  132 IS   28 LONG, NEXT SEG    3
```

```
                PROCEDURE SCATTERELBAT;                                      13197000 T  0003:0877:2
                     BEGIN                                                   13198000 T  0003:0877:2
                          REAL T;                                            13199000 T  0003:0877:2
                                                                  START OF SEGMENT **********  133
STACK(F+2) = T
                          T ← ELBAT[I];                                      13200000 T  0133:0000:0
                          KLASSF ← T.CLASS;                                  13201000 T  0133:0001:2
                          FORMALF ← BOOLEAN(T.FORMAL);                       13202000 T  0133:0002:0
                          VONF ← BOOLEAN(T.VO);                              13203000 T  0133:0003:3
                          LEVELF ← T.LVL;                                    13204000 T  0133:0004:1
                          ADDRSF ← T.ADDRESS;                                13205000 T  0133:0006:0
                          INCRF ← T.INCR;                                    13206000 T  0133:0007:2
                          LINKF ← T.LINK;                                    13207000 T  0133:0008:1
                     END  SCATTERELBAT;                                      13208000 T  0133:0009:3
                                                                   133 IS   12 LONG, NEXT SEG    3
```

```
                PROCEDURE CHKSOB;                                            13209000 T  0003:0877:2
PRT(1074) = CHKSOB
                     IF GTA1[J←J-1]≠0 THEN FLAG(23);                         13210000 T  0003:0877:2
```

```
                DEFINE SUBOP=48#,                                            13211000 T  0003:0881:3
```

```
                    ADDC=532480#,                                                    13212000 T  0003:088:3
                    SUBC=1581056#,                                                   13213000 T  0003:088:3
                    EMITSTORE=EMITPAIR#;                                             13214000 T  0003:088:3
                PROCEDURE PURGE(STOPPER);                                            13215000 T  0003:088:3
                    VALUE STOPPER;                                                   13216000 T  0003:088:3
                    REAL STOPPER;                                                    13217000 T  0003:088:3
                BEGIN                                                                13218000 T  0003:088:3
                  INTEGER POINTER;                                                   13219000 T  0003:088:3
                                                                                  START OF SEGMENT **********  134
STACK(F+2) = POINTER
                  LABEL RECOV; DEFINE ELCLASS = KLASSF#;                             13220000 T  0134:000:0
                  REAL J,N,OCR,TL,ADD;                                               13221000 T  0134:000:0
STACK(F+3) = J
STACK(F+4) = N
STACK(F+5) = OCR
STACK(F+6) = TL
STACK(F+7) = ADD
                   POINTER←LASTINFO;                                                 13222000 T  0134:000:0
                  WHILE POINTER ≥ STOPPER                                            13223000 T  0134:000:1
                  DO                                                                 13224000 T  0134:000:2
                    BEGIN                                                            13225000 T  0134:002:0
                        IF ELCLASS←(GT1←TAKE(POINTER)).CLASS=NONLITNO                13226000 T  0134:002:0
                        THEN BEGIN                                                   13227000 T  0134:000:1
                        NCII←NCII-1;                                                 13228000 T  0134:000:3
                            EMITNUM(TAKE(POINTER+1));                                13229000 T  0134:000:1
                            EMITSTORE(MAXSTACK,STD);                                 13230000 T  0134:000:1
                            MAXSTACK←(G←MAXSTACK)+1;                                 13231000 T  0134:000:3
                            J←L; L←GT1.LINK;                                         13232000 T  0134:001:2
                        DO                                                           13233000 T  0134:001:2
                          BEGIN                                                      13234000 T  0134:001:0
                             GT4←GET(L);                                             13235000 T  0134:001:0
                             EMITV(G)                                                13236000 T  0134:001:2
                          END                                                        13237000 T  0134:001:3
                        UNTIL (L←GT4)=4095;                                          13238000 T  0134:001:0
                        L←J;                                                         13239000 T  0134:001:3
                        POINTER←POINTER-GT1.INCR                                     13240000 T  0134:001:1
                        END                                                          13241000 T  0134:001:1
ELSE                                                                                 13242000 T  0134:002:0
BEGIN                                                                                13243000 T  0134:002:0
IF NOT BOOLEAN(GT1,FORMAL)                                                           13244000 T  0134:002:0
  THEN BEGIN                                                                         13245000 T  0134:002:0
                        IF ELCLASS = LABELID                                         13246000 T  0134:002:3
                          THEN BEGIN                                                 13247000 T  0134:002:3
                            ADD ← GT1.ADDRESS;                                       13248000 T  0134:002:1
                            IF NOT BOOLEAN(OCR←TAKE(GIT(POINTER))),[1:1]             13249000 T  0134:002:0
                               THEN IF OCR.[36:12] ≠ 0 OR ADD ≠ 0                    13250000 T  0134:002:0
                                  THEN BEGIN GT1 ← 160; GO TO RECOV END;             13251000 T  0134:003:1
                            IF ADD ≠ 0 THEN GT1←PROGDESCBLDR(2,OCR,ADD) END           13252000 T  0134:003:2
                        ELSE IF ELCLASS = SWITCHID                                   13253000 T  0134:003:0
                          THEN BEGIN                                                 13254000 T  0134:003:1
                             IF TAKE(POINTER+1) < 0                                  13255000 T  0134:003:3
                               THEN BEGIN GT1 ← 162; GO TO RECOV END;                13256000 T  0134:003:2
                             OCR ←(J ← TAKE(GIT(POINTER))).[24:12];                  13257000 T  0134:004:2
                             N ← GET( (J←J.[36:12])+4); TL ← L;                      13258000 T  0134:004:0
                             IF ADD ← GT1.ADDRESS ≠ 0                                13259000 T  0134:004:3
                              THEN BEGIN                                             13260000 T  0134:004:1
                                GT5 ← PROGDESCBLDR(0,J,ADD);                         13261000 T  0134:004:3
```

```
                    IF OCR ≠ 0                                                  13262000 T  0134:005:3
            THEN BEGIN L←OCR-2; CALLSWITCH(POINTER); EMITO(BFW);END;            13263000 T  0134:005:3
                    L←J+11; EMITL(15); EMITO(RTS);                              13264000 T  0134:005:3
                    FOR J ← 4 STEP 4 UNTIL N                                    13265000 T  0134:005:0
                      DO BEGIN                                                  13266000 T  0134:005:0
                        EMITL(GNAT(GET(L)×4096+GET(L+1)));                      13267000 T  0134:005:2
                        EMITO(RTS) END END                                      13268000 T  0134:006:1
                  ELSE BEGIN                                                    13269000 T  0134:006:3
                    L ← J+13;                                                   13270000 T  0134:006:0
                    FOR J ← 4 STEP 4 UNTIL N                                    13271000 T  0134:006:2
                      DO BEGIN                                                  13272000 T  0134:006:2
                        GT1 ← GET(L)×4096+GET(L+1);                             13273000 T  0134:007:0
                        GOGEN(GT1,BFW) END;END;                                 13274000 T  0134:007:2
                                                                                13275000 T  0134:007:1
                                                                                13276000 T  0134:007:1
                L ← TL END                                                      13277000 T  0134:007:1
              ELSE IF ELCLASS ≥ PROCID AND ELCLASS ≤ INTPROCID                  13278000 T  0134:007:2
               THEN IF TAKE(POINTER+1) < 0                                      13279000 T  0134:008:0
                 THEN BEGIN GT1 ← 161;                                          13280000 T  0134:008:1
RECOV:             MOVE(9,INFO[POINTER.LINKR,POINTER.LINKC],ACCUM);            13281000 T  0134:008:0
                   Q ← ACCUM[1]; FLAG(GT1); ERRORTOG ← TRUE END                 13282000 T  0134:008:1
        END;                                                                    13283000 T  0134:009:2
    XREFDUMP(POINTER); % DUMP XREF INFO                              %116-      13283500 C  0134:009:2
    GT2←TAKE(POINTER+1);                                                        13284000 T  0134:009:2
    GT3←GT2.PURPT;                                                              13285000 T  0134:009:2
STACKHEAD[(0&GT2[12:12:36])MOD 125]←TAKE(POINTER).LINK;                         13286000 T  0134:009:0
    POINTER←POINTER-GT3                                                         13287000 T  0134:010:0
        END                                                                     13288000 T  0134:010:0
  END ;                                                                         13289000 T  0134:010:2
    LASTINFO←POINTER;                                                           13290000 T  0134:010:3
    NEXTINFO←STOPPER                                                            13291000 T  0134:010:2
  END;                                                                          13292000 T  0134:010:2
                                                                        134 IS  107 LONG, NEXT SEG    3


                PROCEDURE F;                                                    13293000 T  0003:088:3
                COMMENT                                                         13294000 T  0003:088:3
                      F IS THE PROCEDURE WHICH PLACES AN ENTRY IN INFO AND      13295000 T  0003:088:3
                      HOOKS IT INTO STACKHEAD. THE PREVIOUS STACKHEAD LINK      13296000 T  0003:088:3
                      IS SAVED IN THE LINK OF THE ELBAT WORD IN THE NEW ENTRY    13297000 T  0003:088:3
                      F PREVENTS AN ENTRY FROM OVERFLOWING A ROW,STARTING AT THE 13298000 T  0003:088:3
                      BEGINNING OF THE NEXT ROW IF NECESSARY            ;       13299000 T  0003:088:3
                 BEGIN                                                          13300000 T  0003:088:3
                   REAL WORDCOUNT,RINX;                                         13301000 T  0003:088:3
                                                                       START OF SEGMENT ********** 135
STACK(F+2) = WORDCOUNT
STACK(F+3) = RINX
                      IF RINX←(NEXTINFO+WORDCOUNT←(COUNT+18)DIV 8    ).LINKR ≠  13302000 T  0135:000:0
                           NEXTINFO.LINKR                                       13303000 T  0135:000:2
                  THEN BEGIN PUT(0&(RINX×256-NEXTINFO)[27:40:8],NEXTINFO);      13304000 T  0135:000:2
                       NEXTINFO←256×RINX END;                                   13305000 T  0135:000:7
                       IF SPECTOG THEN                                          13305100 T  0135:000:9
                       IF NOT MACROID THEN                                      13305200 T  0135:000:9
                            UNHOOK;                                             13305300 T  0135:001:0
                       KOUNT←COUNT;                                             13306000 T  0135:001:2
```

```
          ACCUM[0].INCR←WORDCOUNT;                                                                    13307000 T  0135:0012:0
           ACCUM[0].LINK ←STACKHEAD[SCRAM];STACKHEAD[SCRAM]←NEXTINFO;                                 13308000 T  0135:0014:1
           ACCUM[1].PURPT←NEXTINFO-LASTINFO;                                                          13309000 T  0135:0018:1
MOVE(WORDCOUNT,ACCUM,INFO[NEXTINFO.LINKR,NEXTINFO.LINKC]);                                            13310000 T  0135:0021:3
          IF XREF THEN % MAKE DECLARATION REFERENCE                                        %116-      13310050 C  0135:0025:2
           IF (ACCUM[0].CLASS ≠ DEFINEDID OR NOT                                           %116-      13310075 C  0135:0026:0
               BOOLEAN(ACCUM[0].FORMAL)) THEN % NOT DEFINE PARAMETER                       %116-      13310080 C  0135:0028:0
            BEGIN                                                                          %116-      13310100 P  0135:0029:3
              XREFINFO[NEXTINFO] :=                                                        %116-      13310200 P  0135:0030:0
                IF SPECTOG THEN                                                            %116-      13310300 P  0135:0032:1
                  XREFINFO[ELBAT[I]]                                                       %116-      13310350 C  0135:0033:2
                ELSE                                                                       %116-      13310400 P  0135:0036:1
                  ((XLUN := XLUN + 1) & SGNO SEGNOF);                                      %116-      13310450 P  0135:0037:2
              IF SPECTOG THEN % JUST GO BACK AND FIX UP XREF ENTRY                         %116-      13310500 P  0135:0040:0
                XMARK(DECLREF)                                                             %116-      13310525 C  0135:0040:0
              ELSE                                                                         %116-      13310550 C  0135:0045:2
                XREFIT(NEXTINFO,CARDNUMBER,IF PTOG AND NOT STREAMTOG                       %116-      13310575 C  0135:0045:2
                       THEN NORMALREF ELSE DECLREF);                                       %116-      13310580 C  0135:0045:2
            END                                                                            %116-      13310600 P  0135:0050:0
           ELSE % DEFINE PARAMETERS - DONT CROSS REF.                                      %116-      13310700 C  0135:0050:0
             XREFINFO[NEXTINFO] := 0                                                       %116-      13310750 C  0135:0050:0
          ELSE                                                                             %116-      13310800 C  0135:0053:3
            IF DEFINING.[1:1]  THEN % WE ARE DOING XREFING                                 %116-      13310900 C  0135:0054:0
              XREFINFO[NEXTINFO] := 0;                                                     %116-      13310950 C  0135:0055:3
           LASTINFO←NEXTINFO;                                                                        13311000 T  0135:0059:3
          IF NEXTINFO + NEXTINFO+WORDCOUNT ≥ 8192 THEN                                               13312000 T  0135:0060:0
          BEGIN FLAG(199); GO TO ENDOFITALL END;                                                     13312500 T  0135:0062:0
END;                                                                                                 13313000 T  0135:0065:3
                                                                                                    135 IS    69 LONG.  NEXT SEG.    3


                    PROCEDURE ENTRY(TYPE);                                                           13314000 T  0003:0881:3
                              VALUE TYPE;                                                            13315000 T  0003:0881:3
                              REAL TYPE;                                                             13316000 T  0003:0881:3
                    COMMENT                                                                          13317000 T  0003:0881:3
                            ENTRY ASSUMES THAT I IS POINTING AT AN IDENTIFIER WHICH                  13318000 T  0003:0881:3
                            IS BEING DECLARED AND MAKES UP THE ELBAT ENTRY FOR IT                    13319000 T  0003:0881:3
                            ACCORD TO TYPE        .IF THE ENTRY IS AN ARRAY AND NOT                  13320000 T  0003:0881:3
                            A SPECIFICATION THEN A DESCRIPTOR IS PLACED IN THE STACK                 13321000 T  0003:0881:3
                            FOR THE UPCOMING COMMUNICATE TO GET STORAGE FOR THE ARRAY(S)     ;       13322000 T  0003:0881:3
                     BEGIN                                                                           13323000 T  0003:0881:3
                             BOOLEAN SVTOG;%                                                         13323010 T  0003:0881:3
                                                                                                  START OF SEGMENT **********  136
STACK(F+2) = SVTOG                                                                                   13324000 T  0136:0000:0
                          J←0;I←I-1;                                                                 13325000 T  0136:0002:0
                          DO                                                                         13326000 T  0136:0002:0
                            BEGIN                                                                    13327000 T  0136:0002:0
                                 STOPDEFINE ←TRUE; STEPIT; SCATTERELBAT;                             13328000 T  0136:0003:3
                                 IF FORMALF←SPECTOG                                                  13329000 T  0136:0003:3
                                    THEN                                                             13330000 T  0136:0004:1
                                     BEGIN                                                           13330550 T  0136:0005:2
                                      IF TYPE≤INTARRAYID AND TYPE≥BOOARRAYID THEN%                   13330600 T  0136:0006:1
                                      IF VONF THEN BEGIN SVTOG ← ERRORTOG;  FLAG(15);%               13330650 T  0136:0009:3
                                          SPECTOG ← ERRORTOG ← SVTOG;  END;%                         13331000 T  0136:0010:1
                                      IF ELCLASS≠SECRET
```

```
                THEN FLAG(002);                                                  13332000 T  0136:0011:2
                     BUP←BUP+1                                                   13333000 T  0136:0012:1
              END                                                                13334000 T  0136:0013:2
             ELSE                                                                13335000 T  0136:0014:0
              BEGIN                                                              13336000 T  0136:0014:0
               IF ELCLASS>IDMAX AND ELCLASS≤FACTOP                               13337000 T  0136:0014:1
                 THEN FLAG(003);                                                 13338000 T  0136:0015:3
               IF ELCLASS = DEFINEDID THEN % CHECK IF NEW DECLARATION            13339000 P  0136:0017:3
                 IF NOT (PTOG OR STREAMTOG) AND LINKF ≥ GLOBALNINF00             13339100 C  0136:0018:0
                   THEN FLAG(1)                                          %118-   13339200 C  0136:0020:0
                  ELSE                                                   %118-   13339300 C  0136:0021:3
               ELSE                                                      %118-   13339400 C  0136:0022:0
                 IF LEVELF = LEVEL THEN % DUPLICATE DECLARATION          %118-   13339500 C  0136:0022:1
                   FLAG(1);                                              %118-   13340000 P  0136:0023:3
               VONF←P2;                                                          13341000 T  0136:0025:2
IF ((FORMALF←PTOG)OR (STREAMTOG AND NOT STOPGSP)) AND NOT P2                     13342000 T  0136:0025:3
                 THEN ADDRSF ← PJ ←PJ+1                                          13343000 T  0136:0027:3
          ELSE IF STOPGSP THEN ADDRSF ← 0                                        13344000 T  0136:0029:2
                          ELSE ADDRSF:=GETSPACE(P2+1); % ID IN ACCUM[1],         13345000 T  0136:0032:0
               IF TYPE≤INTARRAYID AND TYPE≥BOOARRAYID                            13346000 T  0136:0034:1
                 THEN IF P2 THEN BEGIN COMMENT OWN ARRAY;                        13347000 T  0136:0035:3
                                    EMITL(ADDRSF); EMITN(10);                    13347500 T  0136:0037:3
                                 END                                             13347510 T  0136:0039:2
                               ELSE CHECKDISJOINT(ADDRSF);                       13347520 T  0136:0039:2
              END;                                                               13348000 T  0136:0040:1
            IF XREF AND NOT SPECTOG THEN % ERASE PREVIOUS XREF ENTRY,            13348100 P  0136:0040:1
               XREFPT←XREFPT-REAL(ELBAT[I]≠0); % GET RID OF LAST CREF            13348200 T  0136:0042:0
           KLASSF←TYPE; MAKEUPACCUM;E; J←J+1;                                    13349000 T  0136:0044:1
               END                                                               13350000 T  0136:0047:3
              UNTIL STEPI≠COMMA OR STOPENTRY; GTA1[0]←J                          13351000 T  0136:0047:3
            END;                                                                 13352000 T  0136:0050:0
                                                                          136 IS   53 LONG, NEXT SEG    3


              PROCEDURE UNHOOK;                                                  13353000 T  0003:0881:3
            COMMENT                                                              13354000 T  0003:0881:3
               UNHOOK ASSUMES THAT THE WORD IN ELBAT[I] POINTS TO A PSUEDO ENTRY  13355000 T  0003:0881:3
               FOR APARAMETER,ITS JOB IS TO UNHOOK THAT FALSE ENTRY SO THAT      13356000 T  0003:0881:3
               E WILL WORK AS NORMAL,                                    ;       13357000 T  0003:0881:3
               BEGIN                                                             13358000 T  0003:0881:3
                REAL LINKT,A,LINKP;                                              13359000 T  0003:0881:3
                                                                        START OF SEGMENT **********  137
STACK(F+2) = LINKT
STACK(F+3) = A
STACK(F+4) = LINKP
                LABEL L;                                                         13360000 T  0137:0000:0
                      LINKT←STACKHEAD[SCRAM]     ; LINKP←ELBAT[I],LINK;          13361000 T  0137:0000:0
                      IF LINKT=LINKP THEN STACKHEAD[SCRAM]←TAKE(LINKT),LINK      13362000 T  0137:0002:1
                                     ELSE                                        13363000 T  0137:0005:3
                   L: IF A←TAKE(LINKT),LINK=LINKP                                13364000 T  0137:0006:0
                      THEN PUT((TAKE(LINKT))&(TAKE(A))[35:35:13],LINKT)          13365000 T  0137:0008:1
                      ELSE BEGIN LINKT←A; GO TO L END;                           13366000 T  0137:0012:0
               END;                                                              13367000 T  0137:0014:1
                                                                          137 IS   17 LONG, NEXT SEG    3
```

```
                PROCEDURE MAKEUPACCUM;                                                          13368000 T  0003:088!3
                  BEGIN                                                                         13369000 T  0003:088!3
                     IF PTOG                                                                    13370000 T  0003:088!3
                       THEN GT1←LEVELF ELSE GT1←LEVEL;                                          13371000 T  0003:088?0
                     ACCUM[0]← ABS(ELBAT[I] & KLASSF[2:41:7] & REAL(FORMALF)[9:47:1]            13372000 T  0003:088?1
                                & REAL(VONF)[10:47:1] & GT1[11:43:5] &ADDRSF[16:37:11]          13373000 T  0003:088?3
                                )                                                               13374000 T  0003:089?0
                  END;                                                                          13375000 T  0003:089?1


                PROCEDURE ARRAE;                                                                13376000 T  0003:089!3
PRT(1075) = ARRAE
                COMMENT                                                                         13377000 T  0003:089!3
                     ARRAE ENTERS INFO ABOUT ARRAYS AND THEIR LOWER BOUNDS.                     13378000 T  0003:089!3
                     IT ALSO EMITS CODE TO COMMUNICATE WITH THE MCP TO OBTAIN                   13379000 T  0003:089!3
                     STORAGE FOR THE ARRAY AT OBJECT TIME.SPECIAL ANALYSIS IS                   13380000 T  0003:089!3
                     MADE TO GENERATE EFFICIENT CODE WHEN DETERMING THE SIZE OF                 13381000 T  0003:089!3
                     EACH DIMENSION.FOLLOWING ARE A FEW EXAMPLES OF CODE EMITTED:               13382000 T  0003:089!3
                       ARRAY A[0:10],                                                           13383000 T  0003:089!3
                                    MKS       (THIS MARKS STACK TO CUT BACK AFTER COM)         13384000 T  0003:089!3
                                    DESC A    (THIS FORMS A DESCRITOR POINTING TO              13385000 T  0003:089!3
                                               THE ADDRESS OF A)                                13386000 T  0003:089!3
                                    LITC 11   (SIZE OF ARRAY)                                  13387000 T  0003:089!3
                                    LITC 1    (NUMBER OF DIMENSIONS)                           13388000 T  0003:089!3
                                    LITC 1    (NUMBER OF ARRAYS)                               13389000 T  0003:089!3
                                    LITC ARCOM (COMMUNICATE LITERAL FOR NON SAVE,              13390000 T  0003:089!3
                                               NON OWN ARRAYS)                                 13391000 T  0003:089!3
                                    COM       (COMMUNICATE TO MCP TO GET STORAGE)              13392000 T  0003:089!3
                                    DESC XITR (XITR JUST EXITS,THUS CUTTING BACK               13393000 T  0003:089!3
                                               STACK)                                          13394000 T  0003:089!3
                       OWN ARRAY B,C[0:X,-1:10],                                               13395000 T  0003:089!3
                                    MKS                                                        13396000 T  0003:089!3
                                    DESC B                                                     13397000 T  0003:089!3
                                    DESC C                                                     13398000 T  0003:089!3
                                    LITC 0    (LOWER BOUND MUST BE PASSED FOR OWN)             13399000 T  0003:089!3
                                    OPDC X                                                     13400000 T  0003:089!3
                                    LITC JUNK   (JUNK CELL)                                    13401000 T  0003:089!3
                                    ISN         (INTEGERIZE UPPER BOUND)                       13402000 T  0003:089!3
                                    LITC 1    (COMPUTE SIZE                                    13403000 T  0003:089!3
                                    ADD        OF DIMENSION                                    13404000 T  0003:089!3
                                    LITC 1    (LOWER BOUND,SECOND DIMENSION)                   13405000 T  0003:089!3
                                    CHS                                                        13406000 T  0003:089!3
                                    LITC 12   (SIZE SECOND DIMENSION)                          13407000 T  0003:089!3
                                    LITC 2    (NUMBER DIMENSIONS)                              13408000 T  0003:089!3
                                    LITC 2    (NUMBER ARRAYS)                                  13409000 T  0003:089!3
                                    LITC OWNCOM (OWN ARRAY COMMUNICATE)                        13410000 T  0003:089!3
                                    COM                                                        13411000 T  0003:089!3
                                    DESC XITR                                                  13412000 T  0003:089!3
SAVE OWN ARRAY D,E,F[X:Y,M+N:TxV],                                                             13413000 T  0003:089!3
                                    MKS                                                        13414000 T  0003:089!3
                                    DESC D                                                     13415000 T  0003:089!3
                                    DESC E                                                     13416000 T  0003:089!3
                                    DESC F                                                     13417000 T  0003:089!3
```

```
OPDC X                                                                  13418000 T  0003!089!!3
LITC XT     (CELL OBTAINED TO KEEP LOWER BOUND)                         13419000 T  0003!089!!3
ISN         (PUT INTEGERIZED LOWER BOUND AWAY)                          13420000 T  0003!089!!3
DUP         (MUST PASS LOWER BOUND FOR OWN)                             13421000 T  0003!089!!3
OPDC Y      (INTEGERIZE                                                 13422000 T  0003!089!!3
LITC JUNK     UPPER                                                     13423000 T  0003!089!!3
ISN             BOUND)                                                  13424000 T  0003!089!!3
XCH         (COMPUTE SIZE OF FIRST DIMENSION                            13425000 T  0003!089!!3
SUB          UPPER                                                      13426000 T  0003!089!!3
LITC 1       -LOWER                                                     13427000 T  0003!089!!3
ADD          +1)                                                        13428000 T  0003!089!!3
OPDC M       (COMPUTE LOWER BOUND                                       13429000 T  0003!089!!3
OPDC N        SECOND DIM)                                               13430000 T  0003!089!!3
ADD                                                                     13431000 T  0003!089!!3
LITC MNT    (GET CELL FOR SECOND LOWER BOUND)                           13432000 T  0003!089!!3
ISN         (INTEGERIZE)                                                13433000 T  0003!089!!3
DUP         (PASS LOWER BOUND FOR OWN)                                  13434000 T  0003!089!!3
OPDC T                                                                  13435000 T  0003!089!!3
MUL V                                                                   13436000 T  0003!089!!3
LITC JUNK   (INTEGERIZE                                                 13437000 T  0003!089!!3
ISN         UPPER)                                                      13438000 T  0003!089!!3
XCH         (COMPUTE                                                    13439000 T  0003!089!!3
SUB          SIZE                                                       13440000 T  0003!089!!3
LITC 1                                                                  13441000 T  0003!089!!3
ADD               )                                                     13442000 T  0003!089!!3
LITC 2      (NUMBER DIMENSIONS)                                         13443000 T  0003!089!!3
LITC 3      (NUMBER ARRAYS)                                             13444000 T  0003!089!!3
LITC SAVON  (SAVE OWN LITERAL FOR COM)                                  13445000 T  0003!089!!3
COM                                                                     13446000 T  0003!089!!3
DESC XITR ;                                                             13447000 T  0003!089!!3
                    BEGIN                                               13448000 T  0003!089!!3
                     REAL T1,T2,T3,K,LBJ,ARPROGS,SAVEDIM,T,T4,SAVEINFO,SAVEINFO2;   13449000 T  0003!089!!3
                                                                  START OF SEGMENT ********** 138
      STACK(F+2) = T1
      STACK(F+3) = T2
      STACK(F+4) = T3
      STACK(F+5) = K
      STACK(F+6) = LBJ
      STACK(F+7) = ARPROGS
      STACK(F+10) = SAVEDIM
      STACK(F+11) = T
      STACK(F+12) = T4
      STACK(F+13) = SAVEINFO
      STACK(F+14) = SAVEINFO2
                    BOOLEAN LLITOG,ULITOG;                              13450000 T  0138!000010
      STACK(F+15) = LLITOG
      STACK(F+16) = ULITOG
                  REAL ADDCON;                                          13451000 T  0138!000010
      STACK(F+17) = ADDCON
                    LABEL CSZ,BETA1,TWO,START,SLB,BETA2;                13452000 T  0138!000010
                    ARRAYFLAG ← TRUE;                                   13452100 T  0138!000010
                     TYPEV←REALARRAYID;                                 13453000 T  0138!000011
                     IF T1←GTA1[J←J+1]=0 THEN J←J+1                     13454000 T  0138!000113
                ELSE                                                    13455000 T  0138!000411
                    IF T1=OWNV THEN                                     13456000 T  0138!000513
                        BEGIN  P2!=TRUE;IF SPECTOG THEN FLAG(13)  END   13457000 T  0138!000712
                ELSE                                                    13458000 T  0138!000913
                    IF T1= SAVEV THEN                                   13459000 T  0138!000913
```

```
          BEGIN                                                                  13460000 T  0138:0011:2
          P3:=TRUE;                                                              13461000 T  0138:0011:3
          IF SPECTOG THEN FLAG(13);                                              13462000 T  0138:0012:0
%         IF REMOTOG THEN FLAG(508); % NOT ALLOWED IN XALGOL ON TSS.             13463000 T  0138:0013:3
          END                                                                    13464000 T  0138:0013:3
ELSE                                                                             13465000 T  0138:0013:3
    IF T1= AUXMEMV THEN                                                          13466000 T  0138:0013:3
         BEGIN  P4:=TRUE; IF SPECTOG THEN FLAG(13)  END                          13467000 T  0138:0015:2
ELSE                                                                             13468000 T  0138:0017:3
    TYPEV :=REALID+T1;                                                           13469000 T  0138:0017:3
   IF NOT SPECTOG THEN EMITO(MKS); SAVEINFO+NEXTINFO;                            13470000 T  0138:0019:3
    ENTER(TYPEV);      SAVEINFO2+NEXTINFO+NEXTINFO+1;                            13471000 T  0138:0022:0
BETA1:                                                                           13472000 T  0138:0024:1
   IF EL.CLASS#LFTBRKET THEN FLAG(016); LBJ+0;SAVEDIM+1;                         13473000 T  0138:0025:2
TWO:IF STEPI=ADOP THEN                                                           13474000 T  0138:0028:1
                    BEGIN                                                        13475000 T  0138:0030:0
                      T1+ELBAT[I].ADDRESS; I+I+1                                 13476000 T  0138:0030:1
                    END                                                          13477000 T  0138:0032:0
                  ELSE T1+0;IF SPECTOG THEN GO TO BETA2;                         13478000 T  0138:0033:2
   ARPROGS+L;                                                                    13479000 T  0138:0035:3
   IF TABLE(I+1)=COLON AND TABLE(I)=LITNO THEN                                   13480000 T  0138:0036:0
                                          BEGIN                                  13481000 T  0138:0039:3
                                            LLITOG+TRUE;                         13482000 T  0138:0040:0
                                            IF T3+ELBAT[I].ADDRESS#0             13483000 T  0138:0040:1
                                               THEN                              13484000 T  0138:0042:0
                                                 BEGIN                           13485000 T  0138:0042:1
                                                   EMITL(T3);                    13486000 T  0138:0043:2
                                                   IF T1=SUBOP THEN              13487000 T  0138:0044:0
                                                    BEGIN                        13488000 T  0138:0044:1
                                                     EMITO(CHS);                 13489000 T  0138:0045:2
                                                     ADDCON+ADDC                 13490000 T  0138:0046:0
                                                    END         ELSE             13491000 T  0138:0046:0
                                                     ADDCON+SUBC                 13492000 T  0138:0046:1
                                                 END;                            13493000 T  0138:0049:2
                                                T2+T3×4+ADDCON                   13494000 T  0138:0049:3
                                            END                                  13495000 T  0138:0050:1
                                          ELSE                                   13496000 T  0138:0051:3
                                           BEGIN                                 13497000 T  0138:0051:3
                                            LLITOG+FALSE;                        13498000 T  0138:0053:2
                                            IF T1#0 THEN I+I-1;                  13499000 T  0138:0053:3
                                            T2:=GETSPACE(P2,-1);%TEMP.           13500000 T  0138:0056:0
                                            AEXP;EMITSTORE(T2,ISN);              13501000 T  0138:0058:0
                                       T2+T2×4+SUBC+2;                           13502000 T  0138:0059:3
                                            IF ELCLASS#COLON THEN                13503000 T  0138:0061:3
                                             FLAG(017);I+I-1                     13504000 T  0138:0062:1
                                           END;                                  13505000 T  0138:0064:0
                                  IF P2 THEN                                     13506000 T  0138:0065:2
                                  BEGIN                                          13507000 T  0138:0065:2
                                       IF LLITOG AND T3=0 THEN EMITL(0);         13508000 T  0138:0065:3
                                       ARPROGS+L;EMITO(DUP);                     13509000 T  0138:0068:0
                                  END;                                           13510000 T  0138:0069:3
IF ELCLASS+TABLE(I+I+2)=LITNO THEN                                               13511000 T  0138:0069:3
                                   BEGIN                                         13512000 T  0138:0072:1
                                     IF T+TABLE(I+I+1)=COMMA OR                  13513000 T  0138:0073:2
                                        T=RTBRKET                                13514000 T  0138:0075:3
                                      THEN                                       13515000 T  0138:0076:0
                                        BEGIN                                    13516000 T  0138:0076:1
```

```
                                        EMITL(T4+ELBAT[I-1],ADDRESS);                13517000 T  0138:007:2
                                        ULITOG+TRUE;GO TO CSZ                        13518000 T  0138:007:3
                                      END                                            13519000 T  0138:008:2
                                    ELSE                                             13520000 T  0138:008:0
                                      I+I-1                                          13521000 T  0138:008:0
                                  END;                                               13522000 T  0138:008:1
      ULITOG+FALSE;                                                                  13523000 T  0138:008:3
      AEXP;                                                                          13524000 T  0138:008:1
      EMITL(JUNK);                                                                   13525000 T  0138:008:2
      EMITO(ISN);                                                                    13526000 T  0138:008:3
CSZ: IF LLITOG AND ULITOG THEN                                                       13527000 T  0138:008:1
                          BEGIN                                                      13528000 T  0138:008:3
                             L+ARPROGS;                                              13529000 T  0138:008:0
                             IF(T+IF ADDCON=ADDC THEN T4+T3+1 ELSE                   13530000 T  0138:008:2
                                  T4-T3+1)≤0 OR T>1023 THEN FLAG(59);                13531000 T  0138:009:0
                             EMITL(T);                                               13531100 T  0138:009:1
                            IF P3 THEN BEGIN SAVEDIM+SAVEDIM×T;                      13532000 T  0138:009:2
                                             IF SAVEDIM>MAXSAVE                      13533000 T  0138:009:2
                                               THEN MAXSAVE+SAVEDIM                  13534000 T  0138:009:3
                                        END                                          13535000 T  0138:010:1
                                      ELSE                                           13536000 T  0138:010:2
                                        IF T>MAXROW THEN MAXROW+T;                   13537000 T  0138:010:2
                          END                                                        13538000 T  0138:010:2
                        ELSE                                                         13539000 T  0138:010:2
                          BEGIN IF NOT(LLITOG AND T3=0)                              13540000 T  0138:010:2
                    OR P2                                                            13541000 T  0138:010:6
                                      THEN                                           13542000 T  0138:010:7
                                        BEGIN                                        13543000 T  0138:010:7
                                          EMITO(XCH);EMITO(SUB)                      13544000 T  0138:010:8
                                        END;EMITL(1);EMITO(ADD)                      13545000 T  0138:010:9
                          END;                                                       13546000 T  0138:011:1
SLB:PUTNBUMP(T2);LBJ+LBJ+1;IF T+TABLE(I)=COMMA THEN GO TO TWO                        13547000 T  0138:011:2
                                                ELSE                                 13548000 T  0138:011:1
                                  IF T≠RTBRKET THEN FLAG(018);                       13549000 T  0138:011:1
    IF NOT SPECTOG THEN                                                              13550000 T  0138:011:2
          BEGIN                                                                      13551000 T  0138:011:3
COMMENT   KEEP COUNT OF NO. OF ARRAYS DECLARED;                                      13551400 T  0138:011:8
          NOOFARRAYS:=NOOFARRAYS + GTA1[0];                                          13551500 T  0138:011:8
          EMITL(LBJ);EMITL(GTA1[0]);                                                 13552000 T  0138:011:9
          IF P3 AND P4 THEN FLAG(14); % SAVE AND AUXMEM MUTUALLY EXCL.               13552500 T  0138:012:1
          EMITL(REAL(P3)+2×REAL(P2)+REAL(P4)×64);                                    13553000 T  0138:012:3
          EMITV(5)                                                                   13554000 T  0138:012:6
          END;                                                                       13555000 T  0138:012:6
            PUT(LBJ,SAVEINFO2-1);                                                    13556000 T  0138:012:7
 DO BEGIN                                                                            13557000 T  0138:012:8
            T+TAKE(SAVEINFO);                                                        13558000 T  0138:012:9
            K+T.INCR;                                                                13559000 T  0138:013:0
            T.INCR+SAVEINFO2-SAVEINFO-1;                                             13560000 T  0138:013:1
            PUT(T,SAVEINFO);                                                         13561000 T  0138:013:4
     END                                                                             13562000 T  0138:013:5
          UNTIL SAVEINFO+SAVEINFO+K=SAVEINFO2-1;                                     13563000 T  0138:013:5
    IF STEPI≠COMMA THEN GO TO START;                                                 13564000 T  0138:013:8
    IF NOT SPECTOG THEN EMITO(MKS);                                                  13565000 T  0138:013:9
       SAVEINFO+NEXTINFO;                                                            13566000 T  0138:014:1
       I+I+1;ENTRY(TYPEV);SAVEINFO2+NEXTINFO+NEXTINFO+1;GO TO BETA1;                 13567000 T  0138:014:2
BETA2:                                                                               13568000 T  0138:014:6
    IF T+ TABLE(I+I+1)=COMMA OR T=RTBRKET                                            13569000 T  0138:014:7
       THEN                                                                          13570000 T  0138:015:0
```

```
          BEGIN                                                                 13571000 T  0138:0150:1
            IF ELCLASS*TABLE[I-1)=LITNO                                         13572000 T  0138:0151:2
              THEN                                                              13573000 T  0138:0153:2
                BEGIN                                                           13574000 T  0138:0153:3
                  T3*ELBAT[I-1],ADDRESS;                                        13575000 T  0138:0154:0
                  IF T1= SUBOP THEN                                             13576000 T  0138:0156:0
                     ADDCON *ADDC                                               13577000 T  0138:0156:1
                               ELSE                                             13578000 T  0138:0157:3
                     ADDCON *SUBC;                                              13579000 T  0138:0158:0
                  T2*T3×4+ADDCON; GO TO SLB;                                    13580000 T  0138:0160:1
                END;                                                            13581000 T  0138:0164:0
            IF ELCLASS=FACTOP THEN                                              13582000 T  0138:0164:0
                BEGIN                                                           13583000 T  0138:0164:1
                 T2*-SUBC; GO TO SLB                                            13584000 T  0138:0165:2
                END                                                             13585000 T  0138:0166:1
          END;                                                                  13586000 T  0138:0168:0
          FLAG(019);                                                            13587000 T  0138:0168:0
START; ARRAYFLAG * FALSE END;                                                   13588000 T  0138:0168:1
                                                                      138 IS  175 LONG, NEXT SEG    3


 PROCEDURE PUTNBUMP(X);                                                         13589000 T  0003:0891:3
             VALUE X;                                                           13590000 T  0003:0891:3
             REAL X;                                                            13591000 T  0003:0891:3
  BEGIN                                                                         13592000 T  0003:0891:3
        INFO[NEXTINFO,LINKR,NEXTINFO,LINKC]*X;                                  13593000 T  0003:0891:3
        NEXTINFO*NEXTINFO+1                                                     13594000 T  0003:0895:2
   END ;                                                                        13595000 T  0003:0895:2


 PROCEDURE JUMPCHKX;                                                            13596000 T  0003:0896:1
COMMENT THIS PROCEDURE IS CALLED AT THE START OF ANY EXECUTABLE CODE            13597000 T  0003:0896:1
        WHICH THE BLOCK MIGHT EMIT.IT DETERMINES WHETHER ANY JUMPS              13598000 T  0003:0896:1
        ARROUND NONEXECUTABLE CODE MAY BE WAITING AND WHETHER IT                13599000 T  0003:0896:1
        IS THE FIRST EXECUTABLE CODE;                                           13600000 T  0003:0896:1
IF NOT SPECTOG THEN                                                             13601000 T  0003:0896:1
BEGIN                                                                           13602000 T  0003:0897:3
   IF AJUMP                                                                     13603000 T  0003:0898:0
       THEN                                                                     13604000 T  0003:0898:0
   BEGIN ADJUST;                                                                13605000 T  0003:0898:0
         EMITB(BFW,SAVEL,L)                                                     13606000 T  0003:0899:2
   END FLSE                                                                     13607000 T  0003:0900:0
         IF FIRSTX=4095                                                         13608000 T  0003:0900:1
           THEN                                                                 13609000 T  0003:0901:2
             BEGIN                                                              13610000 T  0003:0901:3
               ADJUST;                                                          13611000 T  0003:0902:0
               FIRSTX*L;                                                        13612000 T  0003:0902:1
             END;                                                               13613000 T  0003:0903:3
      AJUMP*FALSE                                                               13614000 T  0003:0903:3
END;                                                                            13615000 T  0003:0903:3
```

```
                 PROCEDURE JUMPCHKNX;                                                    13616000 T  0003:0906:0
                COMMENT JUMPCHKNX DETERMINES WHETHER ANY EXECUTABLE CODE HAS BEEN        13617000 T  0003:0906:0
                        EMITTED AND IF SO WHETHER IT WAS JUST PREVIOUS TO THE            13618000 T  0003:0906:0
                        NON EXECUTABLE ABOUT TO BE EMITTED.IF BOTH THEN L IS BUMPED      13619000 T  0003:0906:0
                        AND SAVED FOR A LATER BRANCH;                                    13620000 T  0003:0906:0
                IF NOT SPECTOG THEN                                                      13621000 T  0003:0906:0
                BEGIN                                                                    13622000 T  0003:0906:1
                   IF FIRSTX≠4095                                                        13623000 T  0003:0907:2
                     THEN                                                                13624000 T  0003:0907:2
                      BEGIN                                                              13625000 T  0003:0907:3
                       IF NOT AJUMP                                                      13626000 T  0003:0908:0
                         THEN                                                            13627000 T  0003:0908:0
                            SAVEL←BUMPL;                                                 13628000 T  0003:0908:1
                      AJUMP←TRUE                                                         13629000 T  0003:0911:2
                      END;ADJUST                                                         13630000 T  0003:0911:2
                END;                                                                     13631000 T  0003:0911:3



                PROCEDURE SEGMENTSTART;                                                  13632000 T  0003:0914:0
                    BEGIN                                                                13632100 T  0003:0914:0
                    IF NOHEADING THEN DATIME;                                            13633000 T  0003:0914:0
                    IF SINGLTOG THEN WRITE(LINE,PRINTSEGNO,SGAVL)                        13633100 T  0003:0915:2
PRT(1076) = *LIST, LABEL, OR SEGMENT DESCRIPTOR*
                                ELSE WRITE(LINE[DBL],PRINTSEGNO,SGAVL);                  13633200 T  0003:0921:2
PRT(1077) = *LIST, LABEL, OR SEGMENT DESCRIPTOR*
                 END SEGMENTSTART;                                                       13633300 T  0003:0932:1
```

```
                PROCEDURE SEGMENT(SIZE,NO,NOO);                                  %106-        13634000 C  0003:0932:1
                    VALUE SIZE,NO,NOO;                                           %106-        13635000 C  0003:0932:1
                    REAL  SIZE,NO,NOO;                                           %106-        13636000 C  0003:0932:1
                    BEGIN                                                        %106-        13637000 C  0003:0932:1
                    INTEGER DUMMY; % THIS IS HERE SO THAT OUR CODE SEGMENT       %106-        13637100 C  0003:0932:1
                                                                                        START OF SEGMENT **********  139
STACK(F+2) = DUMMY
                                % IS NOT TOO BIG                                  %106-        13637200 C  0139:0000:0
                    PDPRT[PDINX,[37:5],PDINX,[42:6]] :=                          %106-        13638000 C  0139:0000:0
                         SIZE & NO[28:38:10] &                                   %106-        13639000 C  0139:0002:0
                         MOVEANDBLOCK(EDOC,ABS(SIZE),-ABS(NO))[13:33:15] &       %106-        13640000 C  0139:0003:3
                         REAL(SAVEPRTOG)[3:47:1];                                %106-        13641000 C  0139:0007:2
                     PDINX:=PDINX+1;  SIZE:=ABS(SIZE);                           %106-        13642000 C  0139:0008:1
                     IF SIZE>SEGSIZEMAX THEN SEGSIZEMAX:=SIZE;                   %106-        13643000 C  0139:0010:1
                     AKKUM:=AKKUM+SIZE;                                          %106-        13644000 C  0139:0012:1
                     IF SAVEPRTOG THEN AUXMEMREQ:=AUXMEMREQ+16×(SIZE.[38:6]+1);  %106-        13645000 C  0139:0014:0
                     IF LISTER OR SEGSTOG THEN                                   %106-        13646000 C  0139:0017:3
                          BEGIN                                                  %106-        13647000 C  0139:0018:1
                          IF NOHEADING THEN DATIME;                              %106-        13648000 C  0139:0019:2
                          IF SINGLTOG THEN WRITE(LINE,PRINTSIZE,NO,SIZE,NOO)     %106-        13649000 C  0139:0020:1
PRT(1100) = *LIST, LABEL, OR SEGMENT DESCRIPTOR*
                               ELSE WRITE(LINE,PRINTSIZE,NO,SIZE,NOO);           %106-        13650000 C  0139:0032:0
PRT(1101) = *LIST, LABEL, OR SEGMENT DESCRIPTOR*
                          END;                                                   %106-        13651000 C  0139:0043:3
                     LDICT[NO,[38:3],NO,[41:7]] :=                               %106-        13652000 C  0139:0043:3
                          IF BUILDLINE THEN                                      %106-        13653000 C  0139:0045:3
```

```
            MOVEANDBLOCK(ENIL,ENILPTR+1,4) & SIZE[18:33:15]      %106-          13654000 C  0139:0046:0
        FLSE -1;                                                 %106-          13655000 C  0139:0049:3
    ENd OF SEGMENT;                                              %106-          13656000 C  0139:0051:2
                                                                        139 IS   54 LONG, NEXT SEG    3

                                                                                13697000 T  0003:0932:1
                                                                                13698000 T  0003:0932:1
                                                                                13699000 T  0003:0932:1
                                                                                13700000 T  0003:0932:1
                                                                                13701000 T  0003:0932:1
                                                                                13702000 T  0003:0932:1
                                                                                13703000 T  0003:0932:1
                                                                                13704000 T  0003:0932:1
                                                                                13705000 T  0003:0932:1
                                                                                13706000 T  0003:0932:1
                                                                                13707000 T  0003:0932:1
                                                                                13708000 T  0003:0932:1
                                                                                13709000 T  0003:0932:1
                                                                                13710000 T  0003:0932:1
                                                                                13711000 T  0003:0932:1
                                                                                13712000 T  0003:0932:1
                                                                                13713000 T  0003:0932:1
PROCEDURE ENTER(TYPE); VALUE TYPE; INTEGER TYPE;                                13714000 T  0003:0932:1
    BEGIN                                                                       13715000 T  0003:0932:1
    G:=GTA1[J:=J-1];                                                            13716000 T  0003:0932:1
    IF NOT SPECTOG THEN                                                         13717000 T  0003:0935:2
        BEGIN                                                                   13718000 T  0003:0935:3
        IF NOT P2 THEN                                                          13719000 T  0003:0936:0
         IF P2:=(G=OWNV) THEN G:=GTA1[J:=J-1];                                  13720000 T  0003:0936:1
        IF NOT P3 THEN                                                          13721000 T  0003:0940:1
          IF P3:=(G=SAVEV) THEN G:=GTA1[J:=J-1];                                13722000 T  0003:0941:2
        IF NOT P4 THEN                                                          13723000 T  0003:0945:3
          IF P4:=(G=AUXMEMV) THEN G:=GTA1[J:=J-1];                              13724000 T  0003:0946:0
        END;                                                                    13725000 T  0003:0950:0
        IF G≠0 THEN FLAG(25) ELSE ENTRY(TYPE)                                   13726000 T  0003:0950:0
    END ENTER;                                                                  13727000 T  0003:0953:2

                                                                                13728000 T  0003:0953:3
                                                                                13729000 T  0003:0953:3
                                                                                13730000 T  0003:0953:3
                PROCEDURE HTTEOAP(GOTSTORAGE,RELAD,STOPPER,PRTAD);              13731000 T  0003:0953:3
PRT(1102) = HTTEOAP
                    VALUE GOTSTORAGE,RELAD,STOPPER,PRTAD;                       13732000 T  0003:0953:3
                    BOOLEAN GOTSTORAGE;                                         13733000 T  0003:0953:3
                    REAL RELAD,STOPPER,PRTAD;                                   13734000 T  0003:0953:3
                BEGIN                                                           13735000 T  0003:0953:3
                    BOOLEAN BT;                                                 13736000 T  0003:0953:3
                                                                        START OF SEGMENT **********  140
STACK(F+2) = BT
                    REAL K,LS;                                                  13737000 T  0140:0000:0
STACK(F+3) = K
```

```
STACK(F+4) = LS
                    LS←RELAD;
                              BT←JUMPCTR=LEVEL;                                                                    13738000 T  0140:000:0
                    IF FUNCTOG                                                                                     13739000 T  0140:000:1
                      THEN                                                                                         13740000 T  0140:000:2:0
                       BEGIN                                                                                       13741000 T  0140:000:2:0
                         EMITV(514);                                                                               13742000 T  0140:000:2:0
                         EMITO(RTN)                                                                                13743000 T  0140:000:2:1
                       END                                                                                         13744000 T  0140:000:3:3
                     ELSE                                                                                          13745000 T  0140:000:3:3
                       EMITO(XIT);                                                                                 13746000 T  0140:000:4:0
                    IF STACKCTR>MAXSTACK THEN MAXSTACK←STACKCTR;                                                   13747000 T  0140:000:4:0
                    CONSTANTCLEAN;                                                                                 13748000 T  0140:000:5:3
                    IF K←MAXSTACK-514>0 OR GOTSTORAGE OR BT OR NCII>0 OR FAULTOG.[46:1]                            13749000 T  0140:000:7:3
                      THEN                                                                                         13750000 T  0140:000:8:0
                       BEGIN ADJUST;LS←L;                                                                          13751000 T  0140:001:1:3
                        IF BT OR GOTSTORAGE OR FAULTOG.[46:1]                                                      13752000 T  0140:001:2:1
                          THEN                                                                                     13753000 T  0140:001:4:1
                            BEGIN                                                                                  13754000 T  0140:001:5:2
                                                                                                                   13755000 T  0140:001:6:0
                %                                                                                                  13755500 T  0140:001:6:1
                              EMITV(BLOCKCTR);                                                                     13756000 T  0140:001:6:1
                              EMITL(1);                                                                            13757000 T  0140:001:7:3
                              EMITO(ADD);                                                                          13758000 T  0140:001:8:0
                              IF GOTSTORAGE OR FAULTOG.[46:1]                                                      13759000 T  0140:001:9:2
                                THEN                                                                               13760000 T  0140:001:9:2
                                  EMITSTORE(BLOCKCTR,SND)                                                          13761000 T  0140:002:0:0
                            END                                                                                    13762000 T  0140:002:1:2
                %                                                                                                  13762500 T  0140:002:1:3
                          ELSE EMITL(0);                                                                           13763000 T  0140:002:1:3
                        K←K+NCII;                                                                                  13764000 T  0140:002:3:2
                        WHILE K←K-1≥0                                                                              13765000 T  0140:002:4:0
                          DO  EMITL(0);                                                                            13766000 T  0140:002:6:0
                        PURGE(STOPPER);                                                                            13767000 T  0140:002:8:1
                         IF FAULTLEVEL≤LEVEL THEN                                                                  13767100 T  0140:002:9:2
                              BEGIN IF FAULTLEVEL=LEVEL THEN FAULTLEVEL←32;                                        13767200 T  0140:003:0:0
                                  EMITPAIR(0,MDS); EMITO(CHS);                                                     13767300 T  0140:003:2:1
                              END OF THIS PART OF ERROR KLUDGE;                                                    13767400 T  0140:003:4:0
                        EMIT(0);                                                  % DC & DISK                     13767500 T  0140:003:4:0
                              BUMPL;                                                                               13768000 T  0140:003:5:2
                                                                                                                   13768100 T  0140:003:6:0
                        EMITB(BBC+L,IF RELAD=4095 THEN 0 ELSE RELAD) ; % DC & DISK                                 13769000 T  0140:003:6:0
                        CONSTANTCLEAN                                                                              13770000 T  0140:003:9:3
                        END ELSE PURGE(STOPPER);                                                                   13771000 T  0140:003:9:3
                    Z←PROGDESCBLDR(PDES,IF LS=4095 THEN 0 ELSE LS,PRTAD);                                          13772000 T  0140:004:2:1
                  END  HTTPOAP;                                                                                    13773000 T  0140:004:6:1
                                                                                                                 140 IS   50 LONG, NEXT SEG    3



                  PROCEDURE FORMATGEN;                                                                             13774000 T  0003:095:3
                   BEGIN                                                                                           13775000 T  0003:095:3
                     INTEGER PRT;LABEL L;                                                                          13776000 T  0003:095:3:3
                                                                                                                 START OF SEGMENT **********  141
STACK(F+2) = PRT
                         BOOLEAN TB2;                                                                              13777000 T  0141:000:1:0
STACK(F+3) = TB2
```

```
                    ARRAY TEDOC[0:7,0:127];
STACK(F+4) = TEDOC
                    MOVFCODE(TEDOC,EDOC);
                    BUILDLINE ← BOOLEAN(2×REAL(BUILDLINE)) ;
                     TB2←GTA1[J-1]=SWITCHV;
                      GT5←SGNO;
                  L: GT1:=(?×SGAVL-1)&2[4:46:2]; STOPENTRY:=TRUE;
                   IF LISTER OR SEGSTOG THEN SEGMENTSTART;
                         SGNO←SGAVL;

                     F:=0; PRT:=GETSPACE(TRUE,1);STOPGSP:=TRUE; % FORMAT,
                          Z←PROGDESCBLDR(LDES,0,PRT);
                       IF TB2 THEN
                       BEGIN
                         ENTRY(SUPERFRMTID);IF ELCLASS≠ASSIGNOP THEN FLAG(36);
                         PUT(TAKE(LASTINFO)&PRT[16:37:11],LASTINFO);
                         RR4←NEXTINFO;PUTNBUMP(0);
                         DO
                           BEGIN PUTNBUMP(F); IF STEP≠LEFTPAREN THEN FLAG(37);
                                 ELCLASS:="<";
                           TB1←FORMATPHRASE;
                           END
                         UNTIL ELCLASS≠",";
                         RR3←NEXTINFO-1;NEXTINFO←RR4;PUTNBUMP(F);
                         DO
                           WHIPOUT(TAKE(RR4←RR4+1))
                         UNTIL RR4=RR3; IF F>1022 THEN FLAG(38);

                       END ELSE
                       BEGIN
                      I←I-1;
                      DO
                        BEGIN
                           STOPDEFINE←TRUE;STEPIT                    ;
                           ENTRY(FRMTID); IF ELCLASS≠LEFTPAREN THEN FLAG(32); ELCLASS:="<";
                          PUT(TAKE(LASTINFO)&PRT[16:37:11]&F[27:40:8],LASTINFO);
                          TB1←FORMATPHRASE;
                        END
                       UNTIL ELCLASS≠"," OR TB1←F≥256      ;


                       END;
                            SEGMENT(-F,SGNO,GT5);SGAVL←SGAVL+1;
                      IF TB1 AND ELCLASS="," THEN BEGIN I←I+1;GO TO L END;
                       IF ELCLASS≠";" THEN ELBAT[I]←0 ELSE ELBAT[I],CLASS←SEMICOLON;
                       STOPGSP←STOPENTRY←FALSE;
                          SGNO←GT5;
                      MOVFCODE(TEDOC,EDOC);
                      BUILDLINE ← BUILDLINE,[46:1] ;
                  END FORMATGEN;
```

```
13777500 T  0141:0000:0
13777600 T  0141:0002:0
13777700 T  0141:0004:0
13778000 T  0141:0005:3
13779000 T  0141:0007:3
13780000 T  0141:0008:0
13780002 T  0141:0012:1
13781000 T  0141:0014:1
13782000 T  0141:0015:3
13783000 T  0141:0015:3
13784000 T  0141:0018:1
13785000 T  0141:0020:0
13786000 T  0141:0020:1
13787000 T  0141:0021:2
13788000 T  0141:0023:3
13789000 T  0141:0026:0
13790000 T  0141:0027:3
13791000 T  0141:0028:0
13791050 T  0141:0031:2
13792000 T  0141:0031:3
13793000 T  0141:0032:1
13794000 T  0141:0032:1
13795000 T  0141:0034:0
13796000 T  0141:0036:1
13797000 T  0141:0037:2
13798000 T  0141:0038:1
13799000 T  0141:0042:1
13800000 T  0141:0042:1
13801000 T  0141:0042:1
13802000 T  0141:0043:2
13803000 T  0141:0044:1
13804000 T  0141:0045:2
13805000 T  0141:0045:2
13806000 T  0141:0046:0
13807000 T  0141:0049:2
13808000 T  0141:0053:2
13809000 T  0141:0054:0
13810000 T  0141:0054:0
13811000 T  0141:0057:2
13812000 T  0141:0057:2
13813000 T  0141:0057:2
13814000 T  0141:0057:2
13815000 T  0141:0059:3
13816000 T  0141:0063:2
13817000 T  0141:0068:1
13818000 T  0141:0070:0
13818500 T  0141:0070:1
13818600 T  0141:0073:2
13819000 T  0141:0074:0
```

141 IS 78 LONG, NEXT SEG 3

```
PROCEDURE CHECKBOUNDLVL   ;                                  13819100 T  0003:0953:3
        COMMENT CHECK    DYNAMIC ARRAY BOUND; MUST NOT BE    13819200 T  0003:0953:3
```

```
                    DECLARED AT SAME LEVEL;                                                    13819300 T  0003:095:3
                IF NOT SPECTOG AND ELBAT[I].LVL=LEVEL                                          13819400 T  0003:095:3
                THEN FLAG(IF REAL(ARRAYFLAG)=3 THEN 509 ELSE 46) ;                             13819410 T  0003:095:3



                    COMMENT 46-ARRAE NON-LITERAL ARRAY BOUND NOT GLOBAL TO                     13819500 T  0003:095:3
                            ARRAY DECLARATION;                                                 13819600 T  0003:095:3
            PROCEDURE FAULTDEC; COMMENT FAULTDEC HANDLES THE MONITOR <FAULT LIST>              13900000 T  0003:095:3
                    THING, FOR THE RUN-TIME ERROR BUSINESS, IT GETS STACK OR                   13901000 T  0003:095:3
                    PRT SPACE AND PASSES SOME STUFF TO THE BLOCK CONTROL                       13902000 T  0003:095:3
                    INTRINSIC, WHO WILL BUILD AIT ENTRIES;                                     13903000 T  0003:095:3
            BEGIN INTEGER TP; REAL A;                                                          13903100 T  0003:095:3
                                                                                  START OF SEGMENT **********  142
STACK(F+2) = TP
STACK(F+3) = A
                J+0; JUMPCHKX; EMITO(MKS);                                                     13904000 T  0142:000:0
              IF FAULTLEVEL>LEVEL THEN FAULTLEVEL+LEVEL;                                       13905000 T  0142:000:2:0
              IF MODE=0 THEN FAULTLEVEL+1;                                                     13906000 T  0142:000:4:0
              DO BEGIN IF J>0 THEN STEPIT; J+J+1;                                              13907000 T  0142:000:6:0
                    SCATTERELBAT; A+ACCUM[1];                                                  13908000 T  0142:000:9:2
                    IF TP+REAL((Q="6INTOV")&(Q="6EXPOV")[46:47:1]&(Q="5INDEX"                  13909000 T  0142:001:0:1
                    )[45:47:1]&(Q="4ZERO0")[44:47:1]&(Q="4FLAG0")[43:47:1])=0                  13910000 T  0142:001:3:2
                        THEN ERR (61) ELSE                                                     13911000 T  0142:001:7:3
                      BEGIN IF TABLE(I+1)=ASSIGNOP THEN                                        13911100 T  0142:001:9:3
                         BEGIN STEPIT; COMMENT OVER THE +;                                     13911200 T  0142:002:6:1
                               IF GT1+STEPI>IDMAX AND GT1<FAULTID THEN ERR(3);                 13911300 T  0142:002:7:3
                               LEVELF+ELBAT[I].LVL;                                            13911400 T  0142:003:1:2
                    END ELSE COUNT+(ACCUM[1]+A).[12:6];                                        13911500 T  0142:003:3:2
                    IF LEVELF=LEVEL THEN ERR (1) ELSE                                          13912000 T  0142:003:5:3
                    BEGIN KLASSF:=FAULTID; ADDRSF:=GETSPACE(FALSE,1);                          13913000 T  0142:003:7:3
                          FORMALF+VONF+FALSE;                                                  13913100 T  0142:004:0:1
                          EMITL(TP); IF MODE=0 THEN BEGIN EMITL(0);                            13914000 T  0142:004:1:3
                          EMITPAIR(ADDRSF,STD);END; EMITN(ADDRSF);                             13915000 T  0142:004:4:1
                          EMIT ("≠E"); COMMENT C-TO-F;                                         13916000 T  0142:004:6:0
                          MAKEUPACCUM; E; STEPIT                                               13917000 T  0142:004:7:2
                END; END; END UNTIL ELCLASS≠COMMA;                                             13918000 T  0142:004:8:0
              EMITL(J); EMITL(13); EMITV(5);                                                   13919000 T  0142:004:9:3
            END FAULTDEC;                                                                      13920000 T  0142:005:2:0
                                                                                  142 IS   56 LONG, NEXT SEG    3



            COMMENT THIS SECTION CONTAINS THE BLOCK ROUTINE  ;                                 14000000 T  0003:095:9:3
            PROCEDURE BLOCK(SOP);                                                              14001000 T  0003:095:9:3
                    VALUE SOP;                                                                 14002000 T  0003:095:9:3
                    BOOLEAN SOP;                                                               14003000 T  0003:095:9:3
            COMMENT SOP IS TRUE IF THE BLOCK WAS CALLED BY ITSELF THROUGH THE                  14004000 T  0003:095:9:3
                    PROCEDURE DECLARATION-OTHERWISE IT WAS CALLED BY STATEMENT.                14005000 T  0003:095:9:3
                    THE BLOCK ROUTINE IS RESPONSIBLE FOR HANDLING THE BLOCK                    14006000 T  0003:095:9:3
                    STRUCTURE OF AN ALGOL PROGRAM-SEGMENTING EACH BLOCK,HANDLING               14007000 T  0003:095:9:3
                    ALL DECLARATIONS,DOING NECESSARY BOOKKEEPING REGARDING EACH                14008000 T  0003:095:9:3
                    BLOCK, AND SUPPLYING THE SCANNER WITH ALL NECESSARY INFORMATION            14009000 T  0003:095:9:3
                    ABOUT DECLARED IDENTIFIERS.                                                14010000 T  0003:095:9:3
                    IT ALSO WRITES EACH SEGMENT ONTO THE PCT;                                  14011000 T  0003:095:9:3
            BEGIN                                                                              14012000 T  0003:095:9:3
```

```
                    LABEL OWNERR,SAVERR,BOOLEANDEC,REALDEC,ALPHADEC,INTEGERDEC,                    14013000 T  0003:095:3
                                                                                          START OF SEGMENT *********  143
                          LABELDEC,DUMPDEC,LISTDEC,OUTDEC,INDEC,MONITORDEC,                        14014000 T  0143:000:0
                          SWITCHDEC,PROCEDUREDEC,ARRAYDEC,FORMATDEC,FILEDEC,                       14015000 T  0143:000:0
                         GOTSCHK,FIELDDEC,AUXMEMERR,                                %117-         14016000 P  0143:000:0
                          STREAMERR,DEFINEDEC,CALLSTATEMENT,HF,START;                              14017000 T  0143:000:0
                    SWITCH DECLSW:= OWNERR,SAVERR,BOOLEANDEC,REALDEC,ALPHADEC,INTEGERDEC,          14018000 T  0143:000:0
                                    LABELDEC,DUMPDEC,LISTDEC,OUTDEC,INDEC,MONITORDEC,              14019000 T  0143:002:1
                                    SWITCHDEC,PROCEDUREDEC,ARRAYDEC,FORMATDEC,FILEDEC,             14020000 T  0143:002:1
                                    STREAMERR,DEFINEDEC,AUXMEMERR,FIELDDEC;                %117-  14021000 P  0143:002:1
                    DEFINE NLOCS=10#,LOCBEGIN=PRT1#,                                              14022000 T  0143:001:3
                           LBP=[36:12]#,                                                          14023000 T  0143:001:3
                           SPACEITDOWN = BEGIN   WRITE(LINE[DBL]); WRITE(LINE[DBL])  END#;        14023100 T  0143:001:3
                     ARRAY TEDOC[0:7,0:127],LOCALS[0:NLOCS];                                      14024000 T  0143:001:3
STACK(F+2) = TEDOC
STACK(F+3) = LOCALS
                     ARRAY TENIL[0:7,0:127];                                                      14024100 T  0143:001:3
STACK(F+4) = TENIL
                      INTEGER OLDLASTADDRESS;                                                     14024200 T  0143:001:3
STACK(F+5) = OLDLASTADDRESS
                      INTEGER OLDENILPTR;                                                         14024300 T  0143:001:3
STACK(F+6) = OLDENILPTR
                     BOOLEAN GOTSTORAGE;                                                          14025000 T  0143:001:3
STACK(F+7) = GOTSTORAGE
                      BOOLEAN FWDTOG; COMMENT PREVIOUS FORWARD DECLARATION INDICATOR;             14025100 T  0143:001:3
STACK(F+10) = FWDTOG
                      INTEGER PINFOO,BLKAD;                                                       14026000 T  0143:001:3
STACK(F+11) = PINFOO
STACK(F+12) = BLKAD
                         COMMENT LOCAL TO BLOCK TO SAVE WHERE A PROCEDURE IS EMTERED              14027000 T  0143:001:3
                         IN INFO;                                                                 14028000 T  0143:001:3
                       REAL MAXSTACKO,LASTINFOT,RELAD,LO,TSUBLEVEL,STACKCTRO;                     14029000 T  0143:001:3
STACK(F+13) = MAXSTACKO
STACK(F+14) = LASTINFOT
STACK(F+15) = RELAD
STACK(F+16) = LO
STACK(F+17) = TSUBLEVEL
STACK(F+20) = STACKCTRO
                       INTEGER SGNOO,LOLD,SAVELO,PRTIO,NINFOO;                                    14030000 T  0143:001:3
STACK(F+21) = SGNOO
STACK(F+22) = LOLD
STACK(F+23) = SAVELO
STACK(F+24) = PRTIO
STACK(F+25) = NINFOO
                        INTEGER NCIIO;                                                            14031000 T  0143:001:3
STACK(F+26) = NCIIO
                         INTEGER PROAD ;                                                          14032000 T  0143:001:3
STACK(F+27) = PROAD
                           INTEGER NTEXTO;                                                        14032500 T  0143:001:3
STACK(F+30) = NTEXTO
                           INTEGER FIRSTXO;                                                       14033000 T  0143:001:3
STACK(F+31) = FIRSTXO
                          BOOLEAN FUNCTOGO,AJUMPO,FAULTOGO;                                       14034000 T  0143:001:3
STACK(F+32) = FUNCTOGO
STACK(F+33) = AJUMPO
STACK(F+34) = FAULTOGO
                          BOOLEAN SAVEPRTOGO, NEXTSAVE;                                           14034100 T  0143:001:3
```

```
STACK(F+35) = SAVEPRTOGO
STACK(F+36) = NEXTSAVE
                    BEGINCTR←BEGINCTR+1;
                        IF SOP THEN BLKAD←PROADO
                                                                                                14035000 T  0143:0019:3
                                                                                                14036000 T  0143:0021:2
                                                                                                14037000 T  0143:0021:3
                                                                                                14038000 T  0143:0021:3
                                                                                                14039000 T  0143:0021:3
                                                                                                14040000 T  0143:0021:3
                                                                                                14041000 T  0143:0021:3
                                                                                                14042000 T  0143:0021:3
                                                                                                14043000 T  0143:0021:3
                    ELSE BEGIN BLKAD:=GETSPACE(TRUE,-6); % SEG. DESCR.                          14044000 T  0143:0021:3
                               EMITV(BLKAD);                                                    14045000 T  0143:0024:1
                               EMITO(BFW);                                                      14046000 T  0143:0025:3
                               CONSTANTCLEAN                                                    14047000 T  0143:0026:0
                          END;                                                                  14048000 T  0143:0026:0
 MOVECODE(TEDOC,EDOC);  MOVECODE(TENIL,ENIL);                                                   14049000 T  0143:0026:1
 OLDLASTADDRESS ← LASTADDRESS;  LASTADDRESS ← -1;                                               14049100 T  0143:0031:2
 OLDENILPTR ← ENILPTR;  ENILPTR ← 0;                                                            14049200 T  0143:0033:2
ENILSPOT←0&CARDNUMBER[10:20:28] ;  ENILPTR←1 ;                                                  14049300 T  0143:0034:1
 MOVE(NLOCS,LOCBEGIN,LOCALS);                                                                   14050000 T  0143:0039:2
       FIRSTX0←FIRSTX;                                                                          14051000 T  0143:0041:2
 FIRSTX←4095;                                                                                   14052000 T  0143:0041:3
            IF LEVEL < 31 THEN LEVEL ← LEVEL + 1                                                14053000 T  0143:0042:1
                          ELSE FLAG(039);                                                       14053100 T  0143:0044:0
    LOLD←L;FUNCTOGO←FUNCTOG;AJUMPO←AJUMP;PRTIO←PRTI;SGNOO←SGNO;                                  14054000 T  0143:0047:3
    SAVELO := SAVEL;                                                          %118-             14055000 P  0143:0051:3
    AJUMP := FALSE;     % NO PENDING JUMPS IN THIS BLOCK YET.                %118-             14055100 C  0143:0052:0
    L := 0;             % START GENERATING CODE AT WORD 0, SYLLABLE 0.                          14055200 C  0143:0053:2
    OLDNINFOO := GLOBALNINFOO;  % REMEMBER WHERE PREVIOUS BLOCKS              %118-             14055250 C  0143:0053:3
                                % SYMBOLS BEGAN IN SYMBOL TABLE.              %118-             14055260 C  0143:0054:1
    GLOBALNINFOO := NINFOO := NEXTINFO;  % REMEMBER WHERE THE SYMBOLS                           14055300 C  0143:0054:1
                                         % FROM THIS BLOCK WILL GO            %118-             14055400 C  0143:0055:3
                                         % IN THE SYMBOL TABLE.               %118-             14055450 C  0143:0055:3
    NTEXTO ← NEXTTEXT;                                                                          14055500 T  0143:0055:3
  NCIIO←NCII;                                                                                   14056000 T  0143:0056:1
  NCII←0;                                                                                       14057000 T  0143:0057:2
  STACKCTRO←STACKCTR;                                                                           14058000 T  0143:0058:0
   FAULTOGO←FAULTOG; FAULTOG←                                                                   14058100 T  0143:0058:1
GOTSTORAGE←FALSE;                                                                               14059000 T  0143:0059:3
       SAVEPRTOG := (SAVEPRTOGO := SAVEPRTOG) OR SOP.[46:1];                                    14059100 T  0143:0060:1
IF LISTER OR SEGSTOG THEN SEGMENTSTART;                                                         14060000 T  0143:0063:2
 SGNO←SGAVL; SGAVL←SGAVL+1;                                                                     14061000 T  0143:0065:2
 ELBAT[I].CLASS←SEMICOLON;                                                                      14062000 T  0143:0067:2
START: IF TABLE(I)≠SEMICOLON                                                                    14063000 T  0143:0069:3
        THEN                                                                                    14064000 T  0143:0070:1
          BEGIN                                                                                 14065000 T  0143:0071:2
            FLAG(0);                                                                            14066000 T  0143:0071:3
            I←I-1                                                                               14067000 T  0143:0072:1
          END;                                                                                  14068000 T  0143:0072:1
       GTA1[0]←J←0;                                                                             14069000 T  0143:0073:3
       IF SPECTOG                                                                               14070000 T  0143:0075:3
          THEN                                                                                  14071000 T  0143:0075:3
            BEGIN                                                                               14072000 T  0143:0075:3
              IF BUP=PJ                                                                         14073000 T  0143:0076:0
                THEN                                                                            14074000 T  0143:0076:1
                  BEGIN                                                                         14075000 T  0143:0077:2
```

```
                                BEGIN LABEL GETLP;                                                  14076000 T  0143:007?3
PRT(1103) = *SEGMENT DESCRIPTOR*

                                                                                  START OF SEGMENT **********  144
                                      IF STREAMTOG THEN F←0 ELSE                                    14077000 T  0144:000?0
                                   F←FZERO;                                                         14078000 T  0144:000?3
                                 BUP←LASTINFO;                                                      14079000 T  0144:000?1
                                   DO                                                               14080000 T  0144:000?3
                                    BEGIN                                                           14081000 T  0144:000?0
                                 IF NOT STREAMTOG THEN                                              14082000 T  0144:000?0
                                      BUP←LASTINFO;                                                 14083000 T  0144:000?1
                                      GETLP: G←TAKE(BUP);                                           14084000 T  0144:000?3
                                             IF K←G.ADDRESS≠PJ                                      14085000 T  0144:000?2
                                               THEN                                                 14086000 T  0144:000?1
                                                 BEGIN                                              14087000 T  0144:000?2
IF BUP ≠ BUP:=BUP= TAKE(BUP + 1).PURPT THEN                                     %102-               14088000 P  0144:000?3
                                                   GO TO GETLP                                      14089000 T  0144:001?1
                                                 END;                                               14090000 T  0144:001?1
                                 IF TYPEV←G.CLASS=FRMTID OR TYPEV=SUPERFRMTID THEN                  14091000 T  0144:001?2
                                 G.ADDRESS←F←F+2                                                    14092000 T  0144:001?0
                                                   ELSE                                             14093000 T  0144:001?2
                                             IF TYPEV←G.CLASS≤INTARRAYID AND                        14094000 T  0144:001?2
                                                    TYPEV≥BOOARRAYID                                14095000 T  0144:002?3
                                                THEN                                                14096000 T  0144:002?3
                                                  BEGIN                                             14097000 T  0144:002?1
                                                    T1←G.INCR;                                      14098000 T  0144:002?2
                                                      GT1 ←N←TAKE(BUP+T1);                          14099000 T  0144:002?0
                                                    G.ADDRESS←F←F+N+1;                              14100000 T  0144:002?1
                                                    WHILE N≠0                                       14101000 T  0144:002?3
                                                    DO                                              14102000 T  0144:003?0
                                                     BEGIN                                          14103000 T  0144:003?2
                                                      IF T2←TAKE(BUP+T1+N)<0                        14104000 T  0144:003?2
                                                        THEN                                        14105000 T  0144:003?3
                                                         BEGIN                                      14106000 T  0144:003?0
                                                          T2←-T2;                                   14107000 T  0144:003?1
                                                          T2.LBP←4×(F-N)+2;                         14108000 T  0144:003?3
                                                           PUT(T2,BUP+T1+N)                         14109000 T  0144:003?1
                                                         END;                                       14110000 T  0144:004?0
                                                        N←N-1                                       14111000 T  0144:004?1
                                                     END                                            14112000 T  0144:004?2
                                                  END                                               14113000 T  0144:004?0
                                                ELSE                                                14114000 T  0144:004?1
                                                  G.ADDRESS←F←F+1;                                  14115000 T  0144:004?1
                                               PUT(G,BUP); G.INCR←GT1;                              14116000 T  0144:004?3
                                   IF FWDTOG THEN COMMENT CHECK CORRESPONDENCE W/ FWD;               14116100 T  0144:004?1
                                     BEGIN                                                          14116200 T  0144:004?1
                                       IF(GT1←TAKE(MARK+PJ)).CLASS ≠ G.CLASS                        14116300 T  0144:004?2
                 COMMENT CLASS ERROR;                    THEN FLAG(49);                             14116400 T  0144:005?3
                 COMMENT VALUE ERROR;  IF GT1.VO ≠ G.VO THEN FLAG(50)                               14116500 T  0144:005?3
                                     END ELSE                                                       14116600 T  0144:005?0
                                               PUT(G,MARK+PJ)                                       14117000 T  0144:005?1
                                            ;BUP←BUP-TAKE(BUP+1).PURPT                              14118000 T  0144:005?0
                                     END                                                            14119000 T  0144:006?1
                                    UNTIL PJ←PJ-1=0                                                 14120000 T  0144:006?3
                                   END;                                                             14121000 T  0144:006?1
PRT(1104) = *SEGMENT DESCRIPTOR*

                                                                                  144 IS   65 LONG, NEXT SEG  143
                                 SPECTOG←FALSE;                                                     14122000 T  0143:007?2
```

```
                    GO TO HF                                                                    14123000 T  0143:0079:3
                   END                                                                          14124000 T  0143:0080:0
              END;                                                                              14125000 T  0143:0080:0
    STACKCT ← 0;                                                                    %A          14125500 T  0143:0080:0
    WHILE STEPI=DECLARATORS                                                                     14126000 T  0143:0081:2
    DO                                                                                          14127000 T  0143:0081:3
     BEGIN                                                                                      14128000 T  0143:0082:1
         STOPDEFINE←(GTA1[J←J+1]←ELBAT[I].ADDRESS)≠MONITORV AND                                 14129000 T  0143:0082:1
                  GTA1[J]≠DUMPV;ERRORTOG←TRUE;                                              14130000 T  0143:0086:0
     END;                                                                                       14131000 T  0143:0088:1
IF J =0 THEN GO TO CALLSTATEMENT;                                                               14132000 T  0143:0089:2
     P2:=P3:=P4:=FALSE;                                                                         14133000 T  0143:0090:0
    GO TO DECLSW[GTA1[J]];                                                                      14134000 T  0143:0092:0
OWNERR:FLAG(20);J←J+1;GO TO REALDEC;                                                            14135000 T  0143:0094:0
SAVERR:FLAG(21);J←J+1;GO TO REALDEC;                                                            14136000 T  0143:0097:3
AUXMEMERR:FLAG(618);J:=J+1;GO TO REALDEC;                                       %117-           14136100 C  0143:0100:1
STREAMERR:FLAG(22);J←J+1;GO TO PROCEDUREDEC;                                                    14137000 T  0143:0103:3
REALDEC:P3←TRUE;ENTER(REALID);GO TO START;                                                      14138000 T  0143:0106:1
ALPHADEC:P3←TRUE;ENTER(ALFAID);GO TO START;                                                     14139000 T  0143:0109:2
BOOLEANDEC:P3←TRUE;ENTER(BOOID);GO TO START;                                                    14140000 T  0143:0111:2
INTEGERDEC:P3←TRUE;ENTER(INTID);GO TO START;                                                    14141000 T  0143:0113:2
       MONITORDEC:IF SPECTOG                                                                    14142000 T  0143:0115:2
                  THEN BEGIN COMMENT ERROR 463 MEANS THAT A MONITOR                             14143000 T  0143:0115:2
                             DECLARATION APPEARS IN THE SPECIFICATION                           14144000 T  0143:0115:3
                             PART OF A PROCEDURE;                                               14145000 T  0143:0115:3
                        FLAG(463);                                                              14146000 T  0143:0115:3
                       END;                                                                     14147000 T  0143:0116:1
            IF MERRIMAC THEN BEGIN FAULTDEC; GO GOTSCHK END; GO START;                          14148000 T  0143:0116:1
      DUMPDEC:IF SPECTOG                                                                        14149000 T  0143:0119:2
              THEN BEGIN COMMENT ERROR 464 MEANS A DUMP DECLARATION                             14150000 T  0143:0120:0
                         APPEARS IN THE SPECIFICATION PART OF A                                 14151000 T  0143:0120:1
                         PROCEDURE;                                                             14152000 T  0143:0120:1
                        FLAG(464);                                                              14153000 T  0143:0120:1
                       END;                                                                     14154000 T  0143:0121:3
                     DMUP;              GO TO START;                                            14155000 T  0143:0121:3
            ARRAYDEC:JUMPCHKX;ARRAE;GO TO GOTSCHK;                                              14156000 T  0143:0122:1
            FILEDEC:J←J+1;IODEC(11);GO TO GOTSCHK;                                              14157000 T  0143:0124:1
                INDEC:  IODEC(9);  IF G≠FORMATV THEN GO GOTSCHK;  GO START;%                    14158000 T  0143:0127:3
            OUTDEC:  IODEC(10);      IF G=FORMATV THEN GO TO START;                             14159000 T  0143:0130:1
            GOTSCHK:GOTSTORAGE← NOT SPECTOG OR GOTSTORAGE;GO TO START;                          14160000 T  0143:0133:2
            FORMATDEC:IF SPECTOG THEN ENTRY(FRMTID+REAL(GTA1[J-1]=SWITCHV)) ELSE                14161000 T  0143:0136:0
                       FORMATGEN;  GO TO START;                                                 14162000 T  0143:0139:2
            LISTDEC:                                                                            14163000 T  0143:0140:1
                BEGIN                                                                           14164000 T  0143:0141:2
                REAL SAVEINFO;                                                                  14165000 T  0143:0141:2
PRT(1105) = *SEGMENT DESCRIPTOR*

                                                                        START OF SEGMENT **********  145
STACK(F+37) = SAVEINFO
                LABEL START;                                                                    14166000 T  0145:0000:0
                IF  G:=GTA1[J]=LISTV AND G:=GTA1[J-1]=SWITCHV THEN                              14167000 T  0145:0000:0
                     BEGIN HANDLESWLIST; GO TO GOTSCHK END;                                     14168000 T  0145:0003:3
PRT(1106) = GOTSCHK
                IF SPECTOG THEN                                                                 14169000 T  0145:0007:2
                     BEGIN  ENTRY(LISTID); GO TO START  END;                                    14170000 T  0145:0007:3
                STOPENTRY:=STOPGSP:=TRUE; I:=I-1;                                               14171000 T  0145:0009:2
                DO   BEGIN                                                                      14172000 T  0145:0011:3
                     I:=I+1; JUMPCHKX;                                                          14173000 T  0145:0012:0
                     ENTRY(LISTID); IF ELCLASS≠LEFTPAREN THEN FLAG(31) ELSE STEPIT;             14174000 T  0145:0013:3
```

```
                    SAVEINFO:=LASTINFO; % IN CASE C-RELATIVE CONSTANTS ARE                    14175000 T  0145:001713
                                        % EMITTED B4 DOING THE PUT-TAKE BELOW.               14176000 T  0145:001810
                    F:=LISTGEN;                                                              14177000 T  0145:001810
                    PUT(TAKE(SAVEINFO)&(IF MODE=0 THEN F                                     14178000 T  0145:001912
                ELSE F:=GETSPACE(FALSE,SAVEINFO+1)) % LIST DESCR,                            14179000 T  0145:002113
                      [16:11],SAVEINFO);                                                     14180000 T  0145:002410
                    EMITSTORE(F,STD);                                                        14181000 T  0145:002513
                    END UNTIL ELCLASS# COMMA;                                                14182000 T  0145:002611
                STOPENTRY:=STOPGSP:=FALSE;                                                   14183000 T  0145:002713
            START:                                                                           14184000 T  0145:002912
                END LISTDEC;                                                                 14185000 T  0145:002912
PRT(1107) = *SEGMENT DESCRIPTOR*

                                                                                  145 IS   30 LONG, NEXT SEG  143
            GO TO START;                                                                     14186000 T  0143:014210
            LABELDEC:IF SPECTOG AND FUNCTOG THEN FLAG(24);                                   14187000 T  0143:014211
                    STOPENTRY←STOPGSP←TRUE;                                                  14188000 T  0143:014512
                    I←I-1;                                                                   14189000 T  0143:014610
                    DO                                                                       14190000 T  0143:014713
                      BEGIN                                                                  14191000 T  0143:014810
                         STOPDEFINE←TRUE;                                                    14192000 T  0143:014810
                       STEPIT;                                                               14193000 T  0143:014811
                         ENTRY(LABELID);                                                     14194000 T  0143:014912
                         PUTNBUMP(0)                                                         14195000 T  0143:015010
                      END                                                                    14196000 T  0143:015010
                    UNTIL ELCLASS#COMMA;                                                     14197000 T  0143:015011
                    STOPENTRY←STOPGSP←FALSE;                                                 14198000 T  0143:015210
                    GO TO START;                                                             14199000 T  0143:015312
            SWITCHDEC:                                                                       14200000 T  0143:015313
                   BEGIN                                                                     14201000 T  0143:015410
                     LABEL START;                                                            14202000 T  0143:015410
PRT(1110) = *SEGMENT DESCRIPTOR*

                                                                                  START OF SEGMENT **********  146
                      INTEGER GT1,GT2,GT4,GT5;                                               14203000 T  0146:000010
STACK(F+37) = GT1
STACK(F+40) = GT2
STACK(F+41) = GT4
STACK(F+42) = GT5
                     BOOLEAN TB1;                                                            14204000 T  0146:000010
STACK(F+43) = TB1
                     STOPENTRY←NOT SPECTOG;STOPGSP←TRUE;                                     14205000 T  0146:000010
                    SCATTERELBAT; GT1←0; TB1←FALSE;                                          14206000 T  0146:000113
                    IF LEVELF#LEVEL                                                          14207000 T  0146:000313
                       THEN                                                                  14208000 T  0146:000410
                            BEGIN                                                            14209000 T  0146:000411
                                 IF TAKE(LINKF+1)≥0                                          14210000 T  0146:000512
                                 THEN FLAG(1); PUT(-TAKE(LINKF+1),LINKF+1);                  14211000 T  0146:000610
                                 TB1←TRUE;GT2←ADDRSF;                                        14212000 T  0146:001011
                                 GT1← TAKEFRST; GT4←LASTINFO; LASTINFO←LINKF;                14213000 T  0146:001210
                                 STEPIT;GT5←NEXTINFO;NEXTINFO←LINKF+INCRF                    14214000 T  0146:001411
                                 END                                                         14215000 T  0146:001610
                            ELSE                                                             14216000 T  0146:001712
                    ENTRY(SWITCHID); STOPGSP←STOPENTRY←FALSE;IF SPECTOG THEN GO TO            14217000 T  0146:001712
                    START;                                                                   14218000 T  0146:002010
                    IF ELCLASS=ASSIGNOP                                                      14219000 T  0146:002011
                      THEN                                                                   14220000 T  0146:002112
                        BEGIN                                                                14221000 T  0146:002113
                           JUMPCHKNX;PUTNBUMP(L);G←L;                                        14222000 T  0146:002210
```

```
                        IF FORMALF != SWITCHGEN(TB1,GT1)                    %113-
                    THEN
                        BEGIN
                         JUMPCHKX;
                         STUFFF(GT1);
                         IF MODE>0
                           THEN
                        IF TB1 THEN GT1←GT2 ELSE
                              GT1:=GETSPACE(FALSE,LASTINFO+1); % SWITCH.
                         EMITSTORE(GT1,STD)
                        END;
                  END
                ELSE
                  BEGIN
                        IF ELCLASS≠FORWARDV THEN FLAG(33);
                        PUT(-TAKE(LASTINFO+1),LASTINFO+1);
                        PUTNBUMP(GT1:=GETSPACE(TRUE,LASTINFO+1));%SWITCH.
                        IF MODE >0 THEN GT1:=GETSPACE(FALSE,-1);%TEMP. STOR.
                              STEP,T;
                        FORMALF←TRUE
                  END;
              PUT(TAKE(LASTINFO)&REAL(FORMALF)[9:47:1]&GT1[16:37:11],LAST,NFO);
              IF TB1 THEN
                  BEGIN
                        NEXTINFO←GT5;
                        LASTINFO←GT4;
                    END;
          START;
          END SWITCHDEC;
PRT(1111) = *SEGMENT DESCRIPTOR*
                                                          146 IS   53 LONG, NEXT SEG  143
          GO TO START;
            DEFINEDEC;
              BEGIN LABEL START;
PRT(1112) = *SEGMENT DESCRIPTOR*
                                                                START OF SEGMENT **********  147
                REAL J,K,DINFO,LINKA,LINKB;                         %118-
STACK(F+37) = J
STACK(F+40) = K
STACK(F+41) = DINFO
STACK(F+42) = LINKA
STACK(F+43) = LINKB
              STOPENTRY←STOPGSP←TRUE;I←1=1;
                  DEFINING != BOOLEAN(REAL(DEFINING) & 1[47:47:1]);
               DO
                BEGIN
                  STOPDEFINE:=TRUE;
                  STEPIT; MOVE(9,ACCUM[1],GTA1);
                  K←COUNT+1; J←GTA1[0]; ENTRY(DEFINEDID);
                  GTA1[0]←J+"100000"; J←0;
                  DINFO ← LASTINFO;
                  IF ELCLASS=LEFTPAREN OR ELCLASS=LFTBRKET THEN
                  BEGIN
                       IF K > 62 THEN BEGIN ERR(141); GO START END;
                       DO BEGIN STOPDEFINE←TRUE;
                            STEPIT;
                            IF (J←J+1) > 9 THEN BEGIN ERR(172); GO START END;
                            MOVE(9, ACCUM[1], DEFINFO[(J-1)×10])
```

```
                DEFINEPARAM(DINFO+1, J);                                                14259085 T  0147:0028:1
                ACCUM[0] := 0 & DEFINEDID CLASS & 1 FORMAL;      %116-                  14259090 P  0147:0030:0
                LINKA ← LASTINFO; LINKB ← NEXTINFO;                                     14259094 T  0147:0033:3
                E;                                                                      14259096 T  0147:0035:2
                IF LASTINFO ≠ LINKB THEN % NEW INFO ROW ENTERED.                        14259098 T  0147:0035:3
                PUT(TAKE(LINKA)&(LASTINFO-LINKA)[27:40:8], LINKA);                      14259100 T  0147:0036:0
                STACKHEAD[SCRAM] ← TAKE(LASTINFO),LINK;                                 14259102 T  0147:0039:3
                STOPDEFINE ← TRUE;                                                      14259104 T  0147:0042:0
            END UNTIL STEPI≠COMMA;                                                      14259110 T  0147:0042:1
            IF ELCLASS≠RTPAREN AND ELCLASS≠RTBRKET THEN ERR(173);                       14259120 T  0147:0044:0
            STOPDEFINE←TRUE;                                                            14259130 T  0147:0047:2
            STEPIT;                                                                     14259140 T  0147:0048:0
            PUT(←TAKE(DINFO), DINFO);      % MARK AS PARAMETRIC                       14259150 T  0147:0048:1
            PUT(TAKE(LASTINFO) & 0[27:40:8], LASTINFO);                                 14259155 T  0147:0050:0
        END;                                                                            14259160 T  0147:0052:1
                IF ELCLASS≠RELOP OR ACCUM[1]≠"1=0000"                                   14260000 T  0147:0052:1
         THEN                                                                           14261000 T  0147:0054:0
            BEGIN                                                                       14262000 T  0147:0054:1
                    FLAG(45);                                                           14263000 T  0147:0055:2
                    COMMENT ERROR 45 IS NO = FOLLOWING DEFINE ID;                       14263100 T  0147:0056:0
              I←I-1;                                                                    14264000 T  0147:0056:0
            END;                                                                        14265000 T  0147:0057:2
        MACROID←TRUE;                                                                   14265900 T  0147:0057:2
        LASTINFO ← DINFO;                                                               14265930 T  0147:0058:0
        PUT(TAKE(DINFO) & NEXTTEXT[11:32:16], DINFO);                                   14265950 T  0147:0058:1
        DEFINEGEN(FALSE, J & DINFO[18:33:15]);                                          14266000 T  0147:0061:2
        MACROID←FALSE;                                                                  14266100 T  0147:0063:2
     END                                                                                14267000 T  0147:0063:3
    UNTIL STEPI≠COMMA;                                                                  14268000 T  0147:0063:3
                    DEFINING := BOOLEAN(REAL(DEFINING) & 0[47:47:1]);                   14268500 T  0147:0065:2
                START: STOPENTRY←STOPGSP←FALSE; END; GO TO START;                       14269000 T  0147:0067:2
PRT(1113) = *SEGMENT DESCRIPTOR*

                                                                                                147 IS    70 LONG, NEXT SEG  143
            FIELDDEC:                                                           %117-       14269020 C  0143:0157:3
              BEGIN                                                             %117-       14269040 C  0143:0158:0
                REAL SAVEINFO, SB, NB;                                          %117-       14269060 C  0143:0158:0
PRT(1114) = *SEGMENT DESCRIPTOR*

                                                                                                  START OF SEGMENT **********  148
STACK(F+37) = SAVEINFO
STACK(F+40) = SB
STACK(F+41) = NB
                BOOLEAN FOUNDLB; % TRUE IF LEFT-BRACKET WAS USED IN FIELD SPEC.%117-       14269080 C  0148:0000:0
STACK(F+42) = FOUNDLB
                LABEL EXIT, SAVEIT;                                             %117-       14269100 C  0148:0000:0
                STOPENTRY := STOPGSP := TRUE;                                   %117-       14269120 C  0148:0000:0
                I := J - 1;                                                     %117-       14269140 C  0148:0001:2
                DO                                                              %117-       14269160 C  0148:0002:1
                  BEGIN                                                         %117-       14269180 C  0148:0003:2
                    STOPDEFINE := TRUE;                                         %117-       14269200 C  0148:0003:2
                    STEPIT;                                                     %117-       14269220 C  0148:0003:3
                    ENTRY(FIELDID);                                             %117-       14269240 C  0148:0004:0
                    SAVEINFO := LASTINFO;                                       %117-       14269260 C  0148:0005:2
                    IF ELCLASS = RELOP AND ACCUM[1] = "1=0000" THEN             %117-       14269280 C  0148:0005:3
                      BEGIN                                                     %117-       14269300 C  0148:0007:3
                        IF STEPI = LFTBRKET THEN % REMEMBER THIS                %117-       14269320 C  0148:0008:0
                          BEGIN                                                 %117-       14269340 C  0148:0009:2
                            FOUNDLB := TRUE;                                    %117-       14269360 C  0148:0009:3
```

```
                    STEPIT;                                                     %117-          14269380 C  0148:0010:1
                  END                                                           %117-          14269400 C  0148:0011:2
                ELSE                                                            %117-          14269420 C  0148:0011:2
                  FOUNDLB := FALSE;                                             %117-          14269440 C  0148:0011:2
                IF ELCLASS = FIELDID THEN                                       %117-          14269442 C  0148:0013:3
                  BEGIN                                                         %117-          14269444 C  0148:0014:1
                    SB := ELBAT[I].SBITF;                                       %117-          14269446 C  0148:0015:2
                    NB := ELBAT[I].NBITF;                                       %117-          14269448 C  0148:0016:1
                    GO TO SAVEIT;                                               %117-          14269450 C  0148:0018:0
                  END;                                                          %117-          14269452 C  0148:0018:1
                IF ELCLASS = LITNO THEN                                         %117-          14269460 C  0148:0018:1
                  IF STEPI = COLON THEN                                         %117-          14269480 C  0148:0019:2
                    IF STEPI = LITNO THEN                                       %117-          14269500 C  0148:0020:1
                      IF (SB := ELBAT[I-2].ADDRESS) ×                           %117-          14269520 C  0148:0022:0
                            (NB := ELBAT[I].ADDRESS) ≠ 0 AND                    %117-          14269540 C  0148:0024:1
                            SB + NB ≤ 48 THEN                                   %117-          14269560 C  0148:0027:2
                          BEGIN                                                 %117-          14269580 C  0148:0028:1
                  SAVEIT:                                                       %117-          14269590 C  0148:0029:2
                            PUT(TAKE(SAVEINFO) & SB SBITF & NB NBITF,           %117-          14269600 C  0148:0029:2
                                SAVEINFO);                                      %117-          14269620 C  0148:0032:0
                            STEPIT;                                             %117-          14269640 C  0148:0032:1
                            IF FOUNDLB THEN % BETTER HAVE RIGHT BRACKET.        %117-          14269660 C  0148:0033:2
                              IF ELCLASS = RTBRKET THEN                         %117-          14269680 C  0148:0033:2
                                BEGIN                                           %117-          14269700 C  0148:0034:1
                                  STEPIT;                                       %117-          14269705 C  0148:0035:2
                                  GO TO EXIT;                                   %117-          14269710 C  0148:0035:3
                                END                                             %117-          14269715 C  0148:0036:0
                              ELSE                                              %117-          14269720 C  0148:0036:0
                            ELSE                                                %117-          14269740 C  0148:0036:0
                              GO TO EXIT;                                       %117-          14269760 C  0148:0036:1
                          END;                                                  %117-          14269780 C  0148:0036:1
              END;                                                              %117-          14269800 C  0148:0036:1
              FLAG(114);                                                        %117-          14269820 C  0148:0036:1
              DO STEPIT UNTIL ELCLASS = COMMA OR ELCLASS = SEMICOLON;           %117-          14269840 C  0148:0037:2
          EXIT:                                                                 %117-          14269860 C  0148:0040:1
            END                                                                 %117-          14269880 C  0148:0041:2
          UNTIL                                                                 %117-          14269900 C  0148:0041:2
            ELCLASS ≠ COMMA;                                                    %117-          14269920 C  0148:0041:2
          STOPENTRY := STOPGSP := FALSE;                                        %117-          14269940 C  0148:0042:0
        END;                                                                    %117-          14269960 C  0148:0043:3
PRT(1115) = *SEGMENT DESCRIPTOR*
                                                                                      148 IS   45 LONG, NEXT SEG  143
        GO TO START;                                                            %117-          14269980 C  0143:0159:2
      PROCEDUREDEC:                                                                            14270000 T  0143:0159:3
        BEGIN                                                                                  14271000 T  0143:0160:0
          LABEL START,START1;                                                                  14272000 T  0143:0160:0
PRT(1116) = *SEGMENT DESCRIPTOR*
                                                                                 START OF SEGMENT **********  149
      LABEL START2, DOITANYWAY;                                                                14273000 T  0149:0000:0
        COMMENT FWDTOG NOW GLOBAL TO BLOCK;                                                    14274000 T  0149:0000:0
            IF NOT SPECTOG THEN FUNCTOG←FALSE;                                                 14275000 T  0149:0000:0
       FWDTOG := NEXTSAVE := FALSE;                                                            14276000 T  0149:0001:3
              IF LASTENTRY≠0 THEN BEGIN JUMPCHKNX;CONSTANTCLEAN END;                           14276500 T  0149:0003:2
              MAXSTACKO← MAXSTACK;                                                             14277000 T  0149:0005:2
       IF G←GTA1[J←J-1]=STREAMV                                                                14278000 T  0149:0006:0
        THEN                                                                                   14279000 T  0149:0008:0
          BEGIN STREAMTOG←TRUE;                                                                14280000 T  0149:0008:1
            IF G←GTA1[J←J-1]=0 THEN TYPEV←STRPROCID                                            14281000 T  0149:0009:3
```

```
                                ELSE                                                                14282000 T  0149:0013:2
                                 BEGIN                                                              14283000 T  0149:0013:3
                                  IF TYPEV←PROCID +G>INTSTRPROCID OR                                14284000 T  0149:0014:0
                                     TYPEV <BOOSTRPROCID                                            14285000 T  0149:0015:3
                                   THEN FLAG(004);                                                  14286000 T  0149:0016:0
                               IF NOT SPECTOG THEN                                                  14287000 T  0149:0018:0
                                   FUNCTOG←TRUE;                                                    14288000 T  0149:0018:1
                                   CHKSOB                                                           14289000 T  0149:0019:3
                                 END                                                                14290000 T  0149:0019:3
      END                                                                                           14291000 T  0149:0020:1
      ELSE                                                                                          14292000 T  0149:0020:1
        IF G=0 THEN TYPEV←PROCID                                                                    14293000 T  0149:0022:1
              ELSE                                                                                  14294000 T  0149:0023:2
             IF (TYPEV:=REALSTRPROCID+G)=INTSTRPROCID THEN                                          14294100 T  0149:0025:2
             BEGIN NEXTSAVE:=TRUE;  TYPEV:=PROCID END ELSE                                          14295000 T  0149:0027:2
             IF TYPEV<BOOPROCID OR TYPEV>INTPROCID THEN FLAG(005)                                   14295100 T  0149:0030:0
             ELSE BEGIN IF (NEXTSAVE:=GTA1[J-1]=SAVEV) THEN J:=J-1;                                 14296000 T  0149:0035:2
                  IF NOT SPECTOG THEN FUNCTOG:=TRUE;  CHKSOB                                        14297000 T  0149:0036:1
                     END;                                                                           14298000 T  0149:0037:3
  IF SPECTOG                                                                                        14299000 T  0149:0037:3
    THEN                                                                                            14300000 T  0149:0037:3
      BEGIN                                                                                         14301000 T  0149:0038:0
          ENTRY(TYPEV); GO TO START2                                                                14302000 T  0149:0039:3
      END;                                                                                          14303000 T  0149:0039:3
   MODE←MODE+1;                                                                                     14304000 T  0149:0040:1
   LO←PROINFO;                                                                                      14305000 T  0149:0041:3
   SCATTERELBAT;                                                                                    14306000 T  0149:0042:0
COMMENT CHECK TO SEE IF DECLARED FORWARD PREVIOUSLY                                ;                14307000 T  0149:0042:0
   IF LEVELF=LEVEL                                                                                  14308000 T  0149:0042:0
        THEN IF KLASSF≠TYPEV THEN BEGIN FLAG(6); GO DOITANYWAY END ELSE                             14309000 T  0149:0045:3
      BEGIN                                                                                         14310000 T  0149:0046:0
IF G ← TAKE(LINKF+1) ≥ 0 THEN FLAG(006) ELSE PUT(-G,LINKF+1) ;                                      14310500 P  0149:0052:0
      XMARK(DECLREF); % PROCEDURE DECLARED FORWARD, MARK LAST      %116-                         14310501 C  0149:0056:0
                      % XREF ENTRY AS A DECLARATION,             %116-                         14311100 T  0149:0056:0
        IF REAL(NEXTSAVE)≠G,[3:1] THEN FLAG(051);                                                   14312000 T  0149:0058:1
        FWDTOG←TRUE;                                                                                14313000 T  0149:0059:3
        PROAD←ADDRSF;                                                                               14314000 T  0149:0060:0
        PROINFO←ELBAT[I];MARK←LINKF+INCRF;STEPIT                                                    14316000 T  0149:0062:1
      END                                                                                           14317000 T  0149:0063:2
    ELSE                                                                                            14318000 T  0149:0063:2
 DOITANYWAY:  BEGIN STOPENTRY←P2←TRUE;                                                              14319000 T  0149:0065:2
        ENTRY(TYPEV);     MARK←NEXTINFO;PUTNBUMP(0);                                                14320000 T  0149:0067:3
        PROINFO←TAKE(LASTINFO)& LASTINFO[35:35:13];PROAD←ADDRSF;                                    14321000 T  0149:0070:1
     P2←STOPENTRY←FALSE                                                                             14322000 T  0149:0070:1
      END;                                                                                          14323000 T  0149:0071:3
            IF LEVEL < 31 THEN LEVEL ← LEVEL + 1                                                    14323100 T  0149:0073:2
                          ELSE FLAG(039);                                                           14323200 T  0149:0075:3
            PJ ← 0;                                                                                 14324000 T  0149:0076:0
       IF STREAMTOG THEN STREAMWORDS;                                                               14325000 T  0149:0077:3
    IF ELCLASS=SEMICOLON THEN GO TO START1;                                                         14326000 T  0149:0078:1
   IF ELCLASS≠LEFTPAREN THEN FLAG(007);                                                             14327000 T  0149:0080:1
COMMENT: THE FOLLOWING 8  STATEMENTS FOOL THE SCANNER AND BLOCK,PUTTING                             14328000 T  0149:0080:1
        FORMAL PARAMETER ENTRIES IN THE ZERO ROW OF INFO;                                           14329000 T  0149:0080:1
      RR1←NEXTINFO;                                                                                 14330000 T  0149:0081:3
    LASTINFOT←LASTINFO; LASTINFO←NEXTINFO←1;                                                        14331000 T  0149:0083:3
             PUTNBUMP(0);                                                                           14332000 T  0149:0084:0
      PTOG←TRUE;    I←I+1;                                                                        14333000 T  0149:0086:0
    ENTRY(SECRET);
```

```
IF FWDTOG THEN BEGIN IF GT1 ← TAKE(MARK),[40:8] ≠ PJ THEN%                    14333100 T  0149:0087:2
                  FLAG(48); COMMENT WRONG NUMBER OF PARAMETERS;               14333200 T  0149:0090:0
                   COMMENT SO THAT WE DONT CLOBBER INFO; END ELSE             14333300 T  0149:0091:2
  PUT(PJ,MARK);                                                               14334000 T  0149:0091:2
           P←PJ;                                                              14335000 T  0149:0092:1
     IF ELCLASS≠RTPAREN                                                       14336000 T  0149:0093:3
       THEN FLAG(008);                                                        14337000 T  0149:0093:3
     IF STEPI≠SEMICOLON                                                       14338000 T  0149:0095:3
       THEN FLAG(009);                                                        14339000 T  0149:0096:0
COMMENT MARK PARAMETERS VALUE IF THERE IS A VALUE PART;                       14340000 T  0149:0097:3
   IF STEPI=VALUEV                                                            14341000 T  0149:0097:3
      THEN                                                                    14342000 T  0149:0098:0
        BEGIN                                                                 14343000 T  0149:0098:1
          DO                                                                  14344000 T  0149:0099:2
            IF STEPI≠SECRET                                                   14345000 T  0149:0100:0
              THEN FLAG(010)                                                  14346000 T  0149:0100:1
              ELSE                                                            14347000 T  0149:0101:3
                 BEGIN                                                        14348000 T  0149:0102:0
                    IF G←ELBAT[I].ADDRESS=0 OR G>PJ                           14349000 T  0149:0102:1
                     THEN                                                     14350000 T  0149:0105:2
                        FLAG(010);                                            14351000 T  0149:0105:3
                                    G←TAKE(ELBAT[I]);                         14352000 T  0149:0107:2
                                    PUT(G&1[10:47:1],ELBAT[I])                14353000 T  0149:0108:1
                 END                                                          14354000 T  0149:0110:0
          UNTIL                                                               14355000 T  0149:0110:1
             STEPI≠COMMA;                                                     14356000 T  0149:0110:1
   IF ELCLASS≠SEMICOLON                                                       14357000 T  0149:0112:0
      THEN FLAG(011)                                                          14358000 T  0149:0112:1
      ELSE STEPIT                                                             14359000 T  0149:0113:3
        END;I←I-1;                                                            14360000 T  0149:0114:1
                IF STREAMTOG                                                  14361000 T  0149:0116:1
                   THEN                                                       14362000 T  0149:0116:1
                     BEGIN                                                    14363000 T  0149:0116:1
                              BUP←PJ; SPECTOG←TRUE;GO TO START1               14364000 T  0149:0117:2
                     END                                                      14365000 T  0149:0119:2
                   ELSE                                                       14366000 T  0149:0119:2
                     BEGIN                                                    14367000 T  0149:0119:2
                       SPECTOG←TRUE;                                          14368000 T  0149:0119:3
                       BUP←0;                                                 14369000 T  0149:0120:1
                       IF ELCLASS≠DECLARATORS                                 14370000 T  0149:0121:2
                          THEN FLAG(012)                                      14371000 T  0149:0121:3
                     END;                                                     14372000 T  0149:0122:1
             START:PTOG←FALSE;LASTINFO←LASTINFOT;NEXTINFO←IF FWDTOG THEN RR1 ELSE  14373000 T  0149:0123:2
                       MARK←PJ+1;                                             14374000 T  0149:0127:2
             START1:PINFOO←NEXTINFO;                                          14375000 T  0149:0128:1
             START2: END;                                                     14376000 T  0149:0129:3
PRT(1117) = *SEGMENT DESCRIPTOR*

                                                                      149 IS  131 LONG, NEXT SEG  143
                  IF SPECTOG OR STREAMTOG                                     14377000 T  0143:0161:2
                    THEN                                                      14378000 T  0143:0161:2
                         GO TO START;                                         14379000 T  0143:0161:3
             COMMENT IF SPECTOG IS ON THEN THE BLOCK WILL PROCESS THE SPECIFICATION  14380000 T  0143:0162:1
                     PART SIMILARY TO DECLARATIONS WITH A FEW NECESSARY VARIATIONS;  14381000 T  0143:0162:1
             HF:                                                              14382000 T  0143:0162:1
                BEGIN                                                         14383000 T  0143:0163:2
                  LABEL START,STOP;                                           14384000 T  0143:0163:2
PRT(1120) = *SEGMENT DESCRIPTOR*

                                                                      START OF SEGMENT **********  150
```

```
IF STREAMTOG                                                                                                    14385000 T  0150:0000:0
  THEN BEGIN                                                                                                    14386000 T  0150:0000:0
     JUMPCHKNX;G←PROGDESCBLDR(CHAR,L,PROAD);PJ←P;                                                               14387000 T  0150:0000:1
                  PTOG←FALSE;                                                                                   14388000 T  0150:0003:3
            IF FUNCTOG                                                                                          14389000 T  0150:0004:1
              THEN                                                                                              14390000 T  0150:0004:1
            PUT((Z←TAKE(PROINFO))&LOCLID[2:41:7]&(PJ←PJ+1)[16:37:11]                                             14391000 T  0150:0004:1
                 , PROINFO);                                                                                    14392000 T  0150:0009:3
            IF STEPI=BEGINV                                                                                     14393000 T  0150:0010:0
              THEN                                                                                              14394000 T  0150:0010:1
                 BEGIN                                                                                          14395000 T  0150:0011:2
                   WHILE STEPI=DECLARATORS OR ELCLASS=LOCALV                                                    14396000 T  0150:0011:3
                      DO                                                                                        14397000 T  0150:0013:2
                       BEGIN                                                                                    14398000 T  0150:0014:1
                         IF ELBAT[I].ADDRESS=LABELV                                                             14399000 T  0150:0014:1
                          THEN                                                                                  14400000 T  0150:0015:3
                             BEGIN                                                                              14401000 T  0150:0016:0
                               STOPDEFINE←STOPGSP←STOPENTRY←TRUE;                                               14402000 T  0150:0016:1
DO BEGIN STOPDEFINE←TRUE;STEPIT;ENTRY(STLABID);PUTNBUMP(0) END UNTIL                                            14403000 T  0150:0018:0
              ELCLASS≠COMMA;STOPGSP←STOPENTRY←FALSE                                                             14404000 T  0150:0021:3
                                   END                                                                          14405000 T  0150:0023:2
                           ELSE                                                                                 14406000 T  0150:0024:0
                             BEGIN                                                                              14407000 T  0150:0024:0
                                      I←I+1;                                                                    14408000 T  0150:0024:1
                               ENTRY(LOCLID)                                                                    14409000 T  0150:0026:0
                             END                                                                                14410000 T  0150:0026:0
                                 END;                                                                           14411000 T  0150:0026:1
                               COMPOUNDTAIL                                                                     14412000 T  0150:0027:2
                            END                                                                                 14413000 T  0150:0027:2
                         ELSE                                                                                   14414000 T  0150:0027:3
                            STREAMSTMT        ;                                                               14415000 T  0150:0027:3
                  COMMENT THE FOLLOWING BLOCK CONSTITUTES THE STREAM PROCEDURE PURGE;                            14416000 T  0150:0028:1
                       BEGIN                                                                                    14417000 T  0150:0028:1
                       REAL NLOC,NLAB;                                                                          14418000 T  0150:0028:1
PRT(1121) = *SEGMENT DESCRIPTOR*

                                                                                                  START OF SEGMENT **********  151

STACK(F+37) = NLOC
STACK(F+40) = NLAB
                       DEFINE SES=18#,SED=6#,TRW=5#;                                                            14419000 T  0151:0000:0
                       DEFINE  RSA = 43 #;                                                                      14419100 T  0151:0000:0
                       DEFINE LOC=[36:12]#,LASTGT=[24:12]#;                                                     14420000 T  0151:0000:0
                       J←  LASTINFO;                                                                            14421000 T  0151:0000:0
                            NLOC←NLAB←0;                                                                        14422000 T  0151:0000:1
                       DO                                                                                       14423000 T  0151:0002:0
                       BEGIN                                                                                    14424000 T  0151:0002:0
                       IF(GT1←TAKE(J)).CLASS=LOCLID THEN                                                        14425000 T  0151:0002:0
                            BEGIN                                                                               14426000 T  0151:0004:0
                            IF BOOLEAN(GT1,FORMAL) THEN                                                         14427000 T  0151:0004:1
                                 BEGIN                                                                          14428000 T  0151:0005:3
                                 IF GT1<0 THEN                                                                  14429000 T  0151:0006:0
                                 PUT(TAKE(GT2←MARK+P-GT1,ADDRESS+1)&FILEID[2:41:7]                              14430000 T  0151:0006:1
                                      ,GT2);                                                                    14431000 T  0151:0011:3
                                 END                                                                            14432000 T  0151:0012:0
                            ELSE  NLOC←NLOC+1;                                                                  14433000 T  0151:0012:0
                            END                                                                                 14434000 T  0151:0014:0
                       ELSE                                                                                     14435000 T  0151:0014:0
                            BEGIN                                                                               14436000 T  0151:0014:0
```

```
     IF  GT1.ADDRESS≠0 THEN NLAB←NLAB+1;                                    14437000 T  0151:0014:1
     IF(GT3←TAKE(GIT(J))).LASTGT≠0 AND GT3.LOC =0 THEN                      14438000 T  0151:0017:3
          BEGIN                                                             14439000 T  0151:0021:3
          MOVE(9,INFO[0,J],ACCUM[0]);                                       14440000 T  0151:0022:0
          Q←ACCUM[1];                                                       14441000 T  0151:0025:2
          FLAG(267);                                                        14442000 T  0151:0026:0
          ERRORTOG←TRUE;                                                    14443000 T  0151:0026:1
          END;                                                              14444000 T  0151:0027:3
     END;                                                                   14445000 T  0151:0027:3
XREFDUMP(J); % DUMP XREF INFO                                       %116-  14445100 C  0151:0027:3
G←(GT2←TAKE(J+1)).PURPT;                                                    14446000 T  0151:0029:3
IF  GT1.[2:8] ≠ STLABID×2+1  THEN                                           14447000 T  0151:0032:0
     STACKHEAD[(0&GT2[12:12:36])MOD 125]←TAKE(J).LINK;                      14448000 T  0151:0034:1
END UNTIL J←J-G≤1;                                                          14449000 T  0151:0038:1
PUT( P&NLAB[7:42:6]&(NLOC+REAL(FUNCTOG))[1:42:6]&(LPRT+1)                   14450000 T  0151:0041:2
     [13:37:11],MARK);                                                      14451000 T  0151:0044:1
GT1←  L;  L ← FILETHING ;                                                   14451100 T  0151:0046:0
WHILE  L ≠ 4095   DO                                                        14451200 T  0151:0047:3
     BEGIN FILETHING ← GET(L);                                              14451300 T  0151:0049:2
          EMITC(PJ+1,RSA);                                                  14451400 T  0151:0050:1
          L ← FILETHING;                                                    14451500 T  0151:0052:0
     END;                                                                   14451600 T  0151:0052:1
L ← GT1;   FILETHING ← 4095 ;                                               14451700 T  0151:0055:2
IF FUNCTOG THEN                                                             14452000 T  0151:0056:1
     BEGIN                                                                  14453000 T  0151:0056:1
     EMITC(TAKE( PROINFO).ADDRESS,SES);                                     14454000 T  0151:0057:2
     EMITC(PJ+2,SED);                                                       14455000 T  0151:0059:2
     EMITC(1,TRW);                                                          14456000 T  0151:0060:1
     PUT(Z, PROINFO);                                                       14457000 T  0151:0061:3
     END;                                                                   14458000 T  0151:0062:1
EMIT(0);                                                                    14459000 T  0151:0062:1
STREAMWORDS;                                                                14460000 T  0151:0063:3
STREAMTOG←FALSE;                                                            14461000 T  0151:0064:0
IF LISTER AND FORMATOG THEN SPACEITDOWN;                                    14461500 T  0151:0064:1
END;                                                                        14462000 T  0151:0074:1
PRT(1122) = *SEGMENT DESCRIPTOR*

                                                                      151 IS   77 LONG, NEXT SEG  150
          LASTINFO←LASTINFOT;NEXTINFO←MARK+P+1;                             14463000 T  0150:0030:0
     END                                                                    14464000 T  0150:0032:1
ELSE                                                                        14465000 T  0150:0032:1
     BEGIN                                                                  14466000 T  0150:0032:1
 IF STEPI=FORWARDV                                                          14467000 T  0150:0033:2
   THEN                                                                     14468000 T  0150:0033:3
    BEGIN                                                                   14469000 T  0150:0034:0
      XREFIT(PROINFO,0,FORWARDREF); % WE NEED THIS SO WE CAN FIND           14469100 C  0150:0034:1
                                    % THE FORWARD DECL. DURING XREF          14469101 C  0150:0037:2
      PUT(-TAKE(G:=PROINFO.LINK+1) & REAL(NEXTSAVE)[3:47:1],G);             14470000 T  0150:0037:2
          PURGE(PINFOO);                                                    14471000 T  0150:0041:2
      STEPIT                                                                14472000 T  0150:0042:0
    END                                                                     14473000 T  0150:0042:0
   ELSE                                                                     14474000 T  0150:0042:1
    BEGIN                                                                   14475000 T  0150:0042:1
 PROADO←PROAD;                                                              14476000 T  0150:0043:2
     TSUBLEVEL←SUBLEVEL;SUBLEVEL←LEVEL ;STACKCTRO←STACKCTR;                 14477000 T  0150:0043:3
%                                                                           14478000 T  0150:0046:0
     COMMENT ADDITIONS MADE TO COMPILER TO INSURE THAT STACKCELLS           14478010 T  0150:0046:0
     COUNTER DOES NOT OVERFLOW FOR PROCEDURE DECLARATIONS;                  14478020 T  0150:0046:0
          IF MODE = 1 THEN FRSTLEVEL  ←LEVEL;                               14478030 T  0150:0046:0
```

```
                        MAXSTACK←STACKCTR←514 + REAL(FUNCTOG);                                  14478040 T  0150:004:0
                    IF ELCLASS = BEGINV THEN                                                    14479000 T  0150:004:3
                      IF TABLE(I+1) = DECLARATORS THEN                                          14480000 T  0150:005:1
                BEGIN                                                                           14481000 T  0150:005:1
                        BLOCK(TRUE & NEXTSAVE[46:47:1]);                                        14482000 T  0150:005:2
                      ; PURGE(PINFOO);                                                          14483000 T  0150:005:2
                        GO TO STOP END;                                                         14484000 T  0150:005:3
                         BEGIN                                                                  14485000 T  0150:005:0
                            JUMPCHKNX;                                                          14486000 T  0150:005:0
                            RELAD←L  ;                                                          14487000 T  0150:005:1
                            IF NEXTSAVE THEN FLAG(052);                                         14487010 T  0150:005:3
                    STMT;                                                                       14488000 T  0150:005:2
                        IF FAULTOG.[46:1] THEN BEGIN EMITL(10); EMITO(COM); END;                14488500 T  0150:005:3
                        HTTEOAP(FALSE,RELAD,PINFOO,PROAD);                                      14489000 T  0150:006:0
                         END;                                                                   14490000 T  0150:006:0
                STOP;                                                                           14491000 T  0150:006:0
                    SUBLEVEL←TSUBLEVEL;                                                         14492000 T  0150:006:0
                    STACKCTR←STACKCTRO;                                                         14493000 T  0150:006:1
                    IF LISTER AND FORMATOG THEN SPACEITDOWN;                                    14493500 T  0150:006:3
                      END;                                                                      14494000 T  0150:007:2
                      END;                                                                      14495000 T  0150:007:2
                    PROINFO←LO;                                                                 14496000 T  0150:007:2
                    IF JUMPCTR=LEVEL                                                            14497000 T  0150:007:0
                       THEN                                                                     14498000 T  0150:007:0
                              JUMPCTR←LEVEL-1;                                                  14499000 T  0150:007:1
                         LEVEL←LEVEL-1;                                                         14500000 T  0150:007:1
                        MODE←MODE-1;                                                            14501000 T  0150:007:3
                           MAXSTACK←MAXSTACKO;                                                  14502000 T  0150:008:2
            START;END;                                                                          14503000 T  0150:008:3
PRT(1123) = *SEGMENT DESCRIPTOR*

                                                                                    150 IS   83 LONG, NEXT SEG  143
                          GO TO START;                                                      14504000 T  0143:016:0
                CALLSTATEMENT;                                                              14505000 T  0143:016:1
                              JUMPCHKX;                                                     14506000 T  0143:016:2
                          IF SPECTOG THEN BEGIN                                             14507000 T  0143:016:3
                                        IF (PJ  ≠  BUP) THEN                                14507010 T  0143:016:0
                                         BEGIN                                              14507020 T  0143:016:2
                                           INTEGER II,SSCRAM,SCOUNT;                        14507030 T  0143:016:3
PRT(1124) = *SEGMENT DESCRIPTOR*

                                                                                    START OF SEGMENT **********  152

STACK(F+37) = II
STACK(F+40) = SSCRAM
STACK(F+41) = SCOUNT

                                           MOVE(10,ACCUM,INFO[31,240]);                     14507040 T  0152:000:0
                                           II :=I;SSCRAM:=SCRAM;SCOUNT:=COUNT;              14507050 T  0152:000:1
                                           FOR SCRAM := 0 STEP 1 UNTIL 124                  14507060 T  0152:000:2
                                         DO IF((I←STACKHEAD[SCRAM]) < 256)                  14507070 T  0152:000:2
                                             THEN IF I ≠ 0 THEN                             14507080 T  0152:000:2
                                             BEGIN ELBAT[76]:=INFO[0,I]&I[35:35:            14507090 T  0152:000:1
                                                     13];                                   14507095 T  0152:001:2
                                                   COUNT:=INFO[0,I+1],[12:6];               14507100 T  0152:001:1
                                         MOVE(COUNT,INFO[0,I],ACCUM);                       14507105 T  0152:001:2
                                                   I:=76; SCATTERELBAT;                     14507110 T  0152:001:0
                                                   FORMALF := TRUE;                         14507120 T  0152:001:2
                                                   KLASSF := REALID;                        14507130 T  0152:002:0
                                                   MAKEUPACCUM; E;                          14507140 T  0152:002:1
                                             END;                                           14507150 T  0152:002:3
```

```
                                                I+I;SCRAM+SSCRAM;COUNT+SCOUNT;                    14507160 T  0152:002:0
                                                 MOVE(10,INFO[31,240],ACCUM);                     14507170 T  0152:002:0
                                                BUP+PJ; FLAG(12);SPECTOG+TRUE;                    14507180 T  0152:002:2
PRT(1125) = START                               GO TO START;                                      14507190 T  0152:003:2
                                              END   ;                                             14507200 T  0152:003:3
PRT(1126) = *SEGMENT DESCRIPTOR*

                                                                                          152 IS   35 LONG, NEXT SEG  143
                                                FLAG(12);GO TO HF                                 14508000 T  0143:016:2
                                             END;                                                 14509000 T  0143:017:0
                                BEGINCTR + BEGINCTR+1;                                            14510000 T  0143:017:0
                                IF ERRORTOG                                                       14511000 T  0143:017:3
                                  THEN COMPOUNDTAIL                                               14512000 T  0143:017:3
                                  ELSE                                                            14513000 T  0143:017:0
                                    BEGIN                                                         14514000 T  0143:017:1
                                     STMT;                                                        14515000 T  0143:017:2
                                     IF ELCLASS+TABLE(I+1)=DECLARATORS                            14516000 T  0143:017:3
                                       THEN                                                       14517000 T  0143:017:3
                                         BEGIN                                                    14518000 T  0143:017:0
                                           ELBAT[I],CLASS+SEMICOLON;                              14519000 T  0143:017:1
                                           BEGINCTR+BEGINCTR+1;                                   14520000 T  0143:017:2
                                           GO TO START                                            14521000 T  0143:018:0
                                         END                                                      14522000 T  0143:018:1
                                       ELSE                                                       14523000 T  0143:018:1
                                         COMPOUNDTAIL                                             14524000 T  0143:018:1
                                    END;                                                          14525000 T  0143:018:2
                BEGIN                                                                             14526000 T  0143:018:3
                      RELAD+FIRSTX;                                                               14534000 T  0143:018:3
                 IF STACKCTR>MAXSTACK                                                             14535000 T  0143:018:2
                    THEN MAXSTACK+STACKCTR;                                                       14536000 T  0143:018:2
                         IF GOTSTORAGE OR JUMPCTR=LEVEL OR FAULTOG.[46:1]                         14537000 T  0143:018:1
                           THEN                                                                   14538000 T  0143:018:3
                              IF NOT(GOTSTORAGE OR FAULTOG.[46:1])                                14539000 T  0143:018:1
                                THEN                                                              14540000 T  0143:018:3
                                   BEGIN                                                          14541000 T  0143:018:1
                                     EMITV(BLOCKCTR);                                             14542000 T  0143:018:2
                                     EMITL(1);                                                    14543000 T  0143:019:0
                                     EMITO(SUB);                                                  14544000 T  0143:019:1
                                     EMITSTORE(BLOCKCTR,STD);                                     14545000 T  0143:019:3
                                     GOTSTORAGE+TRUE                                              14546000 T  0143:019:1
                                   END                                                            14547000 T  0143:019:1
                                ELSE                                                              14548000 T  0143:019:2
     BEGIN                                                                                        14549000 T  0143:019:2
       EMITL(10);                                                                                 14550000 T  0143:019:3
       EMITO(COM)                                                                                 14551000 T  0143:019:1
     END;                                                                                         14552000 T  0143:019:1
        FUNCTOG+FUNCTOGO;                                                                         14553000 T  0143:019:2
IF SOP                                                                                            14554000 T  0143:019:0
         THEN HTTEOAP(GOTSTORAGE,FIRSTX,NINFOO,BLKAD)                                             14555000 T  0143:019:0
  ELSE                                                                                            14556000 T  0143:019:3
    BEGIN                                                                                         14557000 T  0143:019:1
   IF LEVEL = 1 THEN EMITO(XIT)                                                                   14557500 T  0143:019:2
                ELSE BEGIN                                                                        14557600 T  0143:020:1
                     EMITV(ADDRSF := GETSPACE(TRUE,*6)); % SEG. DESCR.                            14558000 T  0143:020:3
                     EMITO(BFW);                                                                  14558500 T  0143:020:3
                     END;                                                                         14558600 T  0143:020:1
   CONSTANTCLEAN;                                                                                 14559000 T  0143:020:1
           IF GOTSTORAGE OR NCII>0 OR LEVEL=1                                                     14560000 T  0143:020:2
```

```
          OR FAULTOG.[46:1] THEN
             BEGIN
               ADJUST; RELAD←L;
               IF GOTSTORAGE OR FAULTOG.[46:1]
                  THEN BEGIN EMITV(BLOCKCTR); EMITL(1);
               EMITO(ADD);EMITSTORE(BLOCKCTR,STD);
                                  END;
IF LEVEL=1 THEN IF G←NCII+MAXSTACK-512>0 THEN DO EMITL(0) UNTIL G←G-1
=0;
                 PURGE(NINFOO);
               IF LEVEL=1 THEN IF FAULTLEVEL=1 THEN
                    BEGIN EMITPAIR(0,MDS); EMITO(CHS); END;
               BUMPL;
               EMITB(BBW,L,IF FIRSTX=4095 THEN 0 ELSE FIRSTX);
               CONSTANTCLEAN
                    END ELSE PURGE(NINFOO);
                 IF RELAD =4095 THEN RELAD←0;
               NEXTTEXT ← NTEXTO;
        G←PROGDESCBLDR(LDES-REAL(LEVEL=1),RELAD,BLKAD)
       END;
 ENILSPOT ←  1023 & CARDNUMBER[10:20:28];
  SEGMENT((L+3)DIV 4,SGNO,SGNOO);
```

```
ENILPTR ← OLDENILPTR;  LASTADDRESS ← OLDLASTADDRESS;
MOVECODE(TENIL,ENIL);
 MOVECODE(TEDOC,EDOC);L←LOLD;
          DOUBLE(SGNO,SGNOO,←,SGNOO,SGNO);
    IF NOT SOP AND LEVEL ≠ 1
   THEN
      BEGIN
        ADJUST;
        G←PROGDESCBLDR(LDES,L,ADDRSF);
        IF ELCLASS = FACTOP THEN
          BEGIN  COMMENT SPECIAL CASE FOR COBOL ONLY;




            STEPIT;
          END;
      END;
          IF JUMPCTR=LEVEL THEN JUMPCTR←LEVEL-1;
```

```
14561000 T  0143:0206:1
14562000 T  0143:0208:0
14563000 T  0143:0208:1
14564000 T  0143:0210:0
14564100 T  0143:0210:0
14565000 T  0143:0213:2
14566000 T  0143:0215:2
14567000 T  0143:0215:2
14568000 T  0143:0220:0
14569000 T  0143:0222:0
14569100 T  0143:0222:1
14569200 T  0143:0224:1
14570000 T  0143:0227:2
14571000 T  0143:0228:0
14572000 T  0143:0231:3
14573000 T  0143:0231:3
14574000 T  0143:0234:1
14574500 T  0143:0236:1
14575000 T  0143:0237:3
14576000 T  0143:0239:2
14576100 T  0143:0240:0
14577000 T  0143:0244:0
14578000 T  0143:0246:1
14579000 T  0143:0246:1
14580000 T  0143:0246:1
14581000 T  0143:0246:1
14582000 T  0143:0246:1
14583000 T  0143:0246:1
14584000 T  0143:0246:1
14585000 T  0143:0246:1
14586000 T  0143:0246:1
14587000 T  0143:0246:1
14588000 T  0143:0246:1
14589000 T  0143:0246:1
14590000 T  0143:0246:1
14591000 T  0143:0246:1
14592000 T  0143:0246:1
14593000 T  0143:0246:1
14594000 T  0143:0248:0
14595000 T  0143:0250:0
14596000 T  0143:0253:2
14597000 T  0143:0254:1
14598000 T  0143:0255:3
14599000 T  0143:0256:0
14600000 T  0143:0256:1
14601000 T  0143:0257:2
14601100 T  0143:0259:2
14601200 T  0143:0259:3
14601300 T  0143:0260:0
14601400 T  0143:0260:0
14601500 T  0143:0260:0
14601600 T  0143:0260:0
14601610 T  0143:0260:0
14601700 T  0143:0260:0
14601800 T  0143:0260:1
14602000 T  0143:0260:1
14603000 T  0143:0260:1
```

```
                    LEVEL←LEVEL-1;                                                          14604000 T  0143:026312
                    FUNCTOG←FUNCTOGO;                                                       14605000 T  0143:026411
                    AJUMP←AJUMPO;                                                           14606000 T  0143:026512
                    GLOBALNINFOO := OLDNINFOO;                                  %118-       14606100 C  0143:026610
                    PRTI←PRTIO;                                                             14607000 T  0143:026711
                       FIRSTX←FIRSTXO;                                                      14608000 T  0143:026713
                    SAVEL←SAVELO;                                                           14609000 T  0143:026810
                    STACKCTR←STACKCTRO;                                                     14610000 T  0143:026912
                 SAVEPRTOG := SAVEPRTOGO;                                                   14610100 T  0143:026913
                    NCII←NCIIO; FAULTOG←FAULTOGO AND(FALSE&FAULTLEVEL<LEVEL[46:47:1]);      14611000 T  0143:027011
                 END;                                                                       14612000 T  0143:027410
                 END BLOCK;                                                                 14613000 T  0143:027410
                                                                        143 IS  287 LONG, NEXT SEG    3



                  COMMENT THIS SECTION CONTAINS THE VARIABLE ROUTINE AND ITS SIDEKICKS;      15000000 T  0003:095913
                 PROCEDURE CLSMPMN(ELBATWORD,TYPEDPROC) ;                                   15001000 T  0003:095913
PRT(1127) = CLSMPMN
                 VALUE ELBATWORD,TYPEDPROC ;                                                15002000 T  0003:095913
                         REAL ELBATWORD ;                                                   15003000 T  0003:095913
                 BOOLEAN TYPEDPROC ;                                                        15003010 T  0003:095913
                          BEGIN COMMENT CALL SIMPLE MONITOR IS USED TO CALL PRINTI FOR      15004000 T  0003:095913
                               SIMPLE VARIABLES.  SEE THE MERRIMAC ROUTINE FOR A            15005000 T  0003:095913
                               DESCRIPTION OF PRINTI;                                       15006000 T  0003:095913
                               EMITPAIR(JUNK,SND);EMITO(MKS);          EMITV(JUNK);         15007000 T  0003:095913
                               EMITL(PASSTYPE(ELBATWORD));             EMITPAIR(GNAT(       15008000 T  0003:096211
                               POWERSOFTEN),LOD); PASSALPHA(ELBATWORD);                     15009000 T  0003:096410
                 EMITPAIR(GNAT(CHARI),LOD) ;                                                15010000 T  0003:096610
                 IF TYPEDPROC THEN PASSMONFILE(TAKE(GIT(ELBATWORD)),[27:11]) ELSE           15010010 T  0003:096713
                 PASSMONFILE(TAKE(GIT(ELBATWORD)),SVARMONFILE) ;                            15010020 T  0003:097011
                 EMITNUM(1&CARDNUMBER[1:4:44]);                              %109-          15011000 P  0003:097312
                         EMITV(GNAT(PRINTI));                                               15012000 T  0003:097512
                    END CLSMPMN;                                                            15013000 T  0003:097610



                 INTEGER PROCEDURE PASSTYPE(ELBATWORD);                                     15014000 T  0003:097611
                                 VALUE ELBATWORD ;                                          15015000 T  0003:097611
                                  REAL ELBATWORD ;                                          15016000 T  0003:097611
                  COMMENT PASSTYPE IS USED TO PASS THE TYPE OF VARIABLE BEING               15017000 T  0003:097611
                 MONITORED TO PRINTI;                                                       15018000 T  0003:097611
                 PASSTYPE←(ELBATWORD.CLASS=BOOPROCID) MOD 4;                                15019000 T  0003:097611



                 PROCEDURE PASSALPHA(ELBATWORD);                                            15020000 T  0003:098112
                         VALUE ELBATWORD ;                                                  15021000 T  0003:098112
                          REAL ELBATWORD ;                                                  15022000 T  0003:098112
                    BEGIN COMMENT PASSALPHA GENERATES CODE THAT PASSES THE ID               15023000 T  0003:098112
                         PARAMETER TO PRINTI;                                               15024000 T  0003:098112
                         DEFINE SIZEALPHA = RR9#;  COMMENT SIZEALPHA CONTAINS THE           15025000 T  0003:098112
                                                                  START OF SEGMENT ********** 153
                                                   LENGTH OF THE ALPHA FOR THE              15026000 T  0153:000010
                                                   VARIABLE DESCRIBED BY ELBATWORD;         15027000 T  0153:000010
```

```
            DEFINE INDEX = RR10#;  COMMENT INDEX CONTAINS THE INDEX                  15028000 T  0153:0000:0
                                   INTO INFO FOR ID, INFO[INDEX] = ID;               15029000 T  0153:0000:0
            DEFINE LTEMP = RR11#;  COMMENT LTEMP IS A TEMP FOR L;                    15030000 T  0153:0000:0
            EMITV(IF BOOLEAN(L.[46:1])                                               15031000 T  0153:0000:0
                  THEN CPLUS2                                                        15032000 T  0153:0000:0
                  ELSE CPLUS1);LTEMP←BUMPL;        EMITWORD(GETALPHA(                15033000 T  0153:0001:3
                  INFO[(INDEX←ELBATWORD,LINK+1),LINKR,INDEX,LINKC],                  15034000 T  0153:0005:3
                  IF SIZEALPHA←TAKE(INDEX).ALPHASIZE > 7                             15035000 T  0153:0009:2
                  THEN 7                                                             15036000 T  0153:0011:2
                  ELSE SIZEALPHA));                EMITB(BFW,LTEMP,L);               15037000 T  0153:0011:3
      END PASSALPHA;                                                                 15038000 T  0153:0014:1
                                                                          153 IS   15 LONG, NEXT SEG    3


COMMENT  THE FOLLOWING BLOCK HANDLES THE FOLLOWING CASES                             15039000 T  0003:0981:2
         OF SIMPLE VARIABLES:                                                        15040000 T  0003:0981:2
            1. V ← EXP        ,WHERE V IS FORMAL-CALL BY NAME.                       15041000 T  0003:0981:2
            2. V ← EXP        ,ALL V EXCEPT FORMAL-NAME.                             15042000 T  0003:0981:2
            3. V.[S:L] ← EXP  ,WHERE V IS FORMAL-CALL BY NAME.                       15043000 T  0003:0981:2
            4. V.[S:L] ← EXP  ,ALL V EXCEPT FORMAL-NAME.                             15044000 T  0003:0981:2
            5. V.[S:L]        ,ALL V.                                                15045000 T  0003:0981:2
            6. V              ,ALL V.                                                15046000 T  0003:0981:2
         CODE EMITED FOR THE ABOVE CASES IS AS FOLLOWS:                              15047000 T  0003:0981:2
            1. VN,EXP,M*,XCH,←.                                                      15048000 T  0003:0981:2
            2. EXP,M*,VL,←.                                                          15049000 T  0003:0981:2
            3. VN,DUP,COC,EXP,T,M*,XCH,←.                                            15050000 T  0003:0981:2
            4. VV,EXP,T,M*,VL,←.                                                     15051000 T  0003:0981:2
            5. ZEROL,VV,T .                                                          15052000 T  0003:0981:2
            6. VV .                                                                  15053000 T  0003:0981:2
            WHERE VN = DESC V                                                        15054000 T  0003:0981:2
                  EXP= ARITH. OR BOOLEAN EXPRESSION,AS REQUIRED.                     15055000 T  0003:0981:2
                  M* = CALL ON MONITOR ROUTINE,IF REQUIRED.                          15056000 T  0003:0981:2
                  VL = LITC V                                                        15057000 T  0003:0981:2
                  VV = OPDC V                                                        15058000 T  0003:0981:2
                  ←  = STORE INSTRUCTION(ISD,ISN,SND OR STD).                        15059000 T  0003:0981:2
                  T  = BIT TRANSFER CODE(DIA,DIB,TRB).                               15060000 T  0003:0981:2
                  ZEROL = LITC 0                                                     15061000 T  0003:0981:2
                  DUP,COC,XCH = THE INSTRUCTIONS DUP,COC,AND XCH.                    15062000 T  0003:0981:2
         OF COURSE, EXP WILL CAUSE RECURSION,IN GENERAL,AND THUS                     15063000 T  0003:0981:2
         THE PARAMETER P1 AND THE LOCALS CAN NOT BE HANDLED IN A                     15064000 T  0003:0981:2
         GLOBAL FASHION.                                                             15065000 T  0003:0981:2
         THE PARAMETER P1 IS USED TO TELL THE VARIABLE ROUTINE                       15066000 T  0003:0981:2
         WHO CALLED IT.  SOME OF THE CODE GENERATION AND SOME                        15067000 T  0003:0981:2
         SYNTAX CHECKS DEPEND UPON A PARTICULAR VALUE OF P1 .                        15068000 T  0003:0981:2
                                                                                     15069000 T  0003:0981:2
   PROCEDURE  VARIABLE(P1); REAL P1;                                                 15070000 T  0003:0981:2
         BEGIN                                                                       15071000 T  0003:0981:2
         REAL TALL,  COMMENT ELBAT WORD FOR VARIABLE;                                15072000 T  0003:0981:2
                                                                        START OF SEGMENT ********** 154
   STACK(F+2) = TALL
                     T1 ,  COMMENT 1ST INTEGER OF PARTIAL WORD SYNTAX;               15073000 T  0154:0000:0
   STACK(F+3) = T1
                     T2 ,  COMMENT 2ND INTEGER OF PARTIAL WORD SYNTAX;               15074000 T  0154:0000:0
   STACK(F+4) = T2
                     J  ;  COMMENT SUBSCRIPT COUNTER ;                              15075000 T  0154:0000:0
```

```
STACK(F+5) = J
                        REAL X, Z;                                                    15075500 T  0154:000:0
STACK(F+6) = X
STACK(F+7) = Z
                    REAL REMEMBERSEQNO; % REMEMBERS SEQUENCE NUMBER OF VARIABLE %116-      15075550 C  0154:000:0
STACK(F+10) = REMEMBERSEQNO
                                % ON LEFT HAND SIDE OF ASSIGNMENT SO WE %116-          15075551 C  0154:000:0
                                % CAN XREF IT CORRECTLY.                %116-          15075552 C  0154:000:0
                        LABEL EXIT;                                                   15076000 T  0154:000:0
                        TALL←ELBAT[I] ;                                               15077000 T  0154:000:0
                        IF   ELCLASS ≤ INTPROCID  THEN                                15078000 T  0154:000:2
                           BEGIN                                                      15079000 T  0154:000:3
                             IF TALL.LINK ≠PROINFO.LINK THEN                          15080000 T  0154:002:0
                                BEGIN ERR(211); GO TO EXIT END;                       15081000 T  0154:000:0
                 COMMENT  211  VARIABLE-FUNCTION IDENTIFIER USED OUTSIDE OF ITS SCOPE*;  15082000 T  0154:000:3
                        TALL ← TALL & (ELCLASS+4) [2:4:7] & 514 [16:37:11];           15083000 T  0154:000:3
                           END                                                        15084000 T  0154:000:2
                        ELSE  CHECKER(TALL);                                          15085000 T  0154:000:2
                        REMEMBERSEQNO := CARDNUMBER;                  %116-           15085100 C  0154:001:0
                        IF TALL.CLASS ≤ INTID  THEN                                   15086000 T  0154:001:2
                     BEGIN                                                            15087000 T  0154:001:0
                        LABEL  L1, EXIT ;                                             15088000 T  0154:001:1
PRT(1130) = *SEGMENT DESCRIPTOR*
                                                                        START OF SEGMENT ********** 155
                        DEFINE   FORMALNAME=[9:2]=2 #;                                15089000 T  0155:000:0
                  J ← ELCLASS ;                                                       15089010 T  0155:000:0
                        IF  STEPI= ASSIGNOP THEN                                      15090000 T  0155:000:1
                     BEGIN STACKCT ← 1;                                 %A            15091000 T  0155:001:3
                    XMARK(ASSIGNREF); % ASSIGNMENT TO SIMPLE VARIABLE.  %116-          15091100 C  0155:000:2
              L1:                                                                     15092000 T  0155:000:2
               IF  TALL.FORMALNAME THEN                                               15092020 T  0155:000:0
                     BEGIN                                                            15093000 T  0155:000:2
                     EMITN(TALL.ADDRESS);                                             15094000 T  0155:000:3
                     IF T1≠0 THEN BEGIN EMITO(DUP);EMITO(COC) END;                    15095000 T  0155:001:2
                     END                                                              15096000 T  0155:001:3
                     ELSE IF T1≠0 THEN EMITV(TALL.ADDRESS)                            15097000 T  0155:001:3
                  ; STACKCT ← REAL(T1≠0); STEPIT;                       %A            15098000 T  0155:001:3
                     IF TALL.CLASS =BOOID  THEN BEXP ELSE AEXP;                       15099000 T  0155:001:1
                     EMITD(48+T2 ,T1 ,T2);                                            15100000 T  0155:002:3
            IF TALL<0 THEN CLSMPMN(TALL,J≥BOOPROCID AND J≤INTPROCID) ;                15101000 T  0155:002:3
                  STACKCT ← 0;                                          %A            15101500 T  0155:002:3
                  GT1 ← IF  TALL.CLASS =INTID  THEN IF  P1= FS                        15102000 T  0155:002:0
                            THEN ISD ELSE ISN ELSE                                    15103000 T  0155:003:0
                            IF P1 = FS THEN STD ELSE SND ;                            15104000 T  0155:003:1
                  IF TALL.FORMALNAME  THEN                                            15105000 T  0155:003:3
                     BEGIN  EMITO(XCH); EMITO(GT1) END                                15106000 T  0155:003:1
                     ELSE   EMITPAIR(TALL.ADDRESS,GT1);                               15107000 T  0155:003:1
                  END                                                                 15108000 T  0155:004:1
               ELSE                                                                   15109000 T  0155:004:1
                  BEGIN                                                               15110000 T  0155:004:1
                  IF  ELCLASS= PERIOD THEN                                            15111000 T  0155:004:2
                     BEGIN  IF DOTSYNTAX(T1,T2) THEN GO TO EXIT ;                     15112000 T  0155:004:0
                            IF STEPI=ASSIGNOP  THEN                                   15113000 T  0155:004:0
                              BEGIN                                       %116-       15113100 C  0155:004:2
                                IF P1≠ FS  THEN                                       15114000 T  0155:004:3
                                BEGIN                                     %116-       15115000 P  0155:004:1
                                  ERR(201); % PARTIAL WORD NOT LEFT-MOST               15115100 C  0155:004:2
                                  GO TO EXIT;                             %116-       15115200 C  0155:004:3
```

```
                                    END;                                          %116-      15115300 C  0155:0048:0
                                    XREFIT(TALL,REMEMBERSEQNO,ASSIGNREF);         %116-      15116000 P  0155:0048:0
                                    GO TO L1;                                     %116-      15116100 C  0155:0050:1
                                  END;                                            %116-      15116200 C  0155:0051:2
                                                                                    %A       15117000 T  0155:0051:2
                     END ;                                                                   15118000 T  0155:0051:2
                  IF P1# FP THEN BEGIN ERR(202); GO TO EXIT END;                             15119000 T  0155:0051:2
          COMMENT  202  VARIABLE= A  VARIABLE APPEARS WHICH IS NOT FOLLOWED    *           15120000 T  0155:0053:3
                        BY A LEFT ARROW OR PERIOD                              *;          15121000 T  0155:0053:3
          COMMENT  201  VARIABLE= A PARTIAL WORD DESIGNATOR IS NOT THE         *           15122000 T  0155:0053:3
                        LEFT-MOST OF A LEFT PART LIST                          *;          15123000 T  0155:0053:3
                EMIT(TALL,T1,T2);                                             %A           15124000 T  0155:0053:3
                                                                              %A           15125000 T  0155:0055:2
                     END     ;                                                            15126000 T  0155:0055:2
             EXIT:   END  OF  BLOCK  OF  SIMPLE VARIABLES                                 15127000 T  0155:0055:2
                     ELSE                                                                 15128000 T  0155:0055:2
PRT(1131) = *SEGMENT DESCRIPTOR*
                                                                                  155 IS   56 LONG, NEXT SEG  154
                     COMMENT   THE FOLLOWING BLOCK HANDLES THESE CASES OF SUBSCRIPTED     15129000 T  0154:0014:0
                               VARIABLES:                                             15130000 T  0154:0014:0
                                  1. V[*]          ,ROW DESIGNATOR FOR SINGLE-DIMENSION. 15131000 T  0154:0014:0
                                  2. V[R,*]        ,ROW DESIGNATOR FOR MULTI-DIMENSION.  15132000 T  0154:0014:0
                                  3. V[R]          ,ARRAY ELEMENT,NAME OR VALUE.         15133000 T  0154:0014:0
                                  4. V[R].[S:L]    ,PARTIAL WORD DESIGNATOR, VALUE.      15134000 T  0154:0014:0
                                  5. V[R] ←        ,ASSIGNMENT TO ARRAY ELEMENT,         15135000 T  0154:0014:0
                                  6. V[R].[S:L] ←  ,ASSIGNMENT TO PARTIAL WORD,LEFT-MOST. 15136000 T  0154:0014:0
                               R IS A K-ORDER SUBSCRIPT LIST,I.E. R= R1,R2,...,RK,       15137000 T  0154:0014:0
                               IN THE CASE OF NO MONITORING ON V, THE FOLLOWING CODE     15138000 T  0154:0014:0
                               IS EMITTED FOR THE ABOVE CASES:                           15139000 T  0154:0014:0
                                     1. CASE #1 IS A SPECIAL CASE OF #2,NAMELY,SINGLE    15140000 T  0154:0014:0
                                        DIMENSION, THE CODE EMITTED IS:                  15141000 T  0154:0014:0
                                           VL,LOD .                                      15142000 T  0154:0014:0
                                        EXECUTION: PLACES ARRAY DESCRIPTER IN REG A.     15143000 T  0154:0014:0
                                     2. THIS CODE IS BASIC TO THE SUBSCRIPTION PROCESS.  15144000 T  0154:0014:0
                                        EACH SUBSCRIPT GENERATES THE FOLLOWING SEQUENCE  15145000 T  0154:0014:0
                                        OF CODE:                                         15146000 T  0154:0014:0
                                           AEXP,L*,IF FIRST SUBSCRIPT THEN VN ELSE CDC   15147000 T  0154:0014:0
                                             ,LOD.                                       15148000 T  0154:0014:0
                                        FOR A K-ORDER SUBSCRIPTION,K-1 SEQUENCE ARE      15149000 T  0154:0014:0
                                        PRODUCED, THE AEXP IN EACH SEQUENCE REFERS TO    15150000 T  0154:0014:0
                                        THE CODE PRODUCED BY THE ARITHMETIC EXPRESSION   15151000 T  0154:0014:0
                                        PROCEDURE FOR THE ACTUAL SUBSCRIPT EXPRESSIONS,  15152000 T  0154:0014:0
                                        L* REFERS TO THE CODE PRODUCED FOR SUBTRACTING    15153000 T  0154:0014:0
                                        NON-ZERO LOWER BOUNDS FROM THE SUBSCRIPT         15154000 T  0154:0014:0
                                        EXPRESSION(L* YIELDS NO CODE FOR ZERO BOUNDS).   15155000 T  0154:0014:0
                                        EXECUTION: PLACES ARRAY ROW DESCRIPTOR IN REG A  15156000 T  0154:0014:0
                                                 . THE SPECIFIC ROW DEPENDS UPON THE     15157000 T  0154:0014:0
                                                 VALUES OF THE K-1 SUBSCRIPTS.           15158000 T  0154:0014:0
                     FOR THE REMAINING CASES,                                          15159000 T  0154:0014:0
                        SEQUENCES OF CODE ARE EMITED AS IN CASE #2.                    15160000 T  0154:0014:0
                        HOWEVER,THE ACTUAL SEQUENCES ARE:                              15161000 T  0154:0014:0
                         ONE SEQUENCE ,(AEXP,L*),FOR THE 1ST SUBSCRIPT,                15162000 T  0154:0014:0
                         K-1 SEQUENCES,(IF FIRST SUBSCRIPT THEN VN                     15163000 T  0154:0014:0
                         ELSE CDC,LOD,AEXP,L*), FOR THE REMAINING                      15164000 T  0154:0014:0
                         SUBSCRIPTS,IF K>1.                                            15165000 T  0154:0014:0
                        AT THIS POINT, CASES #3-6 ARE DIFFERENTIATED                   15166000 T  0154:0014:0
                        AND ADDITION CODE,PARTICULAR TO EACH CASE,IS                   15167000 T  0154:0014:0
```

```
                                    EMITTED,                                                    15168000 T  0154:0014:0
                                 3. ADD THE SEQUENCE:                                           15169000 T  0154:0014:0
                                      IF FIRST SUBSCRIPT THEN VV ELSE COC.                      15170000 T  0154:0014:0
                                    EXECUTION: THE ARRAY ELEMENT IS PUT IN REG A.               15171000 T  0154:0014:0
                                 4. ADD THE SEQUENCE:                                           15172000 T  0154:0014:0
                                      IF FIRST SUBSCRIPT THEN VV ELSE COC,ZEROL,                15173000 T  0154:0014:0
                                        XCH,T.                                                  15174000 T  0154:0014:0
                                 5. ADD THE SEQUENCE:                                           15175000 T  0154:0014:0
                                      IF FIRST SUBSCRIPT THEN VN ELSE CDC,FXP,                  15176000 T  0154:0014:0
                                        XCH,*.                                                  15177000 T  0154:0014:0
                                 6. ADD THE SEQUENCE:                                           15178000 T  0154:0014:0
                                      IF FIRST SUBSCRIPT THEN VN ELSE CDC,DUP,LOD,              15179000 T  0154:0014:0
                                        FXP,T, XCH,*.                                           15180000 T  0154:0014:0
                              EXP,T,*,ZEROL,ETC. HAVE SAME MEANINGS AS DEFINED IN               15181000 T  0154:0014:0
                              SIMPLE VARIABLE BLOCK.                                ;           15182000 T  0154:0014:0
                         BEGIN                                                                  15183000 T  0154:0014:0
                              LABEL EXIT,LAST,NEXT ;                                            15184000 T  0154:0014:1
PRT(1132) = *SEGMENT DESCRIPTOR*
                                                                                  START OF SEGMENT ********** 156
                                 INTEGER THENUMBEROFDECLAREDDIMENSIONS;                         15184100 T  0156:0000:0
STACK(F+11) = THENUMBEROFDECLAREDDIMENSIONS
                              DEFINE NODIM = RR1#;  COMMENT NODIM CONTAINS THE NUMBER OF        15185000 T  0156:0000:0
                                                 DIMENSIONS OF A MONITORED SUBSCRIPTED          15186000 T  0156:0000:0
                                                 VARIABLE;                                      15187000 T  0156:0000:0
                              DEFINE TESTVARB = RR2#;  COMMENT TESTVARB CONTAINS THE            15188000 T  0156:0000:0
                                                   INDEX OF THE LAST ENTRY IN INFO              15189000 T  0156:0000:0
                                                   FOR A MONITORED SUBSCRIPTED                  15190000 T  0156:0000:0
                                                   VARIABLE;                                    15191000 T  0156:0000:0
                              DEFINE INC = RR3#;  COMMENT INC IS A COUNTER USED TO INDEX        15192000 T  0156:0000:0
                                                INTO INFO TO PICK OUT SPECIAL MONITOR           15193000 T  0156:0000:0
                                                INFORMATION;                                    15194000 T  0156:0000:0
                              DEFINE SPMON = [11:12]#;  COMMENT SPMON DESIGNATES THE BIT        15195000 T  0156:0000:0
                                                  POSITION OF THE SPECIAL MONITOR               15196000 T  0156:0000:0
                                                  INFORMATION FOR SUBSCRIPTED                   15197000 T  0156:0000:0
                                                  VARIABLES;                                    15198000 T  0156:0000:0
                              DEFINE OPBIT = [11: 1]#;  COMMENT OPBIT TELLS WHETHER TO          15199000 T  0156:0000:0
                                                  EMIT AN OPDC OR LITC FOR PASSING              15200000 T  0156:0000:0
                                                  THE SUBSCRIPTS FOR MONITORED                  15201000 T  0156:0000:0
                                                  SUBSCRIPTED VARIABLES.1 MEANS                 15202000 T  0156:0000:0
                                                  LITC, 0 MEANS OPDC;                           15203000 T  0156:0000:0
                              DEFINE LWRBND = RR4#;  COMMENT LWRBND HOLDS THE LOWER             15204000 T  0156:0000:0
                                                BOUND WORD FROM INFO FOR MONITORED              15205000 T  0156:0000:0
                                                SUBSCRIPTED VARIABLES;                          15206000 T  0156:0000:0
                              DEFINE SPMONADR = [12:11]#;  COMMENT SPMONADR CONTAINS            15207000 T  0156:0000:0
                                                   THE ADDRESS THAT WILL BE                     15208000 T  0156:0000:0
                                                   EMITTED IN AN OPDC OR LITC                   15209000 T  0156:0000:0
                                                   DEPENDING ON OPBIT;                          15210000 T  0156:0000:0
                                 BOOLEAN SPCLMON;  COMMENT SPCLMON IS A BOOLEAN THAT            15211000 T  0156:0000:0
STACK(F+12) = SPCLMON
                                                   IS SET TRUE IF THE VARIABLE IN               15212000 T  0156:0000:0
                                                   TALL IS SPECIAL MONITORED.                   15213000 T  0156:0000:0
                         ;                                                                      15214000 T  0156:0000:0
                         PROCEDURE M4(TALL,J);                                                  15215000 T  0156:0000:0
PRT(1133) = M4
                                VALUE TALL,J ;                                                  15216000 T  0156:0000:0
                                REAL TALL,J ;                                                   15217000 T  0156:0000:0
                                          BEGIN STACKCT + 1;                        %A          15217500 T  0156:0000:0
                                               IF J = 1                                         15218000 T  0156:0000:1
```

```
                    THEN BEGIN COMMENT FIRST TIME AROUND;                                  15219000 T  0156:0001:2
                         IF TALL < 0                                                       15220000 T  0156:0002:0
                         THEN BEGIN COMMENT TALL IS MONITORED;                             15221000 T  0156:0002:0
                                   EMITV(JUNK);   EMITO(XCH);                              15222000 T  0156:0003:2
                              END;                                                         15223000 T  0156:0004:1
                         EMITN(TALL,ADDRESS )                                              15224000 T  0156:0004:1
                         END                                                               15225000 T  0156:0005:2
                    ELSE BEGIN COMMENT NOT THE FIRST TIME AROUND;                          15226000 T  0156:0006:0
                         EMITO(CDC);                                                       15227000 T  0156:0006:1
                         IF TALL < 0                                                       15228000 T  0156:0007:2
                         THEN BEGIN COMMENT CALL SUBSCRIPT;                                15229000 T  0156:0007:3
                                   EMITV(JUNK);   EMITO(XCH);                              15230000 T  0156:0008:1
                              END;                                                         15231000 T  0156:0010:0
               END;      END;                                                %A          15232000 T  0156:0010:0


               IF  STEPI # LFTBRKET THEN BEGIN ERR(207);GO TO EXIT END;                   15233000 T  0156:0010:0
               THENUMBEROFDECLAREDDIMENSIONS ← TAKE(GIT(TALL)),[40:8];                    15233100 T  0156:0013:3
               J ← 0;                                                                     15234000 T  0156:0016:0
               STACKCT ← 0;                                                        %A     15234500 T  0156:0016:1
COMMENT   207  VARIABLE-MISSING LEFTBRACKET ON SUBSCRIPTED VARIABLE  *;                   15235000 T  0156:0017:3
               IF P1 > FP THEN TALL ← ABS(TALL) ELSE                                      15236000 T  0156:0017:3
               IF TALL < 0 THEN                                                           15237000 T  0156:0019:3
COMMENT   ****  MONITOR  FUNCTION M1  GOES  HERE ;                                       15238000 T  0156:0021:2
                    BEGIN COMMENT THIS MAY BE A MONITORED SUBSCRIPTED                     15239000 T  0156:0021:2
                         VARIABLE;                                                        15240000 T  0156:0021:3
                         EMITO(MKS);                                                      15241000 T  0156:0021:3
                         IF SPCLMON←TAKE(GIT(TALL)+1),SPMON # 0                           15242000 T  0156:0022:0
                         THEN BEGIN COMMENT THIS IS SPECIAL MONITORED;                    15243000 T  0156:0024:1
                              TESTVARB←(NODIM←TAKE(INC←GIT(TALL))                         15244000 T  0156:0026:0
                              ,NODIMPART)+INC;                                            15245000 T  0156:0026:1
                              DO IF BOOLEAN(LWRBND←TAKE(INC←INC+1)),                      15246000 T  0156:0029:3
                                   OPBIT                                                  15247000 T  0156:0032:0
                                THEN EMITL(LWRBND,SPMONADR)                               15248000 T  0156:0032:1
                                ELSE EMITV(LWRBND,SPMONADR)                               15249000 T  0156:0033:3
                              UNTIL INC ≥ TESTVARB                                        15250000 T  0156:0035:2
                         END;                                                             15251000 T  0156:0036:1
                    END;                                                                  15252000 T  0156:0037:3
          NEXT;     IF STEPI = FACTOP THEN                                              15253000 T  0156:0037:3
                    BEGIN                                                                 15254000 T  0156:0039:2
                         STLB ← 1;                                                        15254400 T  0156:0039:3
                         WHILE TABLE(I+1) = COMMA DO                                      15254500 T  0156:0040:0
                         BEGIN STEPIT;                                                    15254600 T  0156:0043:2
                              IF STEPI = FACTOP THEN STLB ← STLB+1 ELSE                   15254700 T  0156:0043:3
                              BEGIN ERR(204); GO TO EXIT END;                             15254800 T  0156:0046:1
                         END;                                                             15254900 T  0156:0048:0
                         IF J+STLB # THENUMBEROFDECLAREDDIMENSIONS THEN                   15255000 T  0156:0048:1
                              BEGIN ERR(203);GO EXIT END;                                 15256000 T  0156:0050:0
COMMENT   203  VARIABLE- THE NUMBER OF SUBSCRIPTS USED IN A ROW          *                15257000 T  0156:0051:3
                         ROW DESIGNATER  DOES NOT MATCH THE ARRAY        *                15258000 T  0156:0051:3
                         DECLARATION.                                    *;               15259000 T  0156:0051:3
                         IF STEPI # RTBRKET THEN                                          15260000 T  0156:0051:3
                              BEGIN ERR(204);GO EXIT END;                                 15261000 T  0156:0052:1
COMMENT   204  VARIABLE- COMPILER EXPECTS A  ]  IN A ROW DESIGNATER      *;               15262000 T  0156:0054:1
                         IF P1 # FA THEN                                                  15262500 T  0156:0054:1
```

```
                    IF STLB > 1 THEN FLAG(212) ELSE                                  15262600 T  0156:005:2
                    IF P1≠F1 AND P1≠FL THEN                                          15263000 T  0156:005:3
                    IF P1 = FP AND REL THEN ELSE                                     15263050 T  0156:006:0
                        BEGIN ERR(205); GO TO EXIT; END;                             15263100 T  0156:006:0
COMMENT  205  VARIABLE- A ROW DESIGNATER APPEARS OUTSIDE OF A FILL     *             15264000 T  0156:006:0
                    STATEMENT OR ACTUAL PARAMETER LIST,                  *;          15265000 T  0156:006:0
                    IF  J=0  THEN                                                    15266000 T  0156:006:0
                            EMITPAIR(TALL,ADDRESS,LOD);                              15267000 T  0156:006:1
COMMENT  *****  MONITOR FUNCTION M2 GOES HERE  ;                                    15268000 T  0156:006:1
                    IF  TALL < 0 THEN                                                15269000 T  0156:006:1
                    BEGIN COMMENT DO NOT MONITOR AFTER ALL;                          15270000 T  0156:006:3
                        EMITNUM(5&CARDNUMBER[1:4:44]);              %109-            15271000 P  0156:006:0
                        EMITN(GNAT(PRINT));                         %109-            15271100 C  0156:006:3
                    END;                                                             15272000 T  0156:007:2
                    IF P1 = FA THEN                                                  15272900 T  0156:007:2
                    FOR X ← 1 STEP 1 UNTIL STLB DO                                   15273000 T  0156:007:3
                    BEGIN IF (Z←TAKE(GIT(TALL)+J+X)),[35:11] > 1023                  15273100 T  0156:007:2
                        THEN EMITV(Z) ELSE EMIT(Z);                                  15273200 T  0156:007:0
                        IF Z.[23:10] = ADD THEN EMITO(CHS);                          15273300 T  0156:007:2
                    END;                                                             15273400 T  0156:008:3
                    STEPIT;                                                          15274000 T  0156:008:0
                    GO TO EXIT;                                                      15275000 T  0156:008:1
                END OF ROW DESIGNATOR PORTION  ;                                     15276000 T  0156:008:2
            AEXP;                                                                    15277000 T  0156:008:2
COMMENT  *****      MONITOR FUNCTION M3 GOES HERE  ;                                 15278000 T  0156:008:3
            IF TALL < 0 THEN EMITPAIR(JUNK,ISN);                                     15279000 T  0156:008:3
            J ← J + 1;                                                               15280000 T  0156:008:3
            IF(GT1 ← TAKE( GIT(TALL)+ J)),[35:13] ≠ 0 THEN                           15281000 T  0156:008:2
                BEGIN                                                                15282000 T  0156:009:0
                  IF GT1.[46:2] = 0 THEN EMIT(GT1)                                   15283000 T  0156:009:1
                                    ELSE EMITV(GT1,[35:11]) ;                        15284000 T  0156:009:1
                  EMIT(GT1.[23:12]);                                                 15285000 T  0156:009:2
                END OF LOWER BOUND ADJUSTMENT    ;                                   15286000 T  0156:009:0
            IF  ELCLASS = COMMA THEN                                                 15287000 T  0156:009:0
                BEGIN                                                                15288000 T  0156:009:2
COMMENT  *****      MONITOR FUNCTION M4 GOES HERE ;                                  15289000 T  0156:009:3
                    M4 (TALL,J);                                                     15290000 T  0156:009:3
                    EMITO(LOD) ;                                                     15291000 T  0156:010:1
                    IF J+1 > THENUMBEROFDECLAREDDIMENSIONS THEN                      15291100 T  0156:010:3
                        BEGIN ERR(208); GO TO EXIT END;                              15291200 T  0156:010:1
 COMMENT  208  VARIABLE- NUMBER OF SUBSCRIPTS DOES NOT MATCH ARRAY     *             15291300 T  0156:010:1
                    DECLARATION                                          *;          15291400 T  0156:010:1
                    GO TO NEXT;                                                      15292000 T  0156:010:1
                END OF SUBSCRIPT COMMA HANDLER ;                                     15293000 T  0156:010:2
            IF ELCLASS ≠ RTBRKET THEN BEGIN ERR(206);GO EXIT END;                    15294000 T  0156:010:2
COMMENT  206  VARIABLE- MISSING RIGHT BRACKET ON SUBSCRIPTED VARIABLE*;              15295000 T  0156:010:3
            IF J ≠ THENUMBEROFDECLAREDDIMENSIONS THEN                                15296000 T  0156:010:3
                    BEGIN ERR(208); GO TO EXIT END;                                  15297000 T  0156:010:0
                                                                                     15298000 T  0156:011:0
                                                                                     15299000 T  0156:011:0
            STACKCT ← 0;                                                %A           15299500 T  0156:011:0
            IF STEPI = ASSIGNOP  THEN                                                15300000 T  0156:011:1
               BEGIN                                                                 15301000 T  0156:011:3
                 XREFIT(TALL,REMEMBERSEQNO,ASSIGNREF); % ASSIGNMENT TO               15301100 C  0156:011:0
                                % SUBSCRIPTED VARIABLE,              %116-           15301200 C  0156:011:1
COMMENT  *****      MONITOR FUNCTION M4 GOES HERE  ;                                 15302000 T  0156:011:1
   LAST;            M4(TALL,J);                                                      15303000 T  0156:011:1
            IF  T1= 0 THEN                                                           15304000 T  0156:011:0
```

```
                    BEGIN IF P1= FR THEN GO TO EXIT END                                    15305000 T  0156:011:2
               ELSE BEGIN EMITO(DUP); EMITO(COC) END; STEPIT;       %WF                    15306000 T  0156:011:1
                    IF TALL.CLASS = BOOARRAYID THEN BEXP ELSE AEXP ;                       15307000 T  0156:012:2
                    EMITO(48-T2,T1,T2) ;                                                   15308000 T  0156:012:1
                    EMITO(XCH);                                                            15309000 T  0156:012:0
COMMENT  *****      MONITOR FUNCTION M6 GOES BEFORE EMITO(XCH);                            15310000 T  0156:012:2
                    IF TALL < 0                                                            15311000 T  0156:012:2
                    THEN BEGIN COMMENT STORE THE VALUE OF THE EXPRESSION                   15312000 T  0156:012:2
                              IN JUNK AND CALL PRINTI, THEN RECALL THE                     15313000 T  0156:012:0
                              VALUE FROM JUNK;                                             15314000 T  0156:012:0
                              EMITO(                                                       15315000 T  0156:012:0
                              IF TALL.CLASS = INTARRAYID                                   15316000 T  0156:012:1
                              THEN ISN                                                     15317000 T  0156:012:2
                              ELSE SND);                                                   15318000 T  0156:013:1
                                   IF P1 # FS                                              15319000 T  0156:013:3
                                   THEN EMITPAIR(JUNK,SND);                                15320000 T  0156:013:3
                              EMITL(J);      EMITL(PASSTYPE(TALL));                       15321000 T  0156:013:3
                              EMITPAIR(GNAT(POWERSOFTEN),LOD);                            15322000 T  0156:013:3
                              PASSALPHA(TALL);              EMITPAIR(GNAT(               15323000 T  0156:013:2
                              CHARI),LOD); PASSMONFILE(TAKE(GIT(TALL)),                    15324000 T  0156:013:1
                              ARRAYMONFILE);                                               15325000 T  0156:014:2
                              EMITNUM((IF SPCLMON THEN 3 ELSE 2)    %109-                  15326000 P  0156:014:3
                                   &CARDNUMBER[1:4:44]);            %109-                  15327000 P  0156:014:3
                              EMITV(GNAT(PRINTI));                  %109-                  15328000 P  0156:014:2
                              IF P1 # FS                                                   15329000 T  0156:014:0
                              THEN EMITV(JUNK);                                            15330000 T  0156:014:1
                              P1←0;          GO TO     EXIT;                               15331000 T  0156:014:0
                         END;                                                              15332000 T  0156:014:3
                    EMITO(IF TALL.CLASS = INTARRAYID  THEN                                 15333000 T  0156:014:3
                              IF  P1 = FS THEN ISD ELSE ISN ELSE                           15334000 T  0156:015:2
                              IF P1=FS THEN STD ELSE SND);                                 15335000 T  0156:015:1
                      P1←0  ;                                                              15336000 T  0156:015:2
                      GO TO EXIT ;                                                         15337000 T  0156:015:0
                    END OF ASSIGNMENT STATEMENT SUBSCRIPTED VARIABLES;                     15338000 T  0156:015:1
               IF  ELCLASS=PERIOD  THEN                                                    15339000 T  0156:015:1
                    BEGIN                                                                  15340000 T  0156:015:2
                      IF DOTSYNTAX(T1,T2) THEN GO TO EXIT;                                 15341000 T  0156:015:3
                      IF STEPI = ASSIGNOP THEN                          %116-              15342000 P  0156:016:3
                        IF P1 = FS THEN % PARTIAL WORD IS LEFT-MOST     %116-              15342100 C  0156:016:1
                          BEGIN                                         %116-              15342200 C  0156:016:3
                            XREFIT(TALL,REMEMBERSEQNO,ASSIGNREF); % PARTIAL              15342300 C  0156:016:0
                                  % WORD ASSIGNMENT TO SUBSCR, VAR,                       15342400 C  0156:016:1
                            GO TO LAST;                                 %116-              15342500 C  0156:016:1
                          END                                           %116-              15342600 C  0156:016:2
                           ELSE BEGIN ERR(209); GO EXIT END;                               15343000 T  0156:016:2
                      IF J=1 THEN EMITV(TALL.ADDRESS)ELSE EMITO(COC);                      15344000 T  0156:016:2
                    END                                                                    15345000 T  0156:017:1
               ELSE                                                                        15346000 T  0156:017:1
COMMENT  *****      MONITOR FUNCTION M10 GOES HERE   ;                                     15347000 T  0156:017:1
                    BEGIN COMMENT MONITOR FUNCTION M10;                                    15348000 T  0156:017:1
                         SPCLMON←P1 = FP OR ELCLASS ≥ AMPERSAND;                           15349000 T  0156:017:2
                         IF J = 1                                                          15350000 T  0156:017:3
                         THEN IF SPCLMON                                                   15351000 T  0156:017:3
                              THEN EMITV(TALL.ADDRESS)                                     15352000 T  0156:017:1
                              ELSE EMITN(TALL.ADDRESS)                                     15353000 T  0156:017:3
                         ELSE EMITO(IF SPCLMON                                             15354000 T  0156:017:3
                                    THEN COC                                               15355000 T  0156:018:2
```

```
                                        ELSE CDC);                                                  15356000 T  0156:018:0
                                IF TALL < 0                                                         15357000 T  0156:018:2
                                THEN BEGIN COMMENT DO NOT MONITOR AFTER ALL;                        15358000 T  0156:018:3
                                        EMITNUM(5&CARDNUMBER[1:4:44]);          %109-               15359000 P  0156:018:1
                                        IF SPCLMON                                                  15360000 T  0156:018:0
                                        THEN EMITV(GNAT(PRINTI))                                    15361000 T  0156:018:0
                                        ELSE EMITN(GNAT(PRINTI))                                    15362000 T  0156:018:3
                                     END;                                                           15363000 T  0156:018:3
                                IF P1 =FS THEN ERR(210);                                            15364000 T  0156:019:0
                                IF P1 = FI THEN P1←0;                                               15364500 T  0156:019:2
                                GO TO     EXIT;                                                     15365000 T  0156:019:0
                             END;                                                                   15366000 T  0156:019:1
                        IF P1=FS THEN BEGIN ERR(210); GO TO EXIT END ;                              15367000 T  0156:019:1
                COMMENT  210  VARIABLE=MISSING LEFT ARROW OR PERIOD.                    *;          15368000 T  0156:019:2
                        IF T1≠0 THEN BEGIN EMITI(0,T1,T2);                                          15369000 T  0156:019:2
                             IF P1≠FI THEN P1←0;                                                    15369100 T  0156:019:3
                        END;                                                                        15369200 T  0156:020:0
                        IF P1=FI THEN                                                               15369300 T  0156:020:0
                             IF ELCLASS≠COMMA AND ELCLASS≠RTPAREN                                   15369400 T  0156:020:1
                                THEN SIMPARITH;                                                     15369500 T  0156:020:0
                        IF P1=FI THEN P1←0;%                                                        15369600 T  0156:020:0
                                                                                        %A          15370000 T  0156:020:0
                COMMENT  *****     MONITOR FUNCTION  M9                          ;                  15371000 T  0156:020:0
                        IF TALL < 0                                                                 15372000 T  0156:020:0
                        THEN BEGIN COMMENT MONITOR FUNCTION M9;                                     15373000 T  0156:020:1
                                EMITNUM(5&CARDNUMBER[1:4:44]);          %109-                       15374000 P  0156:020:3
                                EMITV(GNAT(PRINTI));                    %109-                       15374100 C  0156:021:2
                        END ;                                                                       15375000 T  0156:021:1
                 EXIT;   STACKCT ← 0  END OF SUBSCRIPTED BLOCK;                          %A         15376000 T  0156:021:1
PRT(1134) = *SEGMENT DESCRIPTOR*
                                                                                        156 IS  215 LONG, NEXT SEG  154
    EXIT  ;  END  OF THE VARIABLE ROUTINE;                                                  15377000 T  0154:001:0
                                                                                        154 IS   21 LONG, NEXT SEG    3


COMMENT THIS SECTION GENERATES CODE FOR STREAM PROCEDURES;                                          16000000 T  0003:098:2
 PROCEDURE STREAMSTMT ;                                                                             16001000 T  0003:098:2
        BEGIN                                                                                       16002000 T  0003:098:2
        DEFINE LFTPAREN=LEFTPAREN#,LOC=[36:12]#,LASTGT=[24:12]#,                                    16003000 T  0003:098:2
                                                                                        START OF SEGMENT ********** 157
            LOCFLD=36:36:12#,LGTFLD=24:24:12#;                                                      16004000 T  0157:000:0
        DEFINE LEVEL=LVL#,ADDOP=ADOP#;                                                              16005000 T  0157:000:0
 DEFINE                                                                                             16006000 T  0157:000:0
        JFW  =  39#,   COMMENT  7,5,5,1  JUMP FORWARD UNCONDITIONAL  ;                              16007000 T  0157:000:0
         RCA =  40#, COMMENT    7,5,7,6  RECALL CONTROL ADDRESS      ;                              16008000 T  0157:000:0
        JRV  =  47#,  COMMENT   7,5,5,2  JUMP REVERSE UNCONDITIONAL  ;                              16009000 T  0157:000:0
        CRF  =  35#,  COMMENT    7,5,10,6 CALL REPEAT FIELD          ;                              16010000 T  0157:000:0
         BNS =  42#,  COMMENT   7,5,5,5  BEGIN LOOP                  ;                              16011000 T  0157:000:0
        NOP  =   1#,  COMMENT                                       ;                              16012000 T  0157:000:0
        ENS  =  41#,  COMMENT   7,5,5,6  END LOOP                   ;                              16013000 T  0157:000:0
        TAN  =  30#,  COMMENT   7,5,3,7  TEST FOR ALPHAMERIC        ;                              16014000 T  0157:000:0
        BIT  =  31#,  COMMENT   7,5,3,8  TEST BIT                   ;                              16015000 T  0157:000:0
         JFC =  37#,  COMMENT   7,5,5,3  JUMP FORWARD CONDITIONAL   ;                              16016000 T  0157:000:0
        SFD  =  06#,  COMMENT   7,5,7,8  SET DESTINATION ADDRESS    ;                              16017000 T  0157:000:0
        RSA  =  43#,  COMMENT   7,5,7,4  RECALL SOURCE ADDRESS      ;                              16018000 T  0157:000:0
        TRP  =  60#,  COMMENT   7,5,2,2  TRANSFER PROGRAM CHARACTERS ;                              16019000 T  0157:000:0
```

```
                    BSS  =   3#,  COMMENT   7,5,6,6  SKIP SOURCE BIT                ;   16020000 T  0157:000:0
                    BSD  =   2#,  COMMENT   7,5,6,5  SKIP DESTINATION BITS          ;   16021000 T  0157:000:0
                    SEC  =  34#,     COMMENT   7,5,10,1  SET COUNT                  ;   16022000 T  0157:000:0
                    JNS  =  38#;   COMMENT  7,5,5,7  JUMP OUT LOOP                  ;   16023000 T  0157:000:0
                COMMENT FIXC EMITS  BASICLY FORWARD JUMPS, HOWEVER IN THE CASE          16024000 T  0157:000:0
                        OF INSTRUCTIONS INTERPTED AS JUMPS BECAUSE OF A CRF ON          16025000 T  0157:000:0
                        A VALUE = 0 AND THE JUMP ≥ 64 SYLLABLES A JFW 1 AND             16026000 T  0157:000:0
                        A RCA L (L IS STACK ADDRESS OF A PSEUDO LABEL WHICH             16027000 T  0157:000:0
                        MUST ALSO BE MANUFACTURED) IS EMITTED.  ;                       16028000 T  0157:000:0
            PROCEDURE FIXC(S); VALUE S; REAL S;                                         16029000 T  0157:000:0
PRT(1:35) = FIXC
                    BEGIN                                                               16030000 T  0157:000:0
                    REAL SAVL,D,F;                                                      16031000 T  0157:000:0
                                                                          START OF SEGMENT **********  158
STACK(F+2) = SAVL
STACK(F+3) = D
STACK(F+4) = F
                    SAVL←L;                                                             16032000 T  0158:000:0
                F←GET(  S);                                                             16033000 T  0158:000:1
                    IF  D ← L ←( L←S) ←1≤63 THEN                                        16034000 T  0158:000:2
                        BEGIN                                                           16035000 T  0158:000:4
                        IF F=BNS THEN                                                   16036000 T  0158:000:5
                            BEGIN                                                       16037000 T  0158:000:6
                            S←GET(L←L←1);EMIT(NOP);EMIT(NOP);EMIT(S);D←D←2;             16038000 T  0158:000:6
                            END;                                                        16039000 T  0158:001:2
                        EMITC(D,F); L ← SAVL                                            16040000 T  0158:001:2
                        END                                                             16041000 T  0158:001:3
                    ELSE BEGIN                                                          16042000 T  0158:001:4
                        IF F≠JFW THEN BEGIN                                             16043000 T  0158:001:4
                        EMITC(1,F);                                                     16044000 T  0158:001:5
                        EMITC(1,JFW)    END ;                                           16045000 T  0158:001:6
                        EMITC(PJ←PJ+1,RCA);                                             16046000 T  0158:001:7
                        L ← SAVL;                                                       16047000 T  0158:001:9
                        ADJUST;                                                         16048000 T  0158:002:1
                        LPRT ← PROGDESCBLDR(2,L,0);                                     16049000 T  0158:002:2
                COMMENT  NOW ENTER  PSEUDO LABEL INTO INFO WITH ADDRESS=PJ-1;           16050000 T  0158:002:1
                        PUTNBUMP(0&(STLAB[D×2+1)                                        16051000 T  0158:002:1
                            [2:40:8]&PJ[16:37:11]&2[27:40:8]);                          16052000 T  0158:002:4
                        PUTNBUMP(0&(NEXTINFO-LASTINFO-1)[4:40:8]);                      16053000 T  0158:002:7
                        PUTNBUMP(0);                                                    16054000 T  0158:003:0
                        LASTINFO ← NEXTINFO-3;                                          16055000 T  0158:003:2
                        END;                                                            16056000 T  0158:003:2
                    END  FIXC  ;                                                        16057000 T  0158:003:2
                                                                  158 IS   35 LONG. NEXT SEG  157


                COMMENT EMITJUMP IS CALLED BY GOTOS AND JUMPCHAIN,                      16058000 T  0157:000:0
                        THIS ROUTINE WILL EMIT A JUMP IF THE DISTANCE IS ≤ 63           16059000 T  0157:000:0
                        SYLLABLES ,OTHERWISE, IT GETS A PRT CELL AND STUFFS THE         16060000 T  0157:000:0
                        STACK ADDRESS INTO THE LABEL ENTRY IN INFO AND EMITS AN         16061000 T  0157:000:0
                        RCA ON THIS STACK CELL. AT EXECUTION TIME ACTUAL PARAPART       16062000 T  0157:000:0
                        INSURES US THAT THIS CELL WILL CONATIN A LABEL DESCRIPTOR        16063000 T  0157:000:0
                        POINTING TO OUR LABEL IN QUESTION.  ;                           16064000 T  0157:000:0
            PROCEDURE EMITJUMP( E); VALUE E; REAL E;                                    16065000 T  0157:000:0
PRT(1:36) = EMITJUMP
```

```
                        BEGIN                                                                   16066000 T  0157:000:0
                        REAL T,D;                                                               16067000 T  0157:000:0
                                                                                          START OF SEGMENT **********  159
STACK(F+2) = T
STACK(F+3) = D
                        REAL ADDR;                                                              16068000 T  0159:000:0
STACK(F+4) = ADDR
                        IF ABS(                                                                 16069000 T  0159:000:0
                        D+(T+TAKE(GIT(E)).LOC)-L-1)≥64 THEN                                     16070000 T  0159:000:0
                             BEGIN                                                              16071000 T  0159:000:1
                             IF ADDR+TAKE(E).ADDRESS=0 THEN                                     16072000 T  0159:000:2
                                  BEGIN                                                         16073000 T  0159:000:2
                                  PUT(TAKE(E)&(ADDR+PJ+PJ+1)[16:37:11],E);                      16074000 T  0159:000:3
                                  LPRT ← PROGDESCBLDR(2,T,0);                                   16075000 T  0159:001:3
                                  END ;                                                         16076000 T  0159:001:3
                             EMITC(ADDR,RCA);                                                   16077000 T  0159:001:3
                             END                                                                16078000 T  0159:001:1
                        ELSE EMITC(D,IF D <0 THEN JRV ELSE JFW);                                16079000 T  0159:001:1
                        END  EMIT JUMP;                                                         16080000 T  0159:001:0
                                                                                          159 IS   21 LONG, NEXT SEG  157


                   COMMENT   WHEN JUMPCHAIN IS CALLED THERE IS A LINKEDLIST IN THE CODE      16081000 T  0157:000:0
                             ARRAY WHERE JFWS MUST BE PLACED. THE 1ST LINK IS POINTED        16082000 T  0157:000:0
                             TO BY THE LOC FIELD OF EACH LABEL ENTRY IN INFO. THE LAST       16083000 T  0157:000:0
                             LINK IS = 4096.  ;                                             16084000 T  0157:000:0
                  PROCEDURE JUMPCHAIN( E); VALUE E;REAL E;                                  16085000 T  0157:000:0
PRT(1137) = JUMPCHAIN
                        BEGIN                                                               16086000 T  0157:000:0
                        REAL SAVL ,LINK;                                                    16087000 T  0157:000:0
                                                                                          START OF SEGMENT **********  160
STACK(F+2) = SAVL
STACK(F+3) = LINK
                        SAVL ← L;                                                           16088000 T  0160:000:0
                        L ←  TAKE(GIT(E)).LASTGT ;                                          16089000 T  0160:000:1
                        WHILE L≠ 4095 DO                                                    16090000 T  0160:000:2
                             BEGIN                                                          16091000 T  0160:000:0
                             LINK ← GET(L);                                                 16092000 T  0160:000:0
                             EMITJUMP( E);                                                  16093000 T  0160:000:3
                             L ← LINK                                                       16094000 T  0160:000:1
                             END;                                                           16095000 T  0160:000:1
                        L←SAVL;                                                             16096000 T  0160:000:2
                        END  JUMPCHAIN  ;                                                   16097000 T  0160:000:3
                                                                                          160 IS   12 LONG, NEXT SEG  157


              COMMENT NESTS COMPILES THE NEST STATEMENT,                                    16098000 T  0157:000:0
                        A VARIABLE NEST INDEX CAUSES THE CODE,                              16099000 T  0157:000:0
                             CRF V, BNS 0 ,NOP,NOP, TO BE GENERATED INITIALLY.              16100000 T  0157:000:0
                        AT THE RIGHT PAREN THE BNS IS FIXED WITH THE LENGTH OF              16101000 T  0157:000:0
                        THE NEST (NUMBER OF SYLLABLES) IF THE LENGTH ≤63,OTHERWISE          16102000 T  0157:000:0
                        IT IS FIXED WITH A 1 AND THE NOPS REPLACED WITH JFW 1,              16103000 T  0157:000:0
                        RCA P.  THIS IS DONE BECAUSE THE VALUE OF V AT EXECUTION            16104000 T  0157:000:0
                        MAY = 0 AND THIS CODE CAUSES A JUMP AROUND THE NEST.                16105000 T  0157:000:0
```

```
                    JUMPOUT INFO IS REMEMBERED IN A RECURSIVE CELL AND                  16106000 T  0157!000!0
                    NEST LEVEL INCREASED BY ONE.                                       16107000 T  0157!000!0
                    WHEN THE RIGHT PAREN IS REACHED,(IF THE STATEMENTS IN              16108000 T  0157!000!0
                    THE NEST COMPILED), JOINFO IS CHECKED FOR THE EXISTANCE            16109000 T  0157!000!0
                    OF JUMPOUT STATEMENTS IN THE NEST.IF SO,THE THE JUMPS              16110000 T  0157!000!0
                    ARE FIXED BY FAKING TOTOS INTO COMPILING THE REQUIRED              16111000 T  0157!000!0
                    JUMPS.                                                             16112000 T  0157!000!0
                    FINALLY THE BNS IS FIXED,IF REQUIRED,AND NEST LEVEL                16113000 T  0157!000!0
                    AND JOINFO RESTORED TO THEIR ORIGINAL VALUES.  ;                   16114000 T  0157!000!0
          PROCEDURE NESTS;                                                             16115000 T  0157!000!0
PRT(1140) = NESTS
                BEGIN                                                                  16116000 T  0157!000!0
                LABEL EXIT;                                                            16117000 T  0157!000!0
                                                                    START OF SEGMENT ********** 161
                REAL JOINT,BNSFIX;                                                     16118000 T  0161!000!0
STACK(F+2) = JOINT
STACK(F+3) = BNSFIX
                IF ELCLASS≠LITNO THEN                                                  16119000 T  0161!000!0
                     BEGIN                                                             16120000 T  0161!000!1
                     EMITC(ELBAT[I].ADDRESS,CRF); BNSFIX← L;                           16121000 T  0161!000!2
                     EMIT ( BNS); EMIT(NOP);EMIT(NOP);                                 16122000 T  0161!000!3
                     END                                                               16123000 T  0161!000!0
                ELSE EMITC(ELBAT[I].ADDRESS,BNS);                                      16124000 T  0161!000!0
                IF STEPI ≠ LFTPAREN THEN BEGIN ERR(262); GO TO EXIT END;               16125000 T  0161!000!0
                NESTLEVEL←NESTLEVEL + 1;                                               16126000 T  0161!001!2
                JOINT ← JOINFO;                                                        16127000 T  0161!001!0
                JOINFO ← 0;                                                            16128000 T  0161!001!2
                DO   BEGIN                                                             16129000 T  0161!001!3
                     STEPIT; ERRORTOG ← TRUE; STREAMSTMT                               16130000 T  0161!001!0
                     END UNTIL ELCLASS ≠ SEMICOLON ;                                   16131000 T  0161!001!2
                IF ELCLASS ≠ RTPAREN THEN BEGIN ERR(262);GO TO EXIT END;               16132000 T  0161!001!2
                EMIT (  ENS);                                                          16133000 T  0161!001!3
                IF JOINFO ≠ 0 THEN                                                     16134000 T  0161!002!0
                     BEGIN                                                             16135000 T  0161!002!2
COMMENT              PREPARE TO CALL JUMPCHAIN FORJUMPOUTS;                            16136000 T  0161!002!3
                     ADJUST;                                                           16137000 T  0161!002!3
                     PUT(TAKE(GIT(JOINFO))&L[LOCFLD],GIT(JOINFO));                     16138000 T  0161!002!0
                     JUMPCHAIN(TAKE(JOINFO)&JOINFO[35:35:13]);                         16139000 T  0161!002!3
                     END;                                                              16140000 T  0161!002!0
                IF BNSFIX ≠ 0 THEN FIXC(BNSFIX);                                       16141000 T  0161!002!0
                NESTLEVEL ← NESTLEVEL-1;                                               16142000 T  0161!003!0
                JOINFO ← JOINT ;                                                       16143000 T  0161!003!3
EXIT;      END  NESTS  ;                                                               16144000 T  0161!003!0
                                                                    161 IS   36 LONG, NEXT SEG  157


COMMENT    LABELS HANDLES STREAM LABELS,                                               16145000 T  0157!000!0
           ALL LABELS ARE ADJUSTED TO THE BEGINING OF THE NEXT                         16146000 T  0157!000!0
           WORD (IN THE PROGRAMSTREAM),                                                16147000 T  0157!000!0
           IF A GO TO HAS NOT BEEN ENCOUNTERED BEFORE THE LABEL                        16148000 T  0157!000!0
           THEN THE NEST LEVEL FIELD IS ENTERED AND THE DEFINED BIT,                   16149000 T  0157!000!0
           [1:1], SET TO ONE. FOR DEFINED LABELS,IF WHERE A GO TO                      16150000 T  0157!000!0
           HAS APPEARED, A CHECK IS MADE THAT THE CURRENT NEST LEVEL                   16151000 T  0157!000!0
           MATCHES THE LEVEL OF THE LABEL,                                             16152000 T  0157!000!0
           MULTIPLE OCCURANCES ARE ALSO CHECKED FOR AND FLAGGED,                       16153000 T  0157!000!0
```

```
                    FINALLY,JUMPCHAIN IS CALLED TO FIX UP ANY FORWARD GO TOS          16154000 T  0157:000010
                    AND GET A PRT LOCATION FOR ANY JUMPS ≥64 SYLLABLES.  ;            16155000 T  0157:000010
               PROCEDURE LABELS;                                                      16156000 T  0157:000010
PRT(1141) = LABELS
                    BEGIN                                                             16157000 T  0157:000010
                    ADJUST;                                                           16158000 T  0157:000010
                    GT1 ← ELBAT[I];                                                   16159000 T  0157:000011
                    XMARK(LBLREF); % MARK LABEL OCCURENCE FOR XREF          %116-      16159100 C  0157:000113
                    IF STEPI ≠ COLON THEN ERR(258)                                    16160000 T  0157:000513
                    ELSE                                                              16161000 T  0157:000713
                         BEGIN                                                        16162000 T  0157:000810
                         IF TAKE(GT2←GIT(GT1)).LOC  ≠ 0 THEN FLAG(259) ELSE           16163000 T  0157:000811
                         IF GT1>0 THEN                                                16164000 T  0157:001211
                              BEGIN                                                   16165000 T  0157:001313
                              PUT(=(TAKE(GT1)&NESTLEVEL[11:43:5]),GT1);               16166000 T  0157:001410
                              PUT(=L,GT2)                                             16167000 T  0157:001712
                              END                                                     16168000 T  0157:001713
                         ELSE                                                         16169000 T  0157:001810
                              BEGIN                                                   16170000 T  0157:001810
                              IF GT1.LEVEL≠NESTLEVEL THEN FLAG(257);                  16171000 T  0157:001811
                              PUT((=L)&TAKE(GT2)[LGTFLD],GT2);                        16172000 T  0157:002112
                              JUMPCHAIN(GT1);                                         16173000 T  0157:002410
                              END;                                                    16174000 T  0157:002512
                         END                                                          16175000 T  0157:002512
                    ;  STEPIT;                                                       16176000 T  0157:002512
                    END  LABELS  ;                                                    16177000 T  0157:002513

               COMMENT   IFS COMPILES IF STATEMENTS.                                  16178000 T  0157:002513
                         FIRST THE TEST IS COMPILED. NOTE THAT IN THE                 16179000 T  0157:002513
                         CONSTRUCTS  "SC RELOP DC" AND  "SC RELOP STRING" THAT        16180000 T  0157:002513
                         THE SYLLABLE EMITTED IS FETCHED FROM ONE OF TWO FIELDS       16181000 T  0157:002513
                         IN THE ELBAT WORD FOR THE RELATIONAL OPERATOR, OTHERWISE      16182000 T  0157:002513
                         THE CODE IS EMITTED STRAIGHTAWAY.                            16183000 T  0157:002513
                         A TEST IS MADE TO SEE WHETHER THE STATEMENT AFTER THE        16184000 T  0157:002513
                         "THEN" COULD POSSIBLY BE LONGER THAN 63 SYLLABLES,AND IF     16185000 T  0157:002513
                         SO, Z NOPS ARE EMITTED FOR FIXC IN CASE A RCA WILL HAVE      16186000 T  0157:002513
                         TO BE GENERATED.                                             16187000 T  0157:002513
                         THIS PROCEDURE DOES NO OPTIMAZATION IN THE CASES             16188000 T  0157:002513
                         IF THEN GO TO L,IF THEN STATEMENT ELSE GO TO L, OR           16189000 T  0157:002513
                         IF THEN GO TO L1 ELSE GO TO L2   ;                           16190000 T  0157:002513
                    PROCEDURE  IFS;   BEGIN                                           16191000 T  0157:002513
PRT(1142) = IFS
                     DEFINE COMPARECODE =[42:6]#,TESTCODE=[36:6]#,EQUALV=48#;         16192000 T  0157:002513
                                                                          START OF SEGMENT **********  162
                    LABEL  IFSB,IFTOG,IFSC,EXIT;                                      16193000 T  0162:000010
                    SWITCH IFSW ← IFSB,IFTOG,IFSC;                                    16194000 T  0162:000010
                    REAL ADDR,FIX1,FIX2  ;                                            16195000 T  0162:000411
STACK(F+2) = ADDR
STACK(F+3) = FIX1
STACK(F+4) = FIX2
                    ADDR←1  ;                                                         16196000 T  0162:000411
                    GO TO IFSW[STEPI -SBV+1] ;                                        16197000 T  0162:000513
                    IF ELCLASS=LOCLID THEN                                            16198000 T  0162:000912
                         BEGIN                                                        16199000 T  0162:000913
                         EMITC(ELBAT[I],ADDRESS,CRF);                                 16200000 T  0162:001010
```

```
                    ADDR←0;                                                                  16201000 T  0162↑0012↑0
                    END                                                                      16202000 T  0162↑0012↑1
               ELSE                                                                          16203000 T  0162↑0012↑1
                    IF ELCLASS=LITNO THEN ADDR ← ELBAT[I],ADDRESS                            16204000 T  0162↑0012↑1
               ELSE BEGIN ERR(250); GO TO EXIT END;                                          16205000 T  0162↑0014↑1
               IF STEPI ≠ SCV THEN BEGIN ERR(263);GO TO EXIT END;                            16206000 T  0162↑0017↑3
IFSC;                                                                                        16207000 T  0162↑0020↑1
               IF STEPI≠RELOP THEN BEGIN ERR(264);GO EXIT END;                               16208000 T  0162↑0021↑2
               IF STEPI=DCV THEN EMITC(ADDR,ELBAT[I-1],COMPARECODE)                          16209000 T  0162↑0023↑3
          ELSF IF ELCLASS=STRNGCON THEN                                                      16210000 T  0162↑0026↑1
                    BEGIN                                                                    16211000 T  0162↑0028↑1
                    IF ACCUM[1],[12:6]≠1 OR ELBAT[I-3],CLASS≠IFV THEN                        16211100 T  0162↑0029↑2
                         BEGIN ERR(271); GO EXIT END                                         16211200 T  0162↑0033↑2
                    ELSE EMITC(ACCUM[1],[18:6],ELBAT[I-1],TESTCODE)                          16211300 T  0162↑0034↑1
                    END                                                                      16211400 T  0162↑0037↑2
          ELSF IF ELCLASS=LOCLID THEN                                                        16212000 T  0162↑0038↑0
                    BEGIN                                                                    16212100 T  0162↑0039↑3
                    IF ELBAT[I-3],CLASS≠IFV THEN                                             16212200 T  0162↑0040↑0
                         BEGIN ERR(271);  GO EXIT END                                        16212300 T  0162↑0042↑0
                    ELSE BEGIN                                                               16212400 T  0162↑0043↑3
                         EMITC(0,ELBAT[I-1],TESTCODE); % RESET TFFF,                         16212500 T  0162↑0044↑0
                         EMITC(ELBAT[I],ADDRESS,CRF);                                        16212600 T  0162↑0046↑1
                         EMITC(0,ELBAT[I-1],TESTCODE); % COMPARE.                            16212700 T  0162↑0048↑0
                         END                                                                 16212800 T  0162↑0050↑1
                    END                                                                      16212900 T  0162↑0050↑1
          ELSF IF ACCUM[1]≠"5ALPHA" THEN                                                     16213000 T  0162↑0050↑1
                     BEGIN ERR(265);GO EXIT END                                              16213100 T  0162↑0052↑0
          ELSF IF ELBAT[I-1],COMPARECODE=EQUALV THEN EMITC(17,TAN)                           16214000 T  0162↑0055↑2
               ELSF BEGIN FLAG(270); ERRORTOG!←TRUE  END;                                    16214100 T  0162↑0058↑1
               GO IFTOG;                                                                     16215000 T  0162↑0061↑2
IFSB;          EMITC(1,BIT);                                                                 16216000 T  0162↑0061↑3
IFTOG;         IF STEPI≠THENV THEN BEGIN ERR(266); GO EXIT  END;                             16217000 T  0162↑0063↑2
               FIX1 ← L;                                                                     16218000 T  0162↑0065↑3
               EMIT(JFC);                                                                    16219000 T  0162↑0066↑1
               STEPIT;                                                                       16220000 T  0162↑0067↑2
               IF ELCLASS = BEGINV OR                                                        16221000 T  0162↑0067↑3
                    ELCLASS = IFV OR                                                         16222000 T  0162↑0068↑1
                    ELCLASS = LITNO OR                                                       16223000 T  0162↑0069↑3
                              ELCLASS = STLABID OR                                           16224000 T  0162↑0070↑1
                    ELCLASS = LOCLID AND TABLE(I+1) = LFTPAREN THEN                          16225000 T  0162↑0071↑3
                    BEGIN                                                                    16226000 T  0162↑0074↑1
                    EMIT (NOP);EMIT (NOP)                                                    16227000 T  0162↑0075↑2
                    END;                                                                     16228000 T  0162↑0076↑0
                    IF ELCLASS= ELSEV THEN ELSE                                              16228500 T  0162↑0076↑1
               STREAMSTMT;                                                                   16229000 T  0162↑0077↑3
               IF ELCLASS= ELSEV THEN                                                        16230000 T  0162↑0078↑1
                    BEGIN                                                                    16231000 T  0162↑0079↑3
                    FIX2 ← L;  EMIT(JFW);                                                    16232000 T  0162↑0080↑0
                    FIXC(FIX1);                                                              16233000 T  0162↑0081↑3
                    STEPIT;                                                                  16234000 T  0162↑0082↑1
                    STREAMSTMT;                                                              16235000 T  0162↑0083↑2
                    FIXC(FIX2);                                                              16236000 T  0162↑0083↑3
                    END                                                                      16237000 T  0162↑0084↑1
               ELSE FIXC(FIX1);                                                              16238000 T  0162↑0084↑1
          EXIT;END   IFS  ;                                                                  16239000 T  0162↑0086↑0
                                                                                  162 IS   89 LONG, NEXT SEG  157
```

```
                    COMMENT GOTOS HANDLES GO TO AND THE LAST PART OF JUMP OUT TO                    16240000 T  0157!002513
                            STATEMENTS.                                                            16241000 T  0157!002513
                            IF THE LABEL HAS BEEN ENCOUNTERED THEN EMITJUMP IS CALLED              16242000 T  0157!002513
                            AN PRODUCES A JRV OR RCA IN THE CASE OF JUMPS≥64 SYLLABL               16243000 T  0157!002513
                            ES. OTHERWISE, A LINK IS EMITTED POINTING ANY PREVIOUS                 16244000 T  0157!002513
                            GO TOS IN THE CASE OF FORWARD JUMPS.                                   16245000 T  0157!002513
                            FINALLY, IF THE NEST LEVEL IS DEFINED THEN IT IS CHECKED               16246000 T  0157!002513
                            AGAINST THE CURRENT LEVEL MINUS THE NUMBER OF LEVELS TO                16247000 T  0157!002513
                            BE JUMPED OUT. OTHERWISE,NEST LEVEL IS DEFINED.  ;                     16248000 T  0157!002513
                PROCEDURE GOTOS;                                                                   16249000 T  0157!002513
PRT(1143) = GOTOS
                    BEGIN                                                                          16250000 T  0157!002513
                    LABEL EXIT;                                                                    16251000 T  0157!002513
                                                                                      START OF SEGMENT ********** 163
                    IF STEPI ≠TOV THEN I←I=1 ;                                                     16252000 T  0163!000010
                    IF STEPI ≠ STLABID THEN BEGIN ERR(260);GO TO EXIT END;                         16253000 T  0163!000211
                    IF(GT2←TAKE(GIT(GT1←ELBAT[I]))),MON=1                                          16254000 T  0163!000513
                    OR GT2.LOC≠0 THEN EMITJUMP(GT1)                                                16255000 T  0163!000811
                    ELSE                                                                           16256000 T  0163!001112
                      BEGIN   PUT(0&L[24:36:12],GIT(GT1));                                         16257000 T  0163!001210
                        IF GT1>0 THEN                                                              16258000 T  0163!001512
                             BEGIN                                                                 16259000 T  0163!001513
                             PUT(=(TAKE(GT1)&(NESTLEVEL-JUMPLEVEL)[11:43:5]),GT1);                 16260000 T  0163!001610
                             EMITN(1023);                                                          16261000 T  0163!001912
                             END                                                                   16262000 T  0163!002010
                        ELSE                                                                       16263000 T  0163!002010
                             BEGIN                                                                 16264000 T  0163!002010
                             IF GT1.LEVEL ≠ NESTLEVEL-JUMPLEVEL THEN FLAG(257);                    16265000 T  0163!002011
                             EMIT(GT2,LASTGT);                                                     16266000 T  0163!002313
                        END;                                                                       16267000 T  0163!002512
                      END;                                                                         16268000 T  0163!002512
                    JUMPLEVEL←0 ;                                                                  16269000 T  0163!002512
                EXIT; END  GOTOS   ;                                                               16270000 T  0163!002513
                                                                                      163 IS   27 LONG, NEXT SEG 157


                 COMMENT  RELEASES COMPILES THE STREAM RELEASE STATEMENT.                          16271000 T  0157!002513
                          THE CODE GENERATED IS :                                                  16272000 T  0157!002513
                               SED  FILE                                                           16273000 T  0157!002513
                               RSA  0,                                                             16274000 T  0157!002513
                          AT EXECUTION TIME THIS CAUSES AN INVALID ADDRESS WHICH IS                16275000 T  0157!002513
                          INTERPETED BY THE MCP TO MEAN RELEASE THE FILE POINTED TO                16276000 T  0157!002513
                          BY THE DESTINATION ADDRESS.                                              16277000 T  0157!002513
                          THE MONITOR BIT IS SET IN INFO FOR THE LOCAL VARIABLE SO                 16278000 T  0157!002513
                          THAT ACUTAL PARAPART MAY BE INFORMED LATER THAT A FILE                   16279000 T  0157!002513
                          MUST BE PASSED FOR THIS FORMAL PARAMETER;                                16280000 T  0157!002513
                PROCEDURE RELEASES;                                                                16281000 T  0157!002513
PRT(1144) = RELEASES
                    IF STEPI ≠ LFTPAREN OR STEPI≠LOCLID OR STEPI ≠ RTPAREN OR                      16282000 T  0157!002513
                         (GT1←ELBAT[I-1]).FORMAL=0                                                 16283000 T  0157!002913
                    THEN   ERR(256)   ELSE                                                         16284000 T  0157!003113
                         BEGIN                                                                     16285000 T  0157!003313
                         EMITC( GT1.ADDRESS,SED);                                                  16286000 T  0157!003410
                         EMIT(FILETHING); FILETHING←L=1;                                           16287000 T  0157!003513
```

```
                    INFO[GT1,LINKR,GT1,LINKC].MON ← 1;                                  16288000 T  0157!003!3
                    END  RELEASES;                                                      16289000 T  0157!004!3


                COMMENT   INDEXS  COMPILE STATEMENTS BEGINING WITH SI,DI,CI,TALLY       16290000 T  0157!004!0
                          OR LOCALIDS .                                                 16291000 T  0157!004!0
                          THREE CASES PRESENT THEMSELVES,                               16292000 T  0157!004!0
                               LETING X BE EITHER OF  SI,DI,CI OR TALLY, THEY ARE!      16293000 T  0157!004!0
                               CASE I    LOCLID ← X                                    16294000 T  0157!004!0
                               CASE II   X ← X ...                                     16295000 T  0157!004!0
                               CASE III  X ← EITHER LOC,LOCLID,SC OR DC,               16296000 T  0157!004!0
                          THE VARIABLE "INDEX" IS COMPUTED,DEPENDING UPON WHICH       16297000 T  0157!004!0
                          CASE EXISTS,SUCH THAT ARRAY ELEMENT "MACRO[INDEX]"CONTAINS  16298000 T  0157!004!0
                          THE CODE TO BE EMITTED,                                       16299000 T  0157!004!0
                          EACH ELEMENT OF MACRO HAS  1-3 SYLLABLES ORDERED FROM       16300000 T  0157!004!0
                          RIGHT TO LEFT, UNUSED SYLLABLES MUST = 0.    EACH MACRO     16301000 T  0157!004!0
                          MAY REQUIRE AT MOST ONE REPEAT PART,                          16302000 T  0157!004!0
                          IN THIS PROCEDURE,INDEXS,THE VARIABLE "ADDR" CONTAINS THE   16303000 T  0157!004!0
                          PROPER REPEAT PART BY THE TIME THE LABEL "GENERATE" IS      16304000 T  0157!004!0
                          ENCOUNTERED. THE SYLLABLES ARE FETCHED FROM MACRO[TYPE]     16305000 T  0157!004!0
                          ONE AT A TIME AND IF THE REPEAT PART ≠ 0 THEN"ADDR" IS      16306000 T  0157!004!0
                          USED AS THE REPEAT PART,THUS BUILDING A SYLLABLE WITH       16307000 T  0157!004!0
                          THE PROPER  ADDRESS AND OPERATOR .                          16308000 T  0157!004!0
                          NOTE! IF MACRO[TYPE] = 0 THEN THIS SIGNIFIES A SYNTAX       16309000 T  0157!004!0
                                ERROR.                                                ;  16310000 T  0157!004!0
                    PROCEDURE INDEXS;                                                   16311000 T  0157!004!0
PRT(1145) = INDEXS
                    BEGIN                                                               16312000 T  0157!004!0
                    LABEL EXIT,GENERATE,L,L1;                                           16313000 T  0157!004!0
                                                                          START OF SEGMENT ********** 164
                    INTEGER TCLASS,INDEX,ADDR,J;                                        16314000 T  0164!000!0
STACK(F+2) = TCLASS
STACK(F+3) = INDEX
STACK(F+4) = ADDR
STACK(F+5) = J

                    TCLASS ← ELCLASS ;                                                  16315000 T  0164!000!0
                    IF STEPI ≠ ASSIGNOP THEN BEGIN ERR(251); GO TO EXIT END;            16316000 T  0164!000!1
                    IF TCLASS = LOCLID THEN                                             16317000 T  0164!000!3
                         BEGIN                                                          16318000 T  0164!000!4
                         XMARK(ASSIGNREF);                                  %116-       16318500 C  0164!000!1
                         IF SIV>STEPI OR ELCLASS>TALLYV THEN GO TO L;                   16319000 T  0164!000!2
                         INDEX ← 32 + ELCLASS-SIV;                                      16320000 T  0164!001!3
                         ADDR ← ELBAT[I-2].ADDRESS;                                     16321000 T  0164!001!3
                         GO TO GENERATE;                                                16322000 T  0164!001!3
                         END;                                                           16323000 T  0164!001!0
                    IF TCLASS = STEPI THEN                                              16324000 T  0164!001!0
                         BEGIN                                                          16325000 T  0164!001!2
                         IF STEPI≠ADDOP THEN BEGIN ERR(252); GO EXIT END ELSE           16326000 T  0164!001!3
                         IF STEPI≠LITNO AND ELCLASS≠LOCLID THEN                         16326100 T  0164!002!0
                              BEGIN ERR(253); GO EXIT END;                              16327000 T  0164!002!1
                         INDEX ← TCLASS-SIV                                             16328000 T  0164!002!1
                              +REAL(ELBAT[I-1].ADDRESS=SUB) × 4                         16329000 T  0164!002!2
                              + REAL(ELCLASS =LOCLID) × 8;                              16330000 T  0164!002!2
                         END                                                            16331000 T  0164!003!0
                    ELSE                                                                16332000 T  0164!003!0
```

```
          BEGIN                                                                  16333000 T  01641003010
          INDEX ← TCLASS -SIV                                                    16334000 T  01641003011
              + ( IF ELCLASS = LOCLID THEN 16 ELSE                               16335000 T  01641003112
               IF ELCLASS = LOCV THEN 20 ELSE                                    16336000 T  01641003312
               IF ELCLASS = SCV THEN 24 ELSE                                     16337000 T  01641003512
               IF ELCLASS= DCV  THEN 28 ELSE 25);                                16338000 T  01641003712
          IF ELCLASS = LOCV THEN                                                 16339000 T  01641004010
               IF STEPI ≠ LOCLID THEN GO TO L;                                   16340000 T  01641004112
          IF ELCLASS = LITNO AND TCLASS = TALLYV THEN                            16341000 T  01641004312
               BEGIN EMITC(ELBAT[I],ADDRESS,SEC);GO TO EXIT END;                 16342000 T  01641004411
          END ;                                                                  16343000 T  01641004713
     ADDR ← ELBAT[I],ADDRESS;                                                    16344000 T  01641004713
GENERATE;                                                                        16345000 T  01641004912
     IF MACRO[INDEX]= 0 THEN                                                     16346000 T  01641004912
  L;      BEGIN ERR(250);GO TO EXIT END;                                         16347000 T  01641005010
     J ← 8; TCLASS ←0 ;                                                          16348000 T  01641005210
 L1; MOVECHARACTERS(2,MACRO[INDEX],J←J-2,TCLASS,6 );                             16349000 T  01641005313
     IF TCLASS≠0 THEN                                                            16350000 T  01641005712
          BEGIN                                                                  16351000 T  01641005713
          EMITC(IF TCLASS≥64 THEN ADDR ELSE 0,TCLASS);                           16352000 T  01641005810
          GO TO L1                                                               16353000 T  01641006112
          END;                                                                   16354000 T  01641006113
EXIT;END  INDEXS ;                                                               16355000 T  01641006113
                                                                      164 IS   65 LONG, NEXT SEG  157


COMMENT  DSS COMPILES DESTINATION STREAM STATEMENTS,                             16356000 T  01571004210
         DS← LIT"STRING" IS HANDLED AS A SPECIAL CASE BECAUE THE                 16357000 T  01571004210
             STRING MUST BE SCANED FROM RIGHT TO LEFT,REPEATEDLY IF              16358000 T  01571004210
             NECESSARY, AND EMITTED TO THE PROGRAM STREAM,  IN                   16359000 T  01571004210
             ALL OTHER CASES,THE ELBAT WORD CONTAINS THE OPERATOR IN             16360000 T  01571004210
             THE OPCODE FIELD  ;                                                 16361000 T  01571004210
        PROCEDURE DSS;                                                           16362000 T  01571004210
PRT(1146) = DSS
                BEGIN                                                            16363000 T  01571004210
                INTEGER ADDR,J,K,L,T;                                            16364000 T  01571004210
                                                                      START OF SEGMENT ********** 165
STACK(F+2) = ADDR
STACK(F+3) = J
STACK(F+4) = K
STACK(F+5) = L
STACK(F+6) = T
                LABEL EXIT,L1;                                                   16365000 T  01651000010
                DEFINE OPCODE=[27:6]#;                                           16366000 T  01651000010
                IF STEPI ≠ ASSIGNOP THEN BEGIN ERR(251); GO TO EXIT END;         16367000 T  01651000010
                IF STEPI = LOCLID THEN                                           16368000 T  01651000211
                     BEGIN                                                       16369000 T  01651000313
                     EMITC(ELBAT[I],ADDRESS,CRF);                                16370000 T  01651000410
                     ADDR← 0;                                                    16371000 T  01651000610
                     IF STEPI = LITV THEN GO TO L1                               16372000 T  01651000611
                     END                                                         16373000 T  01651000713
                ELSE IF ELCLASS= LITNO THEN                                      16374000 T  01651000810
                     BEGIN                                                       16375000 T  01651000913
                     ADDR ← ELBAT[I],ADDRESS; STEPIT ;                           16376000 T  01651001010
                     END                                                         16377000 T  01651001210
                ELSE ADDR ← 1 ;                                                  16378000 T  01651001210
```

```
                    IF ELCLASS =  TRNSFER OR ELCLASS = FILLV THEN                    %A          16379000 T  0165:001:2
                       EMITC(ADDR,ELBAT[I],OPCODE)                                   %A          16379500 T  0165:001:5:2
                    ELSE                                                                       16380000 T  0165:001:7:2
                    IF ELCLASS = LITV THEN                                                     16381000 T  0165:001:7:2
                         BEGIN                                                                 16382000 T  0165:001:8:1
                         EMITC(ADDR,TRP);                                                      16383000 T  0165:001:9:2
                         IF STEPI ≠ STRING AND ELCLASS ≠ STRNGCON AND             %111-       16384000 P  0165:002:0:0
                           ELCLASS ≠ LITNO AND ELCLASS ≠ NONLITNO THEN            %111-       16384100 C  0165:002:2:0
                              BEGIN ERR(255);GO TO EXIT END;                                   16384500 T  0165:002:4:0
                           IF ELCLASS = LITNO OR ELCLASS = NONLITNO THEN          %111-       16384700 C  0165:002:5:3
                             MOVECHARACTERS(COUNT:=IF ADDR < 8 THEN ADDR ELSE 8,             16384800 C  0165:002:7:3
                              C,8-COUNT,ACCUM[1],3);                               %111-       16384900 C  0165:003:1:2
                         IF ADDR MOD 2 ≠ 0 THEN                                              16385000 T  0165:003:3:2
                              BEGIN                                                          16386000 T  0165:003:4:0
                              EMIT(ACCUM[1],[18:6]);  J ← 1;                                 16387000 T  0165:003:4:1
                              END ;                                                          16388000 T  0165:003:7:2
                         FOR K ←J+2 STEP 2 UNTIL ADDR DO                                     16389000 T  0165:003:7:2
                              BEGIN                                                          16390000 T  0165:004:1:2
                              FOR L ←6,7 DO                                                  16391000 T  0165:004:1:2
                              MOVECHARACTERS(1,ACCUM[1],2+(IF J←J+1>COUNT THEN J←1           16392000 T  0165:004:5:3
STACK(F+7) = *TEMPORARY STORAGE*
                                        ELSE J),T,L  );                                      16393000 T  0165:004:9:2
                              EMIT(T);                                                       16394000 T  0165:005:2:0
                         END  END                                                            16395000 T  0165:005:3:2
                    ELSE                                                                     16396000 T  0165:005:3:3
                 L1:    ERR(250);                                                           16397000 T  0165:005:3:3
               EXIT:END  DSS  ;                                                              16398000 T  0165:005:4:1
                                                                                  165 IS   59 LONG, NEXT SEG  157

                         COMMENT SKIPS  COMPILES THE SKIP BIT STATEMENT,                     16399000 T  0157:004:2:0
                                 IF THE REPEAT INDEX IS A LOCALID THEN A CRF IS EMITTED,     16400000 T  0157:004:2:0
                                 A  BSS OR BSD IS THEN EMITTED FOR SKIP SOURCE BITS (SB)     16401000 T  0157:004:2:0
                                 OR   SKIP DESTINATION BITS (DB) RESPECTIVELY ;              16402000 T  0157:004:2:0
                    PROCEDURE SKIPS ;                                                        16403000 T  0157:004:2:0
PRT(1147) = SKIPS
                              BEGIN                                                          16404000 T  0157:004:2:0
                              REAL ADDR;                                                     16405000 T  0157:004:2:0
                                                                                  START OF SEGMENT **********  166
STACK(F+2) = ADDR
                              IF STEPI = LOCLID THEN                                         16406000 T  0166:000:0:0
                                   BEGIN                                                     16407000 T  0166:000:1:2
                                   EMITC(ELBAT[I],ADDRESS,CRF); ADDR←0; STEPIT;              16408000 T  0166:000:1:3
                                   END                                                       16409000 T  0166:000:4:1
                              ELSE IF ELCLASS = LITNO THEN                                   16410000 T  0166:000:4:1
                                   BEGIN                                                     16411000 T  0166:000:5:3
                                   ADDR← ELBAT[I],ADDRESS; STEPIT                            16412000 T  0166:000:6:0
                                   END                                                       16413000 T  0166:000:7:3
                              ELSE ADDR ← 1 ;                                                16414000 T  0166:000:8:0
                              IF ELCLASS =SBV THEN EMITC(ADDR,BSS)                           16415000 T  0166:000:9:3
                              ELSE                                                           16416000 T  0166:001:1:2
                              IF ELCLASS =DBV THEN  EMITC(ADDR,BSD)                          16417000 T  0166:001:1:3
                              ELSE ERR(250);                                                 16418000 T  0166:001:4:0
                              END  SKIPS  ;                                                  16419000 T  0166:001:5:3
                                                                                  166 IS   18 LONG, NEXT SEG  157
```

```
                    COMMENT JUMPS COMPILES JUMP OUT AND JUMP OUT TO STATEMENTS.                    16420000 T  0157:004210
                            JUMP OUT TO STATEMENTS CAUSE JUMP LEVEL TO BE SET TO                  16421000 T  0157:004210
                            THE NUMBER OF LEVELS SPECIFIED. THEN THIS NUMBER OF                   16422000 T  0157:004210
                            JNS ARE EMITTED AND GOTOS IS CALLED TO COMPILE THE                    16423000 T  0157:004210
                            JUMP INSTRUCTION.                                                     16424000 T  0157:004210
                            SIMPLE JUMP OUTS ARE HANDLED BY EMITTING ONE JNS,ENTERING             16425000 T  0157:004210
                            A PSEUDO STLABID IN INFO AND SETTING ELBAT[I] SUCH THAT               16426000 T  0157:004210
                            THE GOTOS PROCEDURE WILL PERFORM THE ACTION OF SETTING                16427000 T  0157:004210
                            UP THE LINKS FOR LATER FIX UPS. THE NEST STATEMENT CAUSES             16428000 T  0157:004210
                            THESE FIX UPS(IF EMITTING OF JUMP INSTRUCTIONS) BY CALLING            16429000 T  0157:004210
                            GO TOS WHEN THE RIGHT PAREN IS ENCOUNTERED.     ;                     16430000 T  0157:004210
                PROCEDURE JUMPS;                                                                  16431000 T  0157:004210
PRT(1150) = JUMPS
                    BEGIN                                                                         16432000 T  0157:004210
                    JUMPLEVEL←1;                                                                  16433000 T  0157:004210
                    IF STEPI≠DECLARATORS THEN FLAG(261);                                          16434000 T  0157:004211
                    IF STEPI = LITNO THEN JUMPLEVEL← ELBAT[I].ADDRESS                             16435000 T  0157:004512
                    ELSE BEGIN                                                                    16436000 T  0157:004611
                         IF ELCLASS≠ TOV AND ELCLASS≠ STLABID THEN                                16437000 T  0157:004811
                         BEGIN                                                                    16438000 T  0157:005010
                COMMENT  SIMPLE JUMP OUT STATEMENT;                                               16439000 T  0157:005011
                         IF JOINFO = 0 THEN                                                       16440000 T  0157:005011
                              BEGIN                                                               16441000 T  0157:005113
                              JOINFO ← NEXTINFO ;                                                 16442000 T  0157:005210
                         PUTNBUMP(0&(STLABID×2+1)                                                 16443000 T  0157:005211
                                 [2:40:8]&2[27:40:8 ]);                                           16444000 T  0157:005410
                              PUTNBUMP(0&(JOINFO-LASTINFO    )[ 4:40:8]);                        16445000 T  0157:005611
                              PUTNBUMP (0);                                                       16446000 T  0157:005811
                              LASTINFO ← JOINFO;                                                  16447000 T  0157:005913
                              END;                                                                16448000 T  0157:006010
                         ELBAT[I←  I-1]←  TAKE(JOINFO)&JOINFO[35:35:13];                          16449000 T  0157:006010
                         END; I←I-1 ;                                                             16450000 T  0157:006410
                END;                                                                              16451000 T  0157:006512
                FOR GT1← 1 STEP 1 UNTIL JUMPLEVEL DO                                              16452000 T  0157:006512
                     EMIT(  JNS);                                                                 16453000 T  0157:006610
                GOTOS;                                                                            16454000 T  0157:006912
                END JUMPS;                                                                        16455000 T  0157:006913


COMMENT STREAMSTMT  ENVOKES THE APPROPRIATE PROCEDURE TO HANDLE                                   16456000 T  0157:007010
        THE VARIOUS AND SUNDRY STREAM PROCEDURE STATEMENTS,                                       16457000 T  0157:007010
        THE STATEMENTS ARE BROKEN DOWN AS FOLLOWS:                                                16458000 T  0157:007010
             IDENTIFIED BY              PROCEDURE ENVOKED                                         16459000 T  0157:007010
               END                        GO TO FINI                                             16460000 T  0157:007010
               SEMICOLON                  GO TO FINI                                             16461000 T  0157:007010
               )                          GO TO FINI                                             16462000 T  0157:007010
               IF                         IFS                                                    16463000 T  0157:007010
               GO                         GOTOS                                                  16464000 T  0157:007010
               RELEASE                    RELEASES                                               16465000 T  0157:007010
             BEGIN                        COMPOUNDTAIL                                           16466000 T  0157:007010
             SI,DI,CI,TALLY,LOCALID INDEXS                                                       16467000 T  0157:007010
             DS                         DSS                                                    16468000 T  0157:007010
```

```
                SKIP                        SKIPS                                    16469000 T  0157:007010
                  JUMP                      JUMPS                                    16470000 T  0157:007010
                  LABELID                   LABELS                                   16471000 T  0157:007010
                  LITERAL NO.,LOCALID(      NESTS                                    16472000 T  0157:007010
          UPON EXITING,STREAMSTMT ASSURES THAT "I" POINTS TO                         16473000 T  0157:007010
          THE SEMICOLON ,END OR ) IN SYNTACICALLY CORRECT PROGRAMS;                  16474000 T  0157:007010
      LABEL  L,L1,L2,L3,L4,L5,L6,L7,L8,L9,L10,EXIT,FINI,START;                       16475000 T  0157:007010
      SWITCH TYPE := FINI,L,FINI,L3,L4,L5,L6,L7,L7,L7,L7,L8,L9,L10;                  16476000 T  0157:007010
START:    GO TO TYPE[ ELCLASS-ENDV+1];                                             16477000 T  0157:008010
          IF ELCLASS= RTPAREN THEN GO TO FINI ;                                    16478000 T  0157:008410
          IF ELCLASS = LITNO OR ELCLASS=LOCLID AND TABLE(I+1)                      16479000 T  0157:008512
                    = LFTPAREN THEN GO TO L1;                                      16480000 T  0157:008713
          IF ELCLASS= STLABID THEN GO TO L2 ;                                      16481000 T  0157:008913
          IF ELCLASS= LOCLID THEN GO TO L7 ;                                       16482000 T  0157:009112
    L:    ERR( 250 ); GO TO FINI ;                                                 16483000 T  0157:009210
    L1:   NESTS;    GO TO EXIT;                                                    16484000 T  0157:009410
    L2:   LABELS;   GO TO START;                                                   16485000 T  0157:009610
    L3:   IFS;      GO TO FINI;                                                    16486000 T  0157:009810
    L4:   GOTOS;    GO TO EXIT;                                                    16487000 T  0157:010010
    L5:   RELEASES; GO TO EXIT;                                                    16488000 T  0157:010210
    L6:   I:=I+1 ; COMPOUNDTAIL; GO TO FINI;                                       16489000 T  0157:010410
    L7:   INDEXS;   GO TO EXIT;                                                    16490000 T  0157:010712
    L8:   DSS;      GO TO EXIT;                                                    16491000 T  0157:010912
    L9:   SKIPS;    GO TO EXIT;                                                    16492000 T  0157:011112
    L10:  JUMPS;    GO TO EXIT;                                                    16493000 T  0157:011312
EXIT:     STEPIT;                                                                  16494000 T  0157:011512
FINI:     END   STREAMSTMT;                                                        16495000 T  0157:011611
                                                                  157 IS  118 LONG, NEXT SEG    3
```

```
                                                                                   16496000 T  0003:098112
                TIME1 := TIME(1); PROGRAM;                                         17000000 T  0003:098112
                  ENDOFJTALL;                                                      17000100 T  0003:098410
                IF (XREF OR DEFINING.[1:1]) AND XLUN > 0 THEN          %116-       17001000 P  0003:098410
                  BEGIN DEFINE LSS= <#,GTR=>#,NEQ= #,LEQ=≤#;            %DFB       17002000 T  0003:098611
PRT(1151) = *SEGMENT DESCRIPTOR*
                                                                                START OF SEGMENT **********  167
                  DEFINE XREFINFO[INDEX] = INFO[((INDEX).CF DIV 2).[33:7],  %116-  17002005 P  0167:000010
                                           ((INDEX).CF DIV 2).LINKC]#,      %116-  17002006 C  0167:000010
                       CF = [33:15]#,                                       %116-  17002007 C  0167:000010
                       FF = [18:15]#,                                       %116-  17002008 C  0167:000010
                       NEWID[INDEX] = (IF BOOLEAN(INDEX) THEN XREFINFO[INDEX].FF  17002009 C  0167:000010
                                      ELSE XREFINFO[INDEX].CF)#;            %116-  17002010 C  0167:000010
                  ARRAY TIMINGS[0:2,0:3];                                  %116-  17002012 C  0167:000010
PRT(1152) = TIMINGS
                  PROCEDURE SAVETIMES(I);                                  %116-  17002015 C  0167:000312
PRT(1153) = SAVETIMES
                    VALUE I; INTEGER I;                                    %116-  17002020 C  0167:000312
                  BEGIN                                                    %116-  17002025 C  0167:000312
                    INTEGER J;                                             %116-  17002030 C  0167:000312
                                                                                START OF SEGMENT **********  168
STACK(F+2) := J
                    FOR J := 1 STEP 1 UNTIL 3 DO                           %116-  17002035 C  0168:000010
                      TIMINGS[I,J] := TIME(J);                             %116-  17002040 C  0168:000112
                  END;                                                     %116-  17002045 C  0168:000610
```

```
                                                          168 IS    9 LONG, NEXT SEG  167

                PROCEDURE UPDATETIMES(I);                     %116-     17002050 C  0167:0003:2
PRT(1154) = UPDATETIMES
                  VALUE I; INTEGER I;                         %116-     17002055 C  0167:0003:2
                BEGIN                                         %116-     17002060 C  0167:0003:2
                  INTEGER J;                                  %116-     17002065 C  0167:0003:2
                                                          START OF SEGMENT **********  169
STACK(F+2) = J
                  FOR J := 1 STEP 1 UNTIL 3 DO                %116-     17002070 C  0169:0000:0
                    TIMINGS[I,J] := TIME(J) - TIMINGS[I,J];   %116-     17002075 C  0169:0001:2
                END;                                          %116-     17002080 C  0169:0007:3
                                                          169 IS   10 LONG, NEXT SEG  167

                WRITE(LINE[PAGE]);                                      17002520 T  0167:0003:2
                SAVETIMES(0); % SAVE TIMES FOR START OF IDENTIFIER SORT,  %116-  17002525 C  0167:0007:2
                LASTADDRESS+0;                                          17002530 T  0167:0008:0
               FOR XREFPT:=XREFPT STEP 1 UNTIL 29 DO XREFAY2[XREFPT]:=100000000;  17003000 T  0167:0009:2
               WRITE(DSK2,30,XREFAY2[*]);                     %DFB      17004000 T  0167:0013:3
                TOTALNO := XLUN; % REMEMBER NUMBER OF IDENTIFIERS,  %116-  17004500 P  0167:0017:3
                    XREFPT+XLUN+0;                            %DFB      17004600 T  0167:0018:1
                  FOR I:= 0 STEP 1 UNTIL 8191 DO              %116-     17004700 C  0167:0019:3
                      XREFINFO[I] := 0;                       %116-     17004710 C  0167:0022:0
                    BEGIN                                     %DFB      17005000 T  0167:0029:2
                    BOOLEAN PROCEDURE INPUT1(A);              %DFB      17006000 T  0167:0029:2
PRT(1155) = *SEGMENT DESCRIPTOR*
                                                          START OF SEGMENT **********  170
PRT(1156) = INPUT1
                      ARRAY A[0];                             %DFB      17007000 T  0170:0000:0
                      BEGIN                                   %DFB      17008000 T  0170:0000:0
                      LABEL L,EOF;                            %DFB      17009000 T  0170:0000:0
                                                          START OF SEGMENT **********  171
                      READ(DSK1,10,A[*])[EOF];                %DFB      17010000 T  0171:0000:0
PRT(1157) = EOF
                      GO TO L;                                %DFB      17011000 T  0171:0005:2
                      EOF: INPUT1:=TRUE;                      %DFB      17012000 T  0171:0005:3
                          REWIND(DSK1);                       %DFB      17013000 T  0171:0006:1
                      L:                                      %DFB      17014000 T  0171:0008:1
                      END;                                    %DFB      17015000 T  0171:0009:2
                                                          171 IS   14 LONG, NEXT SEG  170

                  PROCEDURE OUTPUT1(B,A);                     %DFB      17016000 T  0170:0000:0
PRT(1160) = OUTPUT1
                    VALUE B;                                  %DFB      17017000 T  0170:0000:0
                    BOOLEAN B;                                %DFB      17018000 T  0170:0000:0
                    ARRAY A[0];                               %DFB      17019000 T  0170:0000:0
                      BEGIN                                   %DFB      17020000 T  0170:0000:0
                      IF B THEN                               %DFB      17021000 T  0170:0000:0
                        BEGIN                                 %116-     17022000 P  0170:0000:0
```

```
      REWIND(DSK1);                                                    %116-    17022100 C  0170:0000:1
      UPDATETIMES(0); % UPDATE TIMES FOR IDENTIFIER SORT.              %116-    17022200 C  0170:0002:1
      TIMINGS[0,0] := XLUN; % NUMBER OF IDENTIFIERS SORTED.            %116-    17022300 C  0170:0003:3
    END                                                                %116-    17022400 C  0170:0005:3
  ELSE                                                                 %DFB     17023000 T  0170:0005:3
    BEGIN                                                              %DFB     17024000 T  0170:0005:3
    IF BOOLEAN(A[8]) THEN                                              %116-    17025000 P  0170:0006:0
      XREFINFO[A[8]].FF := XLUN := XLUN + 1                            %116-    17025100 C  0170:0006:1
    ELSE                                                               %116-    17025200 C  0170:0013:2
      XREFINFO[A[8]].CF := XLUN := XLUN + 1;                           %116-    17025300 C  0170:0014:1
    A[8].IDNOF := XLUN;                                                %116-    17025400 C  0170:0022:0
    WRITE(DSK1,10,A[*]);                                               %DFB     17026000 T  0170:0024:0
    END;                                                               %DFB     17027000 T  0170:0028:1
  END;                                                                 %DFB     17028000 T  0170:0028:1


PRT(1161) = COMPS1
  BOOLEAN STREAM PROCEDURE COMPS1(A,B);                                %DFB     17029000 T  0170:0028:1
    BEGIN                                                              %DFB     17030000 T  0170:0028:1
    SI:=A;                                                             %DFB     17031000 T  0170:0029:2
    DI:=B;                                                             %DFB     17032000 T  0170:0029:2
    IF 63 SC < DC THEN                                                 %116-    17033000 P  0170:0029:3
      TALLY := 1                                                       %116-    17033100 C  0170:0030:0
    ELSE                                                               %116-    17033200 C  0170:0030:0
      BEGIN                                                            %116-    17033300 C  0170:0030:1
        SI := A;                                                       %116-    17033400 C  0170:0030:1
        DI := B;                                                       %116-    17033500 C  0170:0030:1
        IF 63 SC = DC THEN                                             %116-    17033600 C  0170:0031:2
          TALLY := 2;                                                  %116-    17033700 C  0170:0031:3
      END;                                                             %116-    17033800 C  0170:0031:3
    COMPS1:=TALLY;                                                     %DFB     17034000 T  0170:0031:3
    END;                                                               %DFB     17035000 T  0170:0032:0


PRT(1162) = HVS1
  STREAM PROCEDURE HVS1(A);                                            %DFB     17036000 T  0170:0033:2
    BEGIN                                                              %DFB     17037000 T  0170:0033:2
    DI:=A;                                                             %DFB     17038000 T  0170:0033:2
    DS:=8 LIT "9";                                                     %DFB     17039000 T  0170:0033:2
    SI:=A;                                                             %DFB     17040000 T  0170:0034:1
    DS:= 7 WDS;                                                        %DFB     17041000 T  0170:0034:1
    DS := 8 LIT 3"777777777"; % ID.NO. AND SEG.NO. FIELDS              %116-    17041100 C  0170:0035:2
    END;                                                               %DFB     17042000 T  0170:0036:0


PRT(1163) = COMP1
  BOOLEAN PROCEDURE COMP1(A,B);                                        %DFB     17042100 T  0170:0036:1
        ARRAY A,B[0];                                                  %DFB     17042200 T  0170:0036:1
        IF REAL(COMP1:=COMPS1(A,B)) = 2 THEN % IDS EQUAL               %116-    17042300 P  0170:0036:1
          COMP1 := A[8].IDNOF < B[8].IDNOF;                            %116-    17042350 C  0170:0040:0
```

```
                        PROCEDURE HV1(A);                                                                %DFB        17042400 T  0170:0045:2
PRT(1164) = HV1
                                ARRAY A[0];                                                              %DFB        17042500 T  0170:0045:2
                                HVS1(A);                                                                 %DFB        17042600 T  0170:0045:2



                        XLUN:=0;                                                                         %DFB        17043000 T  0170:0047:3
                        REWIND(DSK1);                                                                    %DFB        17044000 T  0170:0048:1
                        SORT(OUTPUT1,INPUT1,0,HV1,COMP1,10,IF TOTALNO < 1000 THEN                        %116-       17045000 P  0170:0050:1
PRT(1165) = SORT TEMPORARY
PRT(1166) = MERGE
                                7000 ELSE 10000);                                                        %116-       17045100 C  0170:0073:2
PRT(1167) = SORT
                    END;                                                                                 %DFB        17046000 T  0170:0075:2
PRT(1170) = *SEGMENT DESCRIPTOR*

                                                                                                                 170 IS   82 LONG, NEXT SEG  167
                    BEGIN                                                                                %DFB        17047000 T  0167:0031:2
                     ARRAY IDTYPE[0:(IDMAX+4)*4-1];                                                      %117-       17047100 C  0167:0031:2
PRT(1171) = *SEGMENT DESCRIPTOR*

                                                                                                                START OF SEGMENT **********  172
PRT(1172) = IDTYPE
                    STREAM PROCEDURE SETUPHEADING(S,D,SEG,SEQNO,FWDTOG,LBLTOG,                           %116-       17047200 C  0172:0005:3
PRT(1173) = SETUPHEADING
                                FWDSEQNO,TYPE,OWNTOG,PARAMTOG,                                           %116-       17047300 C  0172:0005:3
                                VALTOG);                                                                 %116-       17047350 C  0172:0005:3
  VALUE SEG,SEQNO,FWDTOG,LBLTOG,FWDSEQNO,OWNTOG,PARAMTOG,                                                %116-       17047400 C  0172:0005:3
    VALTOG;                                                                                              %116-       17047450 C  0172:0005:3
BEGIN                                                                                                    %116-       17047500 C  0172:0005:3
  SI := S;                                                                                               %116-       17047700 C  0172:0006:0
  DI := D;                                                                                               %116-       17047800 C  0172:0006:0
  63 (IF SC = " " THEN JUMP OUT ELSE DS := CHR);                                                         %116-       17047900 C  0172:0006:1
  DS := 6 LIT "  --  ";                                                                                  %116-       17048000 P  0172:0009:2
  OWNTOG (DS := 4 LIT "OWN ");                                                                           %116-       17048100 C  0172:0010:0
  SI := TYPE;                                                                                            %116-       17049300 C  0172:0012:0
  32 (IF SC = "," THEN JUMP OUT ELSE DS := CHR);                                                         %116-       17049400 C  0172:0012:0
  PARAMTOG (DS := 6 LIT "  --  ";                                                                        %116-       17049410 C  0172:0015:2
            DS := 4 LIT "NAME";                                                                          %116-       17049420 C  0172:0017:2
            VALTOG (DI := DI - 4; DS := 5 LIT "VALUE");                                                  %116-       17049430 C  0172:0017:3
            DS := 10 LIT " PARAMETER");                                                                  %116-       17049440 C  0172:0020:0
  DS := 26 LIT "  --  DECLARED IN SEGMENT ";                                                             %116-       17049500 C  0172:0022:0
  SI := LOC SEG;                                                                                         %116-       17049600 C  0172:0025:3
  S := DI;                                                                                               %116-       17049700 C  0172:0025:3
  DS := 4 DEC; DI := DI - 4; DS := 3 FILL; % CONV AND ZERO SUPPR                                                     17049800 C  0172:0026:0
  DI := DI + 8; % TO FORCE STORE OF LAST WORD                                                            %116-       17049900 C  0172:0026:1
  SI := S;                                                                                               %116-       17050000 P  0172:0027:2
  DI := S;                                                                                               %116-       17050100 C  0172:0027:2
  4(IF SC # " " THEN DS:= CHR ELSE SI := SI + 1);                                                        %116-       17050200 C  0172:0027:3
  DS := 4 LIT " AT ";                                                                                    %116-       17050300 C  0172:0029:2
  SI := LOC SEQNO;                                                                                       %116-       17050400 C  0172:0030:0
  DS := 8 DEC;                                                                                           %116-       17050500 C  0172:0030:0
  FWDTOG (DS := 17 LIT "  --  FORWARD AT ";                                                             %116-       17050600 C  0172:0030:1
          SI := LOC FWDSEQNO;                                                                            %116-       17050700 C  0172:0034:0
          DS := 8 DEC);                                                                                  %116-       17050800 C  0172:0034:0
  LBLTOG (DS := 16 LIT "  --  OCCURS AT ";                                                               %116-       17050900 C  0172:0034:1
```

```
                    SI := LOC FWDSEQNO;                                    %116-      17051000 P  0172:003B:0
                    DS := 8 DEC);                                          %116-      17051100 C  0172:003B:0
                ENd OF SETUPHEADING;                                       %116-      17051200 C  0172:003B:1


                STREAM PROCEDURE ADDASEQNO(SEQNO,N,STARS,D);               %116-      17051300 C  0172:003B:2
PRT(1174) = ADDASEQNO                                                      %116-      17051400 C  0172:003B:2
                  VALUE SEQNO,N,STARS;
                BEGIN                                                      %116-      17051500 C  0172:003B:2
                  DI := D;                                                 %116-      17051600 C  0172:003B:2
                  DI := DI + 8;                                            %116-      17051700 C  0172:003B:2
                  N (DI := DI + 10);                                       %116-      17051800 C  0172:003B:2
                  STARS(DI := DI - 1; DS := LIT "*");                      %116-      17051900 C  0172:003B:3
                  SI := LOC SEQNO;                                         %116-      17052000 P  0172:004I:2
                  DS := 8 DEC;                                             %116-      17052100 C  0172:004I:2
                  DS := LIT " ";                                           %116-      17052200 C  0172:004I:2
                  STARS (DI := DI - 1; DS := LIT "*");                     %116-      17052300 C  0172:004I:3
                ENd;                                                       %116-      17052400 C  0172:004I:0
                                                                           %116-      17052500 C  0172:004K:0


                STREAM PROCEDURE BLANKET(D);                               %116-      17052600 C  0172:004K:0
PRT(1175) = BLANKET
                BEGIN                                                      %116-      17052700 C  0172:004K:0
                  DI := D;                                                 %116-      17052800 C  0172:004L:2
                  DS := 8 LIT " ";                                         %116-      17052900 C  0172:004L:2
                  SI := D;                                                 %116-      17053000 C  0172:004M:1
                  DS := 16 WDS;                                            %116-      17053100 C  0172:004M:1
                ENd OF BLANKET;                                            %116-      17053200 C  0172:004N:2


                PROCEDURE PRINTXREFSTATISTICS;                             %116-      17053300 C  0172:004N:2
PRT(1176) = PRINTXREFSTATISTICS
                BEGIN                                                      %116-      17053400 C  0172:004N:2
                  SWITCH FORMAT STATS :=                                   %116-      17053500 C  0172:004N:2
                                                                                START OF SEGMENT **********  173
                                                                                START OF SEGMENT **********  174
PRT(1177) = STATS
                    (///, "CROSS REFERENCE STATISTICS", /,                 %116-      17053600 C  0174:0000:0
                          "----- --------- ----------", /),               %116-      17053700 C  0174:0000:0
                    ("PHASE ONE - SORT",I6," IDENTIFIERS"),                %116-      17053800 C  0174:0000:0
                    ("PHASE TWO - SORT",I7," REFERENCES"),                 %116-      17053900 C  0174:0000:0
                    ("PHASE THREE - PRINT CROSS REFERENCE (",I7," LINES)"), %116-      17054000 C  0174:0000:0
                    (X5,I4,":",2I1," ELAPSED TIME (MIN:SEC)"),             %116-      17054100 C  0174:0000:0
                    (X5,I4,":",2I1," PROCESSOR TIME"),                     %116-      17054200 C  0174:0000:0
                    (X5,I4,":",2I1," I/O TIME",/);                         %116-      17054300 C  0174:0000:0
                                                                                174 IS    87 LONG, NEXT SEG  173
                  INTEGER I,J,K;                                           %116-      17054400 C  0173:0000:0
STACK(F+2) = I
STACK(F+3) = J
STACK(F+4) = K
```

```
                    WRITE(LINE,STATS[0]);                                          %116-        17054500 C  0173:000010
                    FOR I := 0 STEP 1 UNTIL 2 DO                                   %116-        17054600 C  0173:000411
                      BEGIN                                                        %116-        17054700 C  0173:000610
                        WRITE(LINE,STATS[I+1],TIMINGS[I,0]);                       %116-        17054800 C  0173:000610
PRT(1200) = *LIST, LABEL, OR SEGMENT DESCRIPTOR*
                        FOR J := 1 STEP 1 UNTIL 3 DO                               %116-        17054900 C  0173:001713
                          BEGIN                                                    %116-        17055000 P  0173:001912
                            K := (TIMINGS[I,J] + 30) DIV 60; % ROUND TO NEAREST SECON          17055010 C  0173:001912
                            WRITE(LINE,STATS[J+3],K DIV 60,(K:=K MOD 60) DIV 10,%116-          17055020 C  0173:002210
PRT(1201) = *LIST, LABEL, OR SEGMENT DESCRIPTOR*
                                                                K MOD 10);         %116-        17055025 C  0173:003313
                          END;                                                     %116-        17055030 C  0173:003913
                      END;                                                         %116-        17055100 P  0173:004113
                    END PRINTXREFSTATISTICS;                                       %116-        17055200 P  0173:004410
                                                                                          173 IS   47 LONG, NEXT SEG  172


                    DEFINE REFCOUNT = TIMINGS[1,0]#; % NUMBER OF REFERENCES SORTED.            17069300 C  0172:004912
                    BOOLEAN FIRSTTIME; % TRUE ON FIRST CALL OF OUTPUT PROCEDURE.               17069400 C  0172:004912
PRT(1202) = FIRSTTIME
                    ARRAY PAY[0:17];                                               %DFB        17069500 T  0172:004912
PRT(1203) = PAY
                    REAL LASTADDRESS;                                              %116-       17069600 C  0172:005312
PRT(1204) = LASTADDRESS
                    BOOLEAN PROCEDURE INPUT2(A);                                   %DFB        17070000 T  0172:005312
PRT(1205) = INPUT2
                    ARRAY A[0];                                                    %DFB        17071000 T  0172:005312
                      BEGIN                                                        %DFB        17072000 T  0172:005312
                      LABEL L,EOF;                                                 %DFB        17073000 T  0172:005312
                                                                                        START OF SEGMENT **********  175
                      DEFINE I = LASTADDRESS#;                                     %116-       17073100 P  0175:000010
                      IF XREFPT:=XREFPT+1=30 THEN                                  %DFB        17074000 T  0175:000010
                        BEGIN                                                      %DFB        17075000 T  0175:000113
                        READ(DSK2,30,XREFAY2[*])[EOF];                             %DFB        17076000 T  0175:000210
PRT(1206) = EOF
                        XREFPT:=0;                                                 %DFB        17077000 T  0175:000713
                        END;                                                       %DFB        17078000 T  0175:000810
                    IF ( I  :=XREFAY2[XREFPT]).[21:27] GTR 99999999 THEN GO TO EOF;            17079000 T  0175:000810
                    A[0] := I & NEWID[I.REFIDNOF] REFIDNOF;                        %116-       17080000 P  0175:001011
                    REFCOUNT := REFCOUNT + 1;                                      %116-       17080100 C  0175:002810
                    GO TO L;                                                       %DFB        17081000 T  0175:003210
                    EOF: INPUT2:=TRUE;                                             %DFB        17082000 T  0175:003410
                         BLANKET(PAY);                                             %DFB        17083000 T  0175:003411
                    XREFAY1[8] := XREFPT := LASTADDRESS := 0;                      %116-       17084000 P  0175:003610
                    FILL IDTYPE[*] WITH                                            %116-       17084010 C  0175:003811
                      "UNKNOWN.                    ", % 0                          %116-       17084020 C  0175:003913
                                                                                        START OF SEGMENT **********  176
                      "STREAM LABEL.               ", % 1                          %116-       17084030 C  0175:004011
                      "STREAM VARIABLE.            ", % 2                          %116-       17084040 C  0175:004011
                      "DEFINE.                     ", % 3                          %116-       17084050 C  0175:004011
                      "LIST.                       ", % 4                          %116-       17084060 C  0175:004011
                      "FORMAT.                     ", % 5                          %116-       17084070 C  0175:004011
                      "SWITCH FORMAT.              ", % 6                          %116-       17084080 C  0175:004011
                      "FILE.                       ", % 7                          %116-       17084090 C  0175:004011
                      "SWITCH FILE.                ", % 8                          %116-       17084100 C  0175:004011
                      "SWITCH LABEL.               ", % 9                          %116-       17084110 C  0175:004011
```

```
                    "PROCEDURE,                    ", % 10                  %116-        17084120 C  0175:0040:1
                    "INTRINSIC,                    ", % 11                  %116-        17084130 C  0175:0040:1
                    "STREAM PROCEDURE,             ", % 12                  %116-        17084140 C  0175:0040:1
                    "BOOLEAN STREAM PROCEDURE,     ", % 13                  %116-        17084150 C  0175:0040:1
                    "REAL STREAM PROCEDURE,        ", % 14                  %116-        17084160 C  0175:0040:1
                    "ALPHA STREAM PROCEDURE,       ", % 15                  %116-        17084170 C  0175:0040:1
                    "INTEGER STREAM PROCEDURE,     ", % 16                  %116-        17084180 C  0175:0040:1
                    "BOOLEAN PROCEDURE,            ", % 17                  %116-        17084182 C  0175:0040:1
                    "REAL PROCEDURE,               ", % 18                  %116-        17084184 C  0175:0040:1
                    "ALPHA PROCEDURE,              ", % 19                  %116-        17084186 C  0175:0040:1
                    "INTEGER PROCEDURE,            ", % 20                  %116-        17084188 C  0175:0040:1
                    "BOOLEAN,                      ", % 21                  %116-        17084190 C  0175:0040:1
                    "REAL,                         ", % 22                  %116-        17084200 C  0175:0040:1
                    "ALPHA,                        ", % 23                  %116-        17084210 C  0175:0040:1
                    "INTEGER,                      ", % 24                  %116-        17084220 C  0175:0040:1
                    "BOOLEAN ARRAY,                ", % 25                  %116-        17084230 C  0175:0040:1
                    "REAL ARRAY,                   ", % 26                  %116-        17084240 C  0175:0040:1
                    "ALPHA ARRAY,                  ", % 27                  %116-        17084250 C  0175:0040:1
                    "INTEGER ARRAY,                ", % 28                  %116-        17084260 C  0175:0040:1
                    "LABEL,                        ", % 29                  %116-        17084270 C  0175:0040:1
                    "FIELD,                        ", % 30 (CLASS = 125)    %117-        17084275 C  0175:0040:1
                    "FAULT,                        ", % 32 (CLASS = 126)    %117-        17084280 C  0175:0040:1
                    "SWITCH LIST,                  "; % 31 (CLASS = 127)    %117-        17084290 C  0175:0040:1
                                                                               176 IS  132 LONG, NEXT SEG  175
                   L:                                                       %DFB         17085000 T  0175:0040:1
                   END;                                                     %DFB         17086000 T  0175:0041:2
                                                                               175 IS   46 LONG, NEXT SEG  172


                   PROCEDURE OUTPUT2(B,A);                                  %DFB         17087000 T  0172:0053:2
PRT(1207) = OUTPUT2
                     VALUE B;                                               %DFB         17088000 T  0172:0053:2
                     BOOLEAN B;                                             %DFB         17089000 T  0172:0053:2
                     ARRAY A[0];                                            %DFB         17090000 T  0172:0053:2
                     BEGIN      DEFINE PRINTER=LINE#;                       %DFB         17091000 T  0172:0053:2
                                                                              START OF SEGMENT ********** 177
                   LABEL EOF2, SKIP;                                        %116-        17091100 P  0177:0000:0
                   OWN BOOLEAN B2, FWDTOG, LBLTOG, WAITINGFORFWDREF;        %116-        17091110 P  0177:0000:0
PRT(1210) = B2
PRT(1211) = FWDTOG
PRT(1212) = LBLTOG
PRT(1213) = WAITINGFORFWDREF
                   DEFINE MATCH(A,B) = REAL(BOOLEAN(A) EQV BOOLEAN(B)) =    %116-        17091115 C  0177:0000:0
                                       REAL(NOT FALSE)#;                    %116-        17091116 C  0177:0000:0
                   REAL I;                                                  %116-        17091120 C  0177:0000:0
STACK(F+2) = I
                   DEFINE LINECOUNT = TIMINGS[2,0]#; % NUMBER OF LINES PRINTED,          17091140 C  0177:0000:0
                   OWN REAL FWDSEQNO;                                       %116-        17091150 C  0177:0000:0
PRT(1214) = FWDSEQNO
                   IF FIRSTTIME THEN % PRINT HEADINGS AND SAVE TIMINGS,     %116-        17091155 C  0177:0000:0
                     BEGIN                                                  %116-        17091160 C  0177:0000:1
                       FIRSTTIME := FALSE;                                  %116-        17091162 C  0177:0001:2
                       TIME1 := TIME(1);                                    %116-        17091165 C  0177:0002:0
                       DATIME;                                              %116-        17091170 C  0177:0003:2
                       UPDATETIMES(1);                                      %116-        17091175 C  0177:0003:3
```

```
    SAVETIMES(2); % SAVE TIMES FOR START OF XREF PRINT.          %116-    17091180 C  0177!000411
  END;                                                           %116-    17091200 P  0177!000513
IF NOT B2 THEN                                                   %116-    17091210 P  0177!000513
  IF B THEN % END OF SORT - LIST OUT REST OF SEQ. NO.            %116-    17091300 C  0177!000611
    IF XREFPT ≠ 0 THEN % WE GOT SOME TO LIST OUT                 %116-    17091400 C  0177!000712
      BEGIN                                                      %116-    17091500 C  0177!000811
        WRITE(LINE[DBL],15,PAY[*]);                              %116-    17091510 C  0177!000912
        LINECOUNT := LINECOUNT + 1;                              %116-    17091520 C  0177!001313
      END                                                        %116-    17091530 C  0177!001713
    ELSE % NOTHING TO LIST OUT                                   %116-    17091600 C  0177!001713
  ELSE % NOT END OF SORT                                         %116-    17091700 C  0177!001713
    IF NOT MATCH(LASTADDRESS,A[0]) AND A[0].REFIDNOF ≠ 0 AND     %116-    17091800 C  0177!001810
     A[0].REFIDNOF ≥ XREFAY1[8].IDNOF THEN                       %116-    17091900 C  0177!002210
      IF A[0].TYPEREF = FORWARDREF THEN %                        %116-    17092000 P  0177!002411
        WAITINGFORFWDREF := TRUE                                 %116-    17092100 C  0177!002611
      ELSE                                                       %116-    17092200 C  0177!002713
        IF A[0].TYPEREF = LBLREF THEN %                          %116-    17092300 C  0177!002810
          BEGIN                                                  %116-    17092400 C  0177!003010
            LBLTOG := TRUE;                                      %116-    17092500 C  0177!003011
            FWDSEQNO := A[0].SEQNOF;                             %116-    17092600 C  0177!003113
          END                                                    %116-    17092700 C  0177!003313
        ELSE                                                     %116-    17092800 C  0177!003313
          IF A[0].TYPEREF = DECLREF THEN                         %116-    17092900 C  0177!003313
            IF WAITINGFORFWDREF THEN % THIS MUST BE IT           %116-    17093000 P  0177!003513
              BEGIN                                              %116-    17093100 C  0177!003611
                WAITINGFORFWDREF := FALSE;                       %116-    17093200 C  0177!003712
                FWDTOG := TRUE;                                  %116-    17093300 C  0177!003810
                FWDSEQNO := A[0].SEQNOF;                         %116-    17093400 C  0177!003912
  END                                                            %116-    17093500 C  0177!004011
ELSE % ITS A NORMAL DECLARATION - NOT FORWARD                    %116-    17093600 C  0177!004011
  BEGIN                                                          %116-    17093700 C  0177!004011
    IF A[0].REFIDNOF > XREFAY1[8].IDNOF THEN                     %116-    17093850 C  0177!004112
      DO                                                         %116-    17093900 C  0177!004313
        READ(DSK1,10,XREFAY1[*]) [EOF2]                          %116-    17093950 C  0177!004410

PRT(1215) = EOF2

      UNTIL                                                      %116-    17094000 P  0177!004810
        A[0].REFIDNOF ≤ XREFAY1[8].IDNOF;                        %116-    17094050 C  0177!004912
    IF A[0]. REFIDNOF < XREFAY1[8].IDNOF THEN                    %116-    17094100 P  0177!005210
      GO TO SKIP;                                                %116-    17094150 C  0177!005410
    IF XREFPT > 0 THEN % THERE IS STUFF TO PRINT                 %116-    17094200 C  0177!005411
     BEGIN                                                       %116-    17094240 C  0177!005513
      IF SINGLTOG THEN                                           %116-    17094250 C  0177!005610
        WRITE(LINE,15,PAY[*])                                    %116-    17094300 C  0177!005611
      ELSE                                                       %116-    17094350 C  0177!006010
        WRITE(LINE[DBL],15,PAY[*]);                              %116-    17094400 C  0177!006113
      LINECOUNT := LINECOUNT + 1;                                %116-    17094410 C  0177!006611
     END                                                         %116-    17094420 C  0177!007011
    ELSE                                                         %116-    17094450 C  0177!007011
      IF NOT SINGLTOG THEN                                       %116-    17094500 C  0177!007011
        WRITE(LINE);                                             %116-    17094550 C  0177!007210
    XREFPT := 0;                                                 %116-    17094600 C  0177!007611
    BLANKET(PAY[*]);                                             %116-    17094650 C  0177!007713
     SETUPHEADING(XREFAY1[*],PAY[*],XREFAY1[8],                  %116-    17094700 C  0177!007912
      SEGNOF,A[0].SEQNOF,FWDTOG,LBLTOG,                          %116-    17094800 C  0177!008011
      FWDSEQNO,IDTYPE((IF (I :=                                  %116-    17094900 C  0177!008313
      XREFAY1[9].CLASS) ≥ FIELDID THEN                           %117-    17095000 P  0177!008410
      (IDMAX + I - FIELDID + 1) ELSE                             %117-    17095100 P  0177!008610
```

```
                    IF I > IDMAX THEN 0 ELSE I) × 4],          %116-    17095200 P  0177:008811
                    REAL(I ≥ BOOID AND XREFAY1[9],[9:2] = 1),           17095300 C  0177:009210
                    REAL((I ≥ BOOID OR I = LOCLID) AND BOOLEAN          17095310 C  0177:009411
                    (XREFAY1[9],[9:1])), XREFAY1[9],[10:1]);            17095320 C  0177:009610
                  FWDTOG := LBLTOG := FALSE;                    %116-    17095400 C  0177:009912
                  WRITE(LINE,15,PAY[*]);                        %116-    17095500 C  0177:010011
                  LINECOUNT := LINECOUNT + 1;                   %116-    17095510 C  0177:010512
                  BLANKET(PAY[*]);                              %116-    17095550 C  0177:010912
                END                                             %116-    17095600 C  0177:011011
              ELSE % IT MUST BE A NORMAL REFERENCE              %116-    17095700 C  0177:011011
                IF A[0],SEQNOF ≠ LASTADDRESS,SEQNOF THEN        %116-    17095750 C  0177:011011
                  BEGIN                                         %116-    17095800 C  0177:011313
                    ADDASEQNO(A[0],SEQNOF,XREFPT,A[0],[5:1],            17095900 C  0177:011410
                              PAY[*]);                          %116-    17096000 C  0177:011611
                    IF (XREFPT := XREFPT + 1) = 11 THEN %FULL           17096100 C  0177:011713
                      BEGIN                                     %116-    17096200 C  0177:011913
                        WRITE(LINE,15,PAY[*]);                  %116-    17096300 C  0177:012010
                       LINECOUNT := LINECOUNT + 1;              %116-    17096350 C  0177:012411
                        XREFPT := 0;                            %116-    17096400 C  0177:012811
                        BLANKET(PAY[*]);                        %116-    17096450 C  0177:012912
                      END                                       %116-    17096500 C  0177:013011
                  END                                           %116-    17096550 C  0177:013011
                ELSE % REFERENCE TO SAME SEQ. NO. SKIP IT       %116-    17096575 C  0177:013011
            ELSE % THIS IS A REFERENCE TO THE SAME SEQ. NO. - SKIP          17096600 C  0177:013011
          ELSE % HIT END OF IDENTIFIER FILE - JUST SKIP OVER REFERENCES     17096700 C  0177:013112
            EOF2: B2 := TRUE; %  SO SORT CAN GO TO NORMAL EOJ       %116-    17096800 C  0177:013113
          IF NOT B THEN SKIP: LASTADDRESS := A[0];                  %116-    17096850 C  0177:013410
        END OF OUTPUT2;                                             %116-    17096900 C  0177:013610
```

177 IS 141 LONG. NEXT SEG 172

PRT(1216) = HV2

```
        PROCEDURE HV2(A);                                           %DFB     17112000 T  0172:005312
          ARRAY A[0];                                               %DFB     17113000 T  0172:005312
          A[0] := 3"777777777777777"; % BIGGEST FLOATING PT. NO.    %116-    17114000 P  0172:005312
```

PRT(1217) = COMP2

```
        BOOLEAN PROCEDURE COMP2(A,B);                               %DFB     17115000 T  0172:005712
          ARRAY A,B[0];                                             %DFB     17116000 T  0172:005712
            COMP2 := IF A[0],REFIDNOF < B[0],REFIDNOF THEN % DIF IDS        17117000 P  0172:005712
                       TRUE                                         %116-    17117100 C  0172:005912
                     ELSE                                           %116-    17117200 C  0172:006010
                       IF A[0],REFIDNOF = B[0],REFIDNOF THEN        %116-    17117300 C  0172:006011
                         IF A[0],[1:4] LSS B[0],[1:4] THEN          %116-    17117400 C  0172:006211
                           TRUE                                     %116-    17117500 C  0172:006513
                         ELSE                                       %116-    17117600 C  0172:006610
                           IF A[0],[1:4] = B[0],[1:4] THEN          %116-    17117700 C  0172:006611
                             IF A[0],SEQNOF < B[0],SEQNOF THEN      %116-    17117702 C  0172:006912
                               TRUE                                 %116-    17117704 C  0172:007113
                             ELSE                                   %116-    17117706 C  0172:007211
                               IF A[0],SEQNOF = B[0],SEQNOF THEN    %116-    17117708 C  0172:007312
                                 BOOLEAN(A[0],[5:1])                %116-    17117710 C  0172:007512
```

```
                                  ELSE                                    %116-          17117712 C  0172:0076:1
                                    FALSE                                 %116-          17117714 C  0172:0077:2
                              ELSE                                        %116-          17117720 C  0172:0077:3
                                FALSE                                     %116-          17117730 C  0172:0078:0
                          ELSE                                            %116-          17117800 C  0172:0078:0
                            FALSE;                                        %116-          17117900 C  0172:0078:1


             SAVETIMES(1); % SAVE TIMES FOR START OF REFERENCES SORT      %116-          17117910 C  0172:0081:2
             FIRSTTIME := TRUE; % LET OUTPUT PROCEDURE KNOW ABOUT FIRST CAL              17117920 C  0172:0083:2
             XREFPT:=29;         REWIND(DSK2);                            %DFB           17118000 T  0172:0084:0
             SORT(OUTPUT2,INPUT2,0,HV2,COMP2,1,6000);                     %116-          17119000 P  0172:0086:1
            UPDATETIMES(2); % UPDATE TIMES FOR PRINTING CROSS REFERENCE%116-             17119100 C  0172:0109:2
            PRINTXREFSTATISTICS;                                          %116-          17119200 C  0172:0110:0
        END;                                                              %DFB           17120000 T  0172:0111:2
PRT(1220) = *SEGMENT DESCRIPTOR*
                                                                                   172 IS  116 LONG, NEXT SEG  167
        END;                                                              %DFB           17121000 T  0167:0032:0
PRT(1221) = *SEGMENT DESCRIPTOR*
                                                                                   167 IS   36 LONG, NEXT SEG    3
                        END MAIN BLOCK;                                                  17121500 T  0003:0988:0
PRT(1222) = *SEGMENT DESCRIPTOR*
                                                                                     3 IS  992 LONG, NEXT SEG    2
        END.                                                              %DFB           17122000 T  0002:0056:0
                                                                                     2 IS   59 LONG, NEXT SEG    1
PRT(714) = EXP INTRINSIC, SEGMENT NUMBER =  178.
PRT(713) = LN INTRINSIC, SEGMENT NUMBER =  179.
PRT(445) = OUTPUT(W) INTRINSIC, SEGMENT NUMBER =  180.
PRT(5) = BLOCK CONTROL INTRINSIC, SEGMENT NUMBER =  181.
PRT(463) = INPUT(W) INTRINSIC, SEGMENT NUMBER =  182.
PRT(1167) = SORT INTRINSIC, SEGMENT NUMBER =  183.
PRT(715) = X TO THE I INTRINSIC, SEGMENT NUMBER =  184.
PRT(503) = GO TO SOLVER  INTRINSIC, SEGMENT NUMBER =  185.
PRT(14) = ALGOL WRITE   INTRINSIC, SEGMENT NUMBER =  186.
PRT(15) = ALGOL READ    INTRINSIC, SEGMENT NUMBER =  187.
PRT(16) = ALGOL SELECT  INTRINSIC, SEGMENT NUMBER =  188.
PRT(1166) = MERGE INTRINSIC, SEGMENT NUMBER =  189.
PRT(526) = DYNAMIC DIALS INTRINSIC, SEGMENT NUMBER =  190.
PRT(241) = FILE ATTRBUTS INTRINSIC, SEGMENT NUMBER =  191.
                                                                                     1 IS    2 LONG, NEXT SEG    0
                                                                                   192 IS   69 LONG, NEXT SEG    0
NUMBER OF ERRORS DETECTED =    1.  COMPILATION TIME =  532 SECONDS.              01831000

PRT SIZE = 659; TOTAL SEGMENT SIZE = 13254 WORDS; DISK SIZE = 720 SEGS; NO. PGM. SEGS = 192

ESTIMATED CORE STORAGE REQUIRED = 27332 WORDS.

ESTIMATED AUXILIARY MEMORY REQUIRED =      0 WORDS.

NUMBER OF CARD-IMAGES PROCESSED =  12712.
```

ALGOL /NUDISK

SOURCE FILE: SYMBOL /ALGOL

```
A -- STREAM VARIABLE -- NAME PARAMETER -- DECLARED IN SEGMENT 3 AT 01561620
     01561640
A -- STREAM VARIABLE -- NAME PARAMETER -- DECLARED IN SEGMENT 3 AT 01733000
     01734000
A -- STREAM VARIABLE -- NAME PARAMETER -- DECLARED IN SEGMENT 3 AT 01735000
     01736200
A -- STREAM VARIABLE -- NAME PARAMETER -- DECLARED IN SEGMENT 3 AT 02062500
     02063500
A -- STREAM VARIABLE -- NAME PARAMETER -- DECLARED IN SEGMENT 15 AT 02238130
     02238150
A -- STREAM VARIABLE -- NAME PARAMETER -- DECLARED IN SEGMENT 15 AT 02238180
     02238200
A -- STREAM VARIABLE -- VALUE PARAMETER -- DECLARED IN SEGMENT 15 AT 02238212
     02238214
A -- REAL ARRAY -- DECLARED IN SEGMENT 17 AT 02264100
     02265100  02265150  02265250  02265300  02265600  02265650  02265700  02265750  02265850  02265900  02265950
     02266000
A -- REAL ARRAY -- DECLARED IN SEGMENT 29 AT 02927140
     02927680  02927700  02927710
A -- REAL -- VALUE PARAMETER -- DECLARED IN SEGMENT 3 AT 03047000
A -- REAL -- VALUE PARAMETER -- DECLARED IN SEGMENT 3 AT 03049000
A -- INTEGER -- VALUE PARAMETER -- DECLARED IN SEGMENT 3 AT 03050000
A -- INTEGER -- VALUE PARAMETER -- DECLARED IN SEGMENT 3 AT 04311000
     04311050  04311060  04314000 *04315000*
A -- REAL -- VALUE PARAMETER -- DECLARED IN SEGMENT 3 AT 04500000
     04529000  04534000  04542000  04548000  04550000  04551000  04556000
A -- INTEGER -- VALUE PARAMETER -- DECLARED IN SEGMENT 3 AT 04609000
     04614000  04616000
A -- REAL -- DECLARED IN SEGMENT 47 AT 05130000
     *05131000* 05132000
A -- REAL -- DECLARED IN SEGMENT 62 AT 06282400
     06285800 *06286800* 06287200 *06287800* 06288000
A -- REAL -- VALUE PARAMETER -- DECLARED IN SEGMENT 62 AT 06283000
     06283400  06283600
A -- REAL ARRAY -- NAME PARAMETER -- DECLARED IN SEGMENT 83 AT 07657000
     07664000  07666500  07673000  07673500  07675000  07677500
A -- DEFINE -- DECLARED IN SEGMENT 89 AT 08015000
     08181000  08188000
A -- REAL -- VALUE PARAMETER -- DECLARED IN SEGMENT 89 AT 08020000
     08021000  08021100  08021200
A -- REAL -- NAME PARAMETER -- DECLARED IN SEGMENT 89 AT 08027000
     *08035000**08037000*
A -- INTEGER -- DECLARED IN SEGMENT 103 AT 08930000
     *08955000* 08957000  08970000  08977000  08980000  08985000  08993500
A -- REAL -- DECLARED IN SEGMENT 137 AT 13359000
     *13364000* 13365000  13366000
```

```
A  --  REAL  --  DECLARED IN SEGMENT 142 AT 13903100
       *13908000* 13911500
A  --  REAL ARRAY  --  NAME PARAMETER  --  DECLARED IN SEGMENT 170 AT 17007000
        17006000  17010000
A  --  REAL ARRAY  --  NAME PARAMETER  --  DECLARED IN SEGMENT 170 AT 17019000
        17016000  17025000  17025100  17025300 *17025400* 17026000
A  --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 170 AT 17029000
        17031000  17033400
A  --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 170 AT 17036000
        17038000  17040000
A  --  REAL ARRAY  --  NAME PARAMETER  --  DECLARED IN SEGMENT 170 AT 17042200
        17042100  17042300  17042350
A  --  REAL ARRAY  --  NAME PARAMETER  --  DECLARED IN SEGMENT 170 AT 17042500
        17042400  17042600
A  --  REAL ARRAY  --  NAME PARAMETER  --  DECLARED IN SEGMENT 172 AT 17071000
        17070000 *17080000*
A  --  REAL ARRAY  --  NAME PARAMETER  --  DECLARED IN SEGMENT 172 AT 17090000
        17087000  17091115  17091800  17091900  17092000  17092300  17092600  17092900  17093400  17093850  17094050
        17094100  17094800  17095750  17095900  17096850
A  --  REAL ARRAY  --  NAME PARAMETER  --  DECLARED IN SEGMENT 172 AT 17113000
        17112000 *17114000*
A  --  REAL ARRAY  --  NAME PARAMETER  --  DECLARED IN SEGMENT 172 AT 17116000
        17115000  17117000  17117300  17117400  17117700  17117702  17117708  17117710
AAW  --  REAL  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 05380000
        05381000  05382000  06061000
AC  --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 2 AT 00512100  --  OCCURS AT 06067000
        00512300
ACC  --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 130 AT 12154000
        12156000  12160000
ACCUM  --  ALPHA ARRAY  --  DECLARED IN SEGMENT 3 AT 01304000
        02013180 *02013195**02013200* 02013205  02013210 *02013290* 02013295 *02013330**02013335* 02013360 *02013365*
       *02013390* 02013410 *02013420* 02013425  02128500 *02238340* 02238385 *02238390**02238420* 02238440 *02238500*
        02251000  02255000  02259000 *02327000* 02366100 *02371000**02499000* 02502000  02581000 *02590000* 02592000
        02602600 *02647000* 02650000  02652000  02697500  02697600  02698000 *02700000* 02703050  02705000  02754000
        02761500  02761501  02770000  02772000  02774000  02776300  02776400  02777000  02777050  02781000  02822000
        02846000  02865000  02873000  02881000 *02934000* 02942000 *02948000* 02952000 *02958500**05043000* 05044000
        05325460  07673500  07676000  07677000  07677500  08389000  08390500  08412010  08412020  08413012  08414012
        08416012  08450010  08619080  08621002  08624002  08629002  08658010  08812100  10141000 *10167300* 10264750
        10269000  10274000 *10573000**12161500* 12162000  12164000  13080000  13088030  13088060  13091000  13095000
        13281000  13282000 *13307000**13308000**13309000* 13310000  13310075  13310080 *13372000* 13908000 *13911500*
        14259000  14259080 *14259090* 14260000  14269280  14440000  14441000  14507040  14507105  14507170  16211100
        16211300  16213000  16384900  16387000  16392000
ACCUM  --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 3 AT 02001831
        02001833
ACCUM  --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 10 AT 02089500
        02122500
ACCUM  --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 45 AT 05016000
        05028000
ACCUM  --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 121 AT 10236000
        10239000
ACCUM  --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 131 AT 13005000
        13008000
ACCUM1  --  DEFINE  --  DECLARED IN SEGMENT 97 AT 08580000
        08619080  08619090  08619095  08620000  08621000  08621010  08621020
ACCUM1  --  REAL  --  DECLARED IN SEGMENT 131 AT 13039500
       *13088030* 13088035  13088040  13088050 *13088060* 13088070  13088075  13088082
ACC1  --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 3 AT 02001020
```

```
        02001130
ACLASS  --  INTEGER  --  DECLARED IN SEGMENT 70 AT 07050000
       *07079000* 07099000  07100000  07105000 *07106000* 07107000 *07109000**07111240* 07111260 *07111285* 07122500
       *07122540**07145000**07155000**07192000* 07204000  07215000  07216000  07217000  07217500  07229000 *07318000*
       *07324000**07330000**07343000**07355000**07377000**07381000*
ACTUALPARAPART  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 07045000
        07385000  07410000  07454000
AD  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 02013060
        02013065  02013100
ADD  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01633000
        05226000  06016000  06062560  06377000  07366000  07625000  08126000  08214000  08415010  08653110  08653140
        08653210  08953000  10961000  13545000  13758000  14565000  15273300
ADD  --  REAL  --  DECLARED IN SEGMENT 134 AT 13221000
       *13248000* 13250000  13252000 *13259000* 13261000
ADDASEQNO  --  STREAM PROCEDURE  --  DECLARED IN SEGMENT 172 AT 17051400
        17095900
ADDC  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 13212000
        13490000  13530000  13577000
ADDCON  --  REAL  --  DECLARED IN SEGMENT 138 AT 13451000
       *13490000**13492000* 13494000  13530000 *13577000**13579000* 13580000
ADDER  --  STREAM PROCEDURE  --  DECLARED IN SEGMENT 3 AT 02013060
        02013140  02013713  02201000
ADDITIONAL  --  DEFINE  --  DECLARED IN SEGMENT 79 AT 07595000
        07601000  07603000  07610000  07614000  07622000  07627000
ADDOP  --  DEFINE  --  DECLARED IN SEGMENT 157 AT 16005000
        16326000
ADDR  --  REAL  --  DECLARED IN SEGMENT 159 AT 16068000
       *16072000**16074000* 16077000
ADDR  --  REAL  --  DECLARED IN SEGMENT 162 AT 16195000
       *16196000**16201000**16204000* 16209000
ADDR  --  INTEGER  --  DECLARED IN SEGMENT 164 AT 16314000
       *16321000**16344000* 16352000
ADDR  --  INTEGER  --  DECLARED IN SEGMENT 165 AT 16364000
       *16371000**16376000**16378000* 16379500  16383000  16384800  16385000  16389000
ADDR  --  REAL  --  DECLARED IN SEGMENT 166 AT 16405000
       *16408000**16412000**16414000* 16415000  16417000
ADDRES  --  INTEGER  --  DECLARED IN SEGMENT 48 AT 05138000
       *05140000* 05143000  05147000
ADDRES  --  OWN REAL  --  DECLARED IN SEGMENT 89 AT 08010000
       *08055000* 08067000  08068000  08075000  08179000  08212000
ADDRESS  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01152000
        02850000  02923000  04508000  05131000  05140000  05175000  05187570  05217000  05233000  05280000  05281000
        06016000  06062155  06062350  06117000  06120100  06232000  06283600  06286200  07122510  07122520  07306000
        07319000  07430000  07463500  07521000  07663500  07926000  08021200  08029000  08044000  08055000  08488000
        08489550  08692000  08696520  08739000  08809000  08910000  08915000  08917000  08970000  08976000  08989000
        08992000  09294000  09296100  09311000  09313000  10277000  10309000  10569000  10598000  10796000  10826000
        10835000  10868000  10874000  12015000  12024000  12036000  12047000  12061000  12061725  12062600  12064500
        13086000  13110000  13205000  13248000  13259000  13476000  13483000  13517000  13575000  14085000  14092000
        14100000  14115000  14129000  14269520  14269540  14349000  14399000  14430000  14437000  14454000  15094000
        15097000  15107000  15224000  15267000  15344000  15352000  15353000  16072000  16121000  16124000  16200000
        16204000  16212600  16286000  16321000  16329000  16342000  16344000  16370000  16376000  16408000  16412000
        16435000
ADDRESS  --  INTEGER  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 04017000
        04018000  04019000
ADDRESS  --  INTEGER  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 04022000
        04023000  04024000
ADDRESS  --  INTEGER  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 04030000
```

```
        04028000  04029000  04032000  04033000
ADDRESS  --  REAL  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 05150000
        05148000  05149000  05153000  05155000
ADDRESS  --  INTEGER  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 05199000
        05201000
ADDRESS  --  REAL  --  DECLARED IN SEGMENT 71 AT 07393000
       *07397000* 07413000
ADDRS  --  INTEGER  --  DECLARED IN SEGMENT 72 AT 07429000
       *07430000* 07454000
ADDRSF  --  REAL  --  DECLARED IN SEGMENT 3 AT 01604000
        06404000  07217510  07225000 *07255000* 07311000 *07312000**07316000* 07335000  07346000  07348000  07358000
        07377000  07380000  07397000  13054000  13068000  13081500  13196410  13196560 *13205000**13343000**13344000*
       *13345000* 13347500  13347520  13373000 *13913000* 13915000  14212000  14313000  14320000 *14558000* 14601000
ADDVALUF  --  INTEGER  --  DECLARED IN SEGMENT 3 AT 01000860
        02193000  02234800 *02592000**09028920*
ADES  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01299000
        10616000  10801000  10978500
ADJUST  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 04083000
        04087000  04144000  04172000  07111315  07459000  07484100  07492100  07597000  07646705  08190000  10365100
        10800000  10981000  10988000  10993000  13605000  13611000  13630000  13752000  14563000  14600000  16048000
        16137000  16158000
ADOP  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01276000
        06013000  06036000  06226000  06286200  06286400  07668500  08029000  08031000  12013000  12022000  12060800
        13474000  16326000
ADR  --  REAL  --  DECLARED IN SEGMENT 81 AT 07646390
       *07646460* 07646510  07646545  07646570  07646705
ADRES  --  INTEGER  --  DECLARED IN SEGMENT 51 AT 05215000
       *05217000**05219000* 05226000  05227000
ADRES  --  DEFINE  --  DECLARED IN SEGMENT 87 AT 07922000
        07926000
AEXP  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 06002000  --  FORWARD AT 03001000
        06007000  06062155  06062165  06062170  06062185  06062195  06072040  06072060  06072220  06072240  06072260
        06073500  06073830  06124000  06238000  06288600  06304000  06319000  06328500  06332100  06416000  06422000
        06509000  07464750  07468000  07470500  07646430  07851000  07903000  07907000  07937000  08101000  08123000
        08131000  08413040  08414022  08415010  08416022  08418200  08418300  08451000  08493750  08519000  08613200
        08613300  08621010  08625000  08627000  08630000  08659000  08918000  08985000  08993000  08994060  08995000
        10616000  10812000  10836000  12046000  12051000  12054000  12064400  12064800  13088090  13088100  13111000
        13148000  13157010  13501000  13524000  15099000  15277000  15307000
AGAIN  --  LABEL  --  DECLARED IN SEGMENT 15 AT 02238120  --  OCCURS AT 02238330
        02238360  02238460  02238530  02238555
AGAIN  --  LABEL  --  DECLARED IN SEGMENT 21 AT 02360000  --  OCCURS AT 02372000
        02377000  02390000  02393000  02395000  02397000  02404000  02418000  02419120  02419200  02421000  02431000
        02437500  02460000  02470000  02472000  02474000  02480000  02493000  02496000  02501000  02506000  02508000
        02513000  02516000  02520000  02522000  02564000  02567000  02574200  02577000  02601000
AGAIN  --  LABEL  --  DECLARED IN SEGMENT 85 AT 07716000  --  OCCURS AT 07728000
        07739000
AGIN  --  STREAM LABEL  --  DECLARED IN SEGMENT 3 AT 02001860  --  OCCURS AT 02001886
        02001880
AJUMP  --  BOOLEAN  --  DECLARED IN SEGMENT 3 AT 01596000
        13603000 *13614000* 13626000 *13629000* 14054000 *14055100**14606000*
AJUMPO  --  BOOLEAN  --  DECLARED IN SEGMENT 143 AT 14034000
       *14054000* 14606000
AKKUM  --  REAL  --  DECLARED IN SEGMENT 3 AT 01693000
        09361005  09418000 *13644000*
ALFA  --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 104 AT 09010000
        09017000
ALFAARRAYID  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01204000
ALFAID  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01200000
```

```
        14139000
ALFAPROCID  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01196000
ALFASTRPROCID  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01192000
ALFAV  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01284000
        13075000
ALL  --  REAL  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 89 AT 08052000
        08054000  08055000  08056000  08057000
ALLTHU  --  LABEL  --  DECLARED IN SEGMENT 39 AT 04168000  --  OCCURS AT 04186000
        04180000
ALONG  --  LABEL  --  DECLARED IN SEGMENT 22 AT 02325000  --  OCCURS AT 02358000
        02336000  02340000  02354000
ALPHADEC  --  LABEL  --  DECLARED IN SEGMENT 143 AT 14013000  --  OCCURS AT 14139000
        14018000
ALPHASIZE  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01707000
        15035000
ALPHAWORDS  --  STREAM PROCEDURE  --  DECLARED IN SEGMENT 17 AT 02264700
      . 02264900  02265600  02265850
AMPERSAND  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01270000
        06034000  06128500  06170000  06267500  15349000
ANDOP  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01274000
        02964500  02971000  02983500  06175000  06193000
ANOTHER  --  LABEL  --  DECLARED IN SEGMENT 69 AT 07007000  --  OCCURS AT 07009000
        07012000  07015000  07024000
ANOTHER  --  LABEL  --  DECLARED IN SEGMENT 70 AT 07065000  --  OCCURS AT 07079000
        07237000
ARC  --  BOOLEAN  --  DECLARED IN SEGMENT 97 AT 08591600
       *08597450* 08611100  08658100  08698000
ARGH  --  LABEL  --  DECLARED IN SEGMENT 27 AT 02638000  --  OCCURS AT 02649000
        02696000  02718000  02753000  02899000
ARITHCOMP  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 06041000  --  FORWARD AT 03004000
        06036000  06052000  06055000
ARITHSEC  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 06011000  --  FORWARD AT 03002000
        06006000  06023000
ARPROGS  --  REAL  --  DECLARED IN SEGMENT 138 AT 13449000
      *13479000**13509000* 13529000
ARRAE  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 13376000
        14156000
ARRAYCHECK  --  BOOLEAN PROCEDURE  --  DECLARED IN SEGMENT 3 AT 05380000
      *05381000* 05392000  05399000  05405000  05406000  08943000
ARRAYDEC  --  LABEL  --  DECLARED IN SEGMENT 143 AT 14015000  --  OCCURS AT 14156000
        14020000
ARRAYFLAG  --  BOOLEAN  --  DECLARED IN SEGMENT 3 AT 01000840
        06100100  06103100  06106100  06243100  06246100  06249100 *13106000**13181000**13452100**13588000* 13819410
ARRAYMONFILE  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01526000
        15325000
ARRAYV  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01294000
ASKIP  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 121 AT 10236000
        10237000  10239000
ASSIGNOP  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01269000
        07401000  07924000  08178000  08229000  08493680  12028000  12062000  13051000  13196380  13787000  13911100
        14219000  15090000  15113000  15300000  15342000  16316000  16367000
ASSIGNREF  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01007500
        07687600  07835500  08178100  15091100  15116000  15301100  15342300  16318500
ASSOP  --  BOOLEAN  --  DECLARED IN SEGMENT 95 AT 08493500
      *08493530**08493680* 08493850
ATSIGN  --  LABEL  --  DECLARED IN SEGMENT 27 AT 02637000  --  OCCURS AT 02683000
        02641000
```

```
ATTRIBUTEINDX  --  REAL  --  DECLARED IN SEGMENT 95 AT 08493490
      *08493530**08493610* 08493710  08493740  08493790  08493795  08493800  08493805  08493810  08493812  08493820
       08493830  08493840  08493880
ATYPE  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01452000
       06005000  06092015  06143000  06239000  06304000  07111240  07145000  07204000  08493890
AUXMEMERR  --  LABEL  --  DECLARED IN SEGMENT 143 AT 14016000  --  OCCURS AT 14136100
       14021000
AUXMEMREQ  --  INTEGER  --  DECLARED IN SEGMENT 3 AT 01411010
       09419000 *13645000*
AUXMEMV  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01298500
       13466000  13724000
AWAY  --  LABEL  --  DECLARED IN SEGMENT 11 AT 02191000  --  OCCURS AT 02196250
       02191250  02194250  02195250
B  --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 3 AT 01561620
       01561640
B  --  REAL  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 01717000
B  --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 3 AT 01735000
       01736200
B  --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 3 AT 02062500
       02063500
B  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 15 AT 02238130
       02238160
B  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 15 AT 02238180
       02238200
B  --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 15 AT 02238212
       02238214
B  --  BOOLEAN  --  DECLARED IN SEGMENT 22 AT 02323000
       *02343000* 02352000  02353000 *02356000* 02357000
B  --  BOOLEAN  --  NAME PARAMETER  --  DECLARED IN SEGMENT 3 AT 02955500
B  --  BOOLEAN  --  DECLARED IN SEGMENT 33 AT 02970000
       *02970500* 02971000  02971500
B  --  BOOLEAN  --  DECLARED IN SEGMENT 34 AT 02973500
       *02976000**02978000**02978500* 02979000
B  --  BOOLEAN  --  NAME PARAMETER  --  DECLARED IN SEGMENT 3 AT 02980000
       *02983500* 02984000  02984500
B  --  REAL  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 03047000
B  --  INTEGER  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 03050000
B  --  REAL  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 04247000
       04248000
B  --  INTEGER  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 04311000
       04311070  04311080  04316000 *04317000*
B  --  REAL  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 04500000
       04526100  04528000  04542000  04544000  04548000  04551000  04556000
B  --  BOOLEAN  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 43 AT 04505000
       04503000  04507000
B  --  BOOLEAN  --  DECLARED IN SEGMENT 57 AT 06059050
       06062280 *06062290* 06062435 *06062500*
B  --  OWN REAL  --  DECLARED IN SEGMENT 89 AT 08010000
       08044000  08047000  08089000  08112000  08116000 *08122000**08169000* 08183000  08199000 *08209000* 08213000
B  --  BOOLEAN  --  NAME PARAMETER  --  DECLARED IN SEGMENT 89 AT 08027000
       *08035000**08037000*
B  --  BOOLEAN  --  DECLARED IN SEGMENT 127 AT 12004100
       *12034100* 12038000
B  --  BOOLEAN  --  DECLARED IN SEGMENT 128 AT 12060250
       *12062250* 12063300
B  --  BOOLEAN  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 170 AT 17018000
       17016000  17017000  17021000
```

```
B  --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 170 AT 17029000
       17032000  17033500
B  --  REAL ARRAY  --  NAME PARAMETER  --  DECLARED IN SEGMENT 170 AT 17042200
       17042100  17042300  17042350
B  --  BOOLEAN  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 172 AT 17089000
       17087000  17088000  17091115  17091300  17096850
B  --  REAL ARRAY  --  NAME PARAMETER  --  DECLARED IN SEGMENT 172 AT 17116000
       17115000  17117000  17117300  17117400  17117700  17117702  17117708
BACK  --  INTEGER  --  DECLARED IN SEGMENT 75 AT 07491000
      *07493000* 07495000
BACK  --  REAL  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 90 AT 08084000
       08082000  08083000  08089000
BACK  --  LABEL  --  DECLARED IN SEGMENT 120 AT 10259000  --  OCCURS AT 10263300
       10276500  10278000  10279000
BACKFIX  --  INTEGER  --  DECLARED IN SEGMENT 90 AT 08091000
      *08097000**08099000**08106000* 08127000 *08146000**08147000* 08157000  08168000  08171000
BACKUP  --  INTEGER  --  DECLARED IN SEGMENT 36 AT 04046000
      *04052000**04056000* 04068000  04074000
BAE  --  INTEGER PROCEDURE  --  DECLARED IN SEGMENT 3 AT 07458000  --  FORWARD AT 03046000
       07129000  07194000  07270000  07304000 *07459000* 10347000
BANA  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 06419000  --  FORWARD AT 03048000
       05143000  05173000  05187600  05225000  06399000  06423000  07122500  07519000  08489000  08653110  08653140
       08653210  08693000
BASENUM  --  INTEGER  --  DECLARED IN SEGMENT 3 AT 01000860
       02192500 *02581000**09028920*
BATMAN  --  REAL  --  DECLARED IN SEGMENT 3 AT 01001600
      *02961500**02966500**02967000* 02968000  02978000
BBC  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01634000
       07487000  08221000  13769000
BBW  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01635000
       07111320  07495000  07523000  08047000  08078000  08166000  08168000  08923000  10997000  14571000
BEF  --  LABEL  --  DECLARED IN SEGMENT 126 AT 10956000  --  OCCURS AT 10978000
       10964000
BEFORE  --  BOOLEAN  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 10954100
       10954000  10954100  10964000
BEGINCTR  --  INTEGER  --  DECLARED IN SEGMENT 3 AT 01499000
      *07008000**07021000* 07023000 *07024000**14035000**14510000**14520000*
BEGINPRINT  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 02196510
       02196610  02910600
BEGINSTACK  --  REAL ARRAY  --  DECLARED IN SEGMENT 3 AT 01007600
       02196580 *02910300*
BEGINV  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01234000
       02910200  05108000  07646440  09252000  14393000  14479000  16221000
BEND  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01001462
       02910600
BENDBIT  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01001171
       02419200  02910600
BETA1  --  LABEL  --  DECLARED IN SEGMENT 138 AT 13452000  --  OCCURS AT 13472000
       13567000
BETA2  --  LABEL  --  DECLARED IN SEGMENT 138 AT 13452000  --  OCCURS AT 13568000
       13478000
BEXP  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 06133000  --  FORWARD AT 03006000
       06120000  06305000  06410000  07487000  07493000  08135000  08142000  08493750  15099000  15307000
BFC  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01636000
       06072090  06301000  07496000  07572000  07573000  07575000  07579000  07583000  07586000  08151000  10928000
       10971000  10994000
BFW  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01637000
```

```
        04218000  06307000  07111325  07138000  07201000  07272000  07311000  07312000  07509000  07524000  07526000
        07529100  07529200  07572000  07586000  07609700  07646460  07646730  07646800  07727080  08125000  08127000
        08146000  08148000  08215000  08228000  08918000  08923000  10350000  10366000  10994000  13263000  13274000
        13606000  14046000  14558500  15037000
BIT  --  INTEGER  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 02307000
        02311000
BIT  --  DEFINE  --  DECLARED IN SEGMENT 157 AT 16015000
        16216000
BLANKET  --  STREAM PROCEDURE  --  DECLARED IN SEGMENT 3 AT 01737350
        01737500  02226750  02265100  02265250  02265600  02265700  02265850  02265950  02927680  05264000  05325450
        09034500  09300150
BLANKET  --  STREAM PROCEDURE  --  DECLARED IN SEGMENT 172 AT 17052600
        17053200  17083000  17094650  17095550  17096450
BLKAD  --  INTEGER  --  DECLARED IN SEGMENT 143 AT 14026000
       *14036000**14044000* 14045000  14555000  14575000
BLOCK  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 14001000  --  FORWARD AT 03067000
        07734000  07755000  09275000  14482000  14613000  15127000  15376000  17121500
BLOCKCTR  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01430000
        08953000  08954000  13756000  13761000  14542000  14545000  14564100  14565000
BNS  --  DEFINE  --  DECLARED IN SEGMENT 157 AT 16011000
        16036000  16122000  16124000
BNSFIX  --  REAL  --  DECLARED IN SEGMENT 161 AT 16118000
       *16121000* 16141000
BOO  --  BOOLEAN  --  DECLARED IN SEGMENT 84 AT 07658500
       *07668500* 07669500  07670500
BOOARRAYID  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01202000
        05381000  06062140  06062390  06073450  06094950  07100000  07150000  07155000  07192000  07204000  07292000
        07464250  07684500  07814000  07834000  07939000  08401000  08457000  08597100  08665000  08728100  10310000
        10323000  10579000  10805000  12062500  12063000  12064200  13330550  13346000  14095000  15307000
BOOCOMP  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 06181000  --  FORWARD AT 03010000
        06175000  06176000  06193000  06210000
BOOFINISH  --  LABEL  --  DECLARED IN SEGMENT 122 AT 10292000  --  OCCURS AT 10326000
        10323000
BOOID  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01198000
        02967000  02977500  05398000  06062140  06062370  07111250  07111260  07111285  07146000  07192000  07377000
        08057000  08058000  10307000  10574000  10872000  12042000  12062500  12062700  12064200  14140000  15099000
        17095300  17095310
BOOLCOMP  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 02980000  --  FORWARD AT 02955500
        02971000  02983000  02985000
BOOLEANDEC  --  LABEL  --  DECLARED IN SEGMENT 143 AT 14013000  --  OCCURS AT 14140000
        14018000
BOOLEXP  --  BOOLEAN PROCEDURE  --  DECLARED IN SEGMENT 3 AT 02969000  --  FORWARD AT 02065600
        02343000 *02971500* 02972000  02976000
BOOLPRIM  --  BOOLEAN PROCEDURE  --  DECLARED IN SEGMENT 3 AT 02972500  --  FORWARD AT 02955000
        02970500 *02979000* 02979500  02982500
BOOPRIM  --  INTEGER PROCEDURE  --  DECLARED IN SEGMENT 3 AT 06213000  --  FORWARD AT 03011000
        06153000  06173000 *06227520**06239000**06241000**06253000**06259000* 06271000 *06274000* 06278000
BOOPROCID  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01194000
        05403000  06062370  07463000  08941000  10654000  12062700  14295000  15019000  15101000
BOOSEC  --  INTEGER PROCEDURE  --  DECLARED IN SEGMENT 3 AT 06150000  --  FORWARD AT 03008000
        06145000 *06153000* 06160000  06190000
BOOSTRPROCID  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01190000
        07088000  07089000  14285000
BOOV  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01282000
        06232000
BRANCH  --  INTEGER  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 03037000
        03036000
BRANCH  --  INTEGER  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 04116000
```

```
        04114000  04115000 *04120400* 04122000  04123000
BRANCHTYPE  --  REAL  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 03025000
        03023000  03024000
BRANCHTYPE  --  REAL  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 07537000
        07535000  07536000  07540000  07544000
BRET  --  OWN REAL  --  DECLARED IN SEGMENT 89 AT 08011000
       *08017800* 08017900 *08033000* 08034000  08035000  08046000 *08094000**08124000**08180000**08223000**08232000*
BRNCH  --  LABEL  --  DECLARED IN SEGMENT 90 AT 08092000  --  OCCURS AT 08152000
        08148000
BS  --  LABEL  --  DECLARED IN SEGMENT 70 AT 07066000  --  OCCURS AT 07229000
        07111340  07112000  07122540  07146000  07167000  07174000  07193000  07217510  07251000  07261000  07335000
        07352000  07359000  07368000  07369000  07382000
BSD  --  DEFINE  --  DECLARED IN SEGMENT 157 AT 16021000
        16417000
BSPOINT  --  INTEGER  --  DECLARED IN SEGMENT 3 AT 01007600
        02196580 *02910300**02910700*
BSS  --  DEFINE  --  DECLARED IN SEGMENT 157 AT 16020000
        16415000
BSX  --  LABEL  --  DECLARED IN SEGMENT 70 AT 07066000  --  OCCURS AT 07382000
        07377000
BSXX  --  LABEL  --  DECLARED IN SEGMENT 70 AT 07066000  --  OCCURS AT 07381000
        07384000
BT  --  BOOLEAN  --  DECLARED IN SEGMENT 140 AT 13736000
       *13739000* 13750000  13753000
BTYPE  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01452000
        06133000  06145000  06157000  06173000  06190000  06241000  06265000  06271000  06274000  07145000  07204000
        07210000  07211000  08493890
BUFFACCUM  --  INTEGER  --  DECLARED IN SEGMENT 3 AT 01583000
BUFFSIZE  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01338000
        01561020  01561030  01561040
BUG  --  FORMAT  --  DECLARED IN SEGMENT 4 AT 01802000
        04279200  04280000  04284200  04285000
BUGGER  --  ALPHA PROCEDURE  --  DECLARED IN SEGMENT 3 AT 04600000  --  FORWARD AT 03100000
        04284300  04286000 *04603000**04605000* 04608000
BUILDLINE  --  BOOLEAN  --  DECLARED IN SEGMENT 2 AT 00504700
        02236000  02331050 *02331060* 02525000 *02525008**02525013**07646635**07646685**07694500**07699000**09028100*
       *09214985**09252050**09314100**09315100* 09399100  13653000 *13777700**13818600*
BUMPL  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01477000
        04216000  06297000  06300000  07111320  07459000  07487000  07493000  07495000  07522000  07540000  07577000
        07585000  07609500  07646780  07727070  08047000  08078000  08106000  08146000  08168000  08190000  08221000
        08914000  08918000  08923000  10802000  10971000  10978000  10997000  13628000  13768000  14570000  15033000
BUP  --  INTEGER  --  DECLARED IN SEGMENT 3 AT 01695000
       *13333000* 14073000 *14079000**14083000* 14084000 *14088000* 14099000  14104000  14109000  14116000 *14118000*
       *14364000**14369000* 14507010 *14507180*
B2  --  OWN BOOLEAN  --  DECLARED IN SEGMENT 177 AT 17091110
        17091210 *17096800*
B2D  --  ALPHA PROCEDURE  --  DECLARED IN SEGMENT 3 AT 04247000  --  FORWARD AT 01717000
        04149000 *04248000* 04279200  04280000  04281000  04284200  04284400  04285000  04287000  05264500  05376100
        09301000  10074000
C  --  REAL  --  DECLARED IN SEGMENT 3 AT 01562000
       *02682000**02688000**02701000* 02705000 *02758000* 02761000  02767000  02771000  02776100 *02777000**02789000*
       *02801000**02803000**02811000* 02827000 *02842000**02845000* 02849000  02850000 *02923000* 06113000  06287800
        06315000  06321500  06330600  07384000 *07670500* 07675000  08017800  08035000  08994040  16384900
C  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 15 AT 02238180
        02238200
C  --  REAL  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 03047000
C  --  REAL  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 04203000
```

```
        04205000  04209000  04211000 *04216000* 04218000
C  --  BOOLEAN  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 62 AT 06283000
        06283400
C  --  REAL  --  DECLARED IN SEGMENT 80 AT 07609200
        *07609500* 07609700
C  --  REAL  --  DECLARED IN SEGMENT 82 AT 07646760
        *07646780* 07646800
C  --  REAL  --  DECLARED IN SEGMENT 86 AT 07802000
        *07820000**07821200**07823000**07825000* 07911000
C  --  BOOLEAN  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 89 AT 08020000
        08021000
C  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 130 AT 12154000
        12157000
CALL  --  PROCEDURE  --  DECLARED IN SEGMENT 89 AT 08072000
        08079000  08098000  08108000  08130000  08217000
CALLPRINTI  --  LABEL  --  DECLARED IN SEGMENT 125 AT 10785000  --  OCCURS AT 10888000
        10834000  10869000  10874000
CALLSTATEMENT  --  LABEL  --  DECLARED IN SEGMENT 143 AT 14017000  --  OCCURS AT 14505000
        14132000
CALLSWITCH  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 05315000
        05316000  07526000  13263000
CALL5  --  DEFINE  --  DECLARED IN SEGMENT 131 AT 13039400
        13153000  13157040  13159000
CARD  --  FILE  --  DECLARED IN SEGMENT 3 AT 01557000
        02213750  02215500  02216250  02220750  09407200
CARD  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 02188000
        02188500
CARDCALL  --  BOOLEAN  --  DECLARED IN SEGMENT 3 AT 01561206
        02013536 *02229000**02230750**02908000*
CARDCOUNT  --  INTEGER  --  DECLARED IN SEGMENT 2 AT 00504150
        *02234500* 09419000
CARDLAST  --  LABEL  --  DECLARED IN SEGMENT 13 AT 02210000  --  OCCURS AT 02216000
        02210750
CARDNUMBER  --  INTEGER  --  DECLARED IN SEGMENT 2 AT 00504100
        *02214260**02234800* 02236750  02882200  02910300  07422100  07516100  07619100  10891100  13310580  14049300
        14576100  15011000  15085100  15271000  15327000  15359000  15374000
CARDONLY  --  LABEL  --  DECLARED IN SEGMENT 13 AT 02210000  --  OCCURS AT 02215250
        02210750
CARDOPTION  --  LABEL  --  DECLARED IN SEGMENT 21 AT 02363000  --  OCCURS AT 02429000
        02436000
CASESTMT  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 07646100  --  FORWARD AT 03083000
        07646840  07825500
CASTA  --  FILE  --  DECLARED IN SEGMENT 3 AT 01561020
        01561050
CASTB  --  FILE  --  DECLARED IN SEGMENT 3 AT 01561030
        01561050
CASTC  --  FILE  --  DECLARED IN SEGMENT 3 AT 01561040
        01561050
CBUFF  --  REAL ARRAY  --  DECLARED IN SEGMENT 3 AT 01561056
        02013617  02013621  02213750  02214000  02215500  02215750  02216250  02216500  02220750  02221000  07025010
        07029000
CDC  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01638000
        05186000  07116000  07249000  08981000  10854000  15227000  15356000
CELL  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 54 AT 05325040
        05325050  05325130  05325150  05325180 *05325220* 05325330  05325350  05325360  05325380  05325390  05325400
        05325410  05325420
CERR  --  LABEL  --  DECLARED IN SEGMENT 70 AT 07067000  --  OCCURS AT 07218000
        07118000  07127000  07165000  07175000  07226000  07233000
```

```
CF  --  DEFINE  --  DECLARED IN SEGMENT 167 AT 17002007
        17004710  17025100  17025300  17080000
CHANGESEQ  --  STREAM PROCEDURE  --  DECLARED IN SEGMENT 3 AT 01741200
        02193250
CHAR  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01299000
        14387000
CHAR  --  REAL  --  NAME PARAMETER  --  DECLARED IN SEGMENT 120 AT 10234000
        10242000  10252000  10255000
CHARCOUNT  --  OWN INTEGER  --  DECLARED IN SEGMENT 120 AT 10230000
        10242000  10250000  10251000 *10256000**10261000* 10285000
CHARI  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01578000
        07421000  07515000  07618000  10890000  15010000  15324000
CHECK  --  BOOLEAN PROCEDURE  --  DECLARED IN SEGMENT 3 AT 05287000
       *05294000* 07082000  07084000  08277000  08395000  08417010  08418700  08427000  08445000  08452000  08465000
        08475000  08489580  08493650  08503000  08516000  08520000  08593000  08613600  08614000  08633000  08639000
        08660000  08673000  08683000  08696550  08698500  08722000  08733000  08749000  08785000  08796000  08818000
        08842000  08853000  08905000  08924000  08966000  10570000  10600000  10630000  10668000  10674000  10792000
        10797000  10828000  10855000  10876000  10902000  10908000  10923000  10935000
CHECK  --  LABEL  --  DECLARED IN SEGMENT 84 AT 07659000  --  OCCURS AT 07680000
        07667500
CHECKBIT  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01000920
        02225750  02496000  09416001
CHECKBOUNDLVL  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 13819100  --  FORWARD AT 01000830
        06100100  06103100  06106100  06243100  06246100  06249100
CHECKDISJOINT  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 04609000
        04617000  13054000  13110000  13196410  13347520
CHECKER  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 05113000
        05120000  05139000  05168000  05187560  05216000  06062150  06283600  06395000  07099500  07215500  07398000
        07468250  07923000  08021100  08054000  08484500  08688500  10308100  10568000  10795000  12044000  12064400
        15085000
CHECKPRESENCE  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 05184000
        05187000  07115000  07469000
CHECKTOG  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01001210
        02225750  09416001
CHKACTIONLABELS  --  LABEL  --  DECLARED IN SEGMENT 92 AT 08380000  --  OCCURS AT 08491000
        08425000  08449000  08469000  08483000
CHKCOMMA  --  LABEL  --  DECLARED IN SEGMENT 124 AT 10560000  --  OCCURS AT 10666000
        10577000  10589000  10640000  10652000  10662000
CHKRTPAREN  --  LABEL  --  DECLARED IN SEGMENT 97 AT 08591100  --  OCCURS AT 08637000
        08613700
CHKSECOND  --  LABEL  --  DECLARED IN SEGMENT 97 AT 08573000  --  OCCURS AT 08643000
        08610000
CHKSOB  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 13209000
        14289000  14296000
CHS  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01639000
        06021000  06283800  06329600  06508000  06512100  07366000  08189000  08219000  08415020  08421000  10359000
        13489000  13767300  14569200  15273300
CHUNK  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01556100
        01556200  02927390  02927400  09418000  09419000
CIV  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01238000
CLASS  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01148000
        02702000  02705100  02761500  02776200  02776600  02850000  02851000  02886000  02910200  02910400  02920000
        02924000  02966500  02968000  04509000  05381000  05390000  05396000  05398000  05403000  07087000  07292000
        08057000  08732100  08795100  08860600  08860650  08860700  08941000  12125000  13201000  13226000  13310075
        13816000  14062000  14091000  14094000  14116300  14259090  14425000  14519000  15019000  15086000  15099000
        15102000  15307000  15316000  15333000  16211100  16212200  17095000
CLCR  --  INTEGER  --  DECLARED IN SEGMENT 3 AT 01330000
```

```
      *02216500**02217750* 02218000  02218250 *02221000* 02224500  02224750
CLCR  -- REAL  -- NAME PARAMETER  -- DECLARED IN SEGMENT 13 AT 02202750
      02202500  02203500  02209000
CLOSESTMT  -- PROCEDURE  -- DECLARED IN SEGMENT 3 AT 08756000  --  FORWARD AT 03064000
      07743000  08824000
CLOSEV  -- DEFINE  -- DECLARED IN SEGMENT 3 AT 01219000
CLSMPMN  -- PROCEDURE  -- DECLARED IN SEGMENT 3 AT 15001000
      15013000  15101000
CLSS  -- INTEGER  -- VALUE PARAMETER  -- DECLARED IN SEGMENT 89 AT 08017500
      08017600
CMPD  -- REAL STREAM PROCEDURE  -- DECLARED IN SEGMENT 3 AT 01561620
      *01561700* 01561710  02224040  02238690
CMPLXSTMT  -- PROCEDURE  -- DECLARED IN SEGMENT 3 AT 12060000  --  FORWARD AT 07777777
      07810100  12065400
CNT  -- STREAM VARIABLE  -- VALUE PARAMETER  -- DECLARED IN SEGMENT 3 AT 02001838
      02001844
CNT  -- STREAM VARIABLE  -- VALUE PARAMETER  -- DECLARED IN SEGMENT 3 AT 02001858
      02001866  02001870  02001892
COC  -- DEFINE  -- DECLARED IN SEGMENT 3 AT 01640000
      07634000  10850000  15095000  15306000  15344000  15355000
CODE  -- FILE  -- DECLARED IN SEGMENT 3 AT 01556200
      01828500  02927530  02927540  02927570  02927610  02927620  02927700  02927740  07147000  09407100
CODE  -- STREAM VARIABLE  -- VALUE PARAMETER  -- DECLARED IN SEGMENT 3 AT 04130000
      04134000
CODE  -- REAL  -- VALUE PARAMETER  -- DECLARED IN SEGMENT 119 AT 10088000
      10086000  10087000  10097000
CODE  -- INTEGER  -- DECLARED IN SEGMENT 119 AT 10106000
     *10127200**10129000**10131000**10137000* 10141000 *10145000**10150000* 10150100  10150225  10151000 *10153000*
      10154000 *10155000**10155500* 10156000 *10157000* 10161000  10163000 *10164000* 10165500 *10165505* 10166000
     *10167000* 10167100 *10167200* 10168000 *10169000* 10171000
CODEFILE  -- DEFINE  -- DECLARED IN SEGMENT 3 AT 01001463
      02927420  02927640  09407020
CODEFILEBIT  -- DEFINE  -- DECLARED IN SEGMENT 3 AT 01001172
      02574200  02927420  02927640  09407020
COL  -- INTEGER  -- DECLARED IN SEGMENT 55 AT 05344000
     *05353000**05354000**05359000**05365000* 05367000
COLON  -- DEFINE  -- DECLARED IN SEGMENT 3 AT 01260000
      05276000  06315500  06320000  06322000  06329100  07597000  08243000  08244000  08698200  08698300  10923000
      13088090  13480000  13503000  14269480  16160000
COLON  -- LABEL  -- DECLARED IN SEGMENT 27 AT 02637000  --  OCCURS AT 02663000
      02641000
COM  -- DEFINE  -- DECLARED IN SEGMENT 3 AT 01641000
      06072280  06073000  06073350  06073600  06073700  06073830  07463500  07465500  07466750  07691500  07693000
      07806000  07817000  07840000  07856000  07911000  09307000  14488500  14551000
COMCOUNT  -- STREAM VARIABLE  -- VALUE PARAMETER  -- DECLARED IN SEGMENT 10 AT 02089500
      02090500  02111500  02118000
COMMA  -- DEFINE  -- DECLARED IN SEGMENT 3 AT 01261000
      05385000  06062135  06062165  06062190  06062210  06072050  06072220  06072240  06073810  06510000  07095000
      07111255  07186000  07205000  07237000  07468500  07470000  07680500  07833000  07905000  07938000  08145000
      08150000  08153000  08412040  08413130  08416000  08418200  08427000  08442000  08452000  08475000  08516000
      08606000  08613300  08620010  08622010  08628000  08639000  08660000  08683000  08733000  08796000  08860600
      08919000  08994000  08995000  10263700  10282000  10297000  10304000  10319000  10358000  10592000  10628000
      10666000  10674000  10855000  10863000  10892000  10926520  10931000  10970000  10993000  12010000  12020000
      12033000  12052000  12056000  12060600  12062400  12064800  12065000  12125000  13071000  13145000  13149000
      13162000  13188000  13196590  13351000  13513000  13547000  13564000  13569000  13918000  14182000  14197000
      14259110  14268000  14269840  14269920  14356000  14404000  15254500  15287000  15293000  15369400
COMMACHECK  -- BOOLEAN PROCEDURE  -- DECLARED IN SEGMENT 3 AT 05384000
     *05385000* 05387000  08912000  08915000  08917000  08972000  08983000  08987000  08990000  08993000
```

```
COMMANTS  --  STREAM LABEL  --  DECLARED IN SEGMENT 10 AT 02092000  --  OCCURS AT 02105000
        02097500
COMMENTS  --  STREAM LABEL  --  DECLARED IN SEGMENT 10 AT 02092000  --  OCCURS AT 02106500
        02107500  02108000
COMMENTV  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01253000
        02886000  02920000  08017900  08034000
COMMON  --  LABEL  --  DECLARED IN SEGMENT 70 AT 07066000  --  OCCURS AT 07234000
        07122000  07221000
COMPAR  --  LABEL  --  DECLARED IN SEGMENT 13 AT 02210500  --  OCCURS AT 02224250
        02216750  02218500  02219500  02220500  02222750  02223500  02224010
COMPARE  --  REAL STREAM PROCEDURE  --  DECLARED IN SEGMENT 3 AT 02188000
       *02190000* 02190250  02203500  02226000  02232000
COMPARECODE  --  DEFINE  --  DECLARED IN SEGMENT 162 AT 16192000
        16209000  16214000
COMPLETE  --  LABEL  --  DECLARED IN SEGMENT 27 AT 02637000  --  OCCURS AT 02909000
        02658000  02667000  02680000  02705200  02715000  02757000  02777100  02782000  02852000  02870000  02877000
        02877020  02893000  02894000
COMPOST  --  LABEL  --  DECLARED IN SEGMENT 27 AT 02639000  --  OCCURS AT 02882000
        02880000
COMPOUNDTAIL  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 07006000
        07033000  07756000  14412000  14512000  14524000  16489000
COMPS1  --  BOOLEAN STREAM PROCEDURE  --  DECLARED IN SEGMENT 170 AT 17029000
       *17034000* 17042300
COMP1  --  BOOLEAN PROCEDURE  --  DECLARED IN SEGMENT 170 AT 17042100
       *17042300**17042350* 17045000
COMP2  --  BOOLEAN PROCEDURE  --  DECLARED IN SEGMENT 172 AT 17115000
       *17117000* 17119000
CONSTANA  --  BOOLEAN  --  DECLARED IN SEGMENT 62 AT 06282600
        06285800 *06286600* 06287200  06287800  06288000
CONSTANA  --  OWN BOOLEAN  --  DECLARED IN SEGMENT 89 AT 08013000
        08074000  08099000  08181000  08188000 *08211000**08224000**08227000*
CONSTANB  --  OWN BOOLEAN  --  DECLARED IN SEGMENT 89 AT 08013000
        08043000  08112000  08116000 *08121000* 08183000  08194000 *08205000* 08213000
CONSTANC  --  OWN BOOLEAN  --  DECLARED IN SEGMENT 89 AT 08013000
        08185000  08195000 *08204000* 08218000
CONSTANTCLEAN  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 04163000  --  FORWARD AT 03034000
        04217000  07111315  07138000  07312000  07459000  07496000  07510000  07585000  07609600  07646790  07727075
        08170000  08190000  10329000  10363000  10986000  13749000  13770000  14047000  14276500  14559000  14572000
CONV  --  REAL STREAM PROCEDURE  --  DECLARED IN SEGMENT 3 AT 02001831
        02001833  02001834  02013360  02013410  02214260  02234900  02251000  02255000  02259000  02502000  02581000
        02592000
CONVERT  --  REAL PROCEDURE  --  DECLARED IN SEGMENT 3 AT 02248000
       *02260000* 02758000  02796000  02803000  02823000  02944000
COP  --  REAL ARRAY  --  DECLARED IN SEGMENT 3 AT 01371000
       *02550000* 04279200  04280000
COPYEOF  --  LABEL  --  DECLARED IN SEGMENT 13 AT 02210600  --  OCCURS AT 02224070
        02224030  02224040
COPYLIB  --  LABEL  --  DECLARED IN SEGMENT 13 AT 02210600  --  OCCURS AT 02224020
        02211000  02224120
CORESZ  --  REAL  --  DECLARED IN SEGMENT 3 AT 01597100
       *08994030* 08994040 *08994500* 09361182
CORNER  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 02011000
        02012000
COUNT  --  INTEGER  --  DECLARED IN SEGMENT 3 AT 01313000
       *02013195**02013200**02013290**02013330**02013335**02013365**02013390**02013420* 02128500  02129000 *02238340*
       *02238385**02238390**02238420**02238440**02238500* 02251000  02252000 *02327000* 02366200 *02371000* 02373000
       *02499000**02590000**02602700**02647000**02691000* 02695000  02697500 *02698000**02699000* 02703000  02705000
```

```
          02705200 *02764000* 02767000 *02770000* 02772000  02774000 *02776100* 02777050  02790000  02796000  02798000
          02803000  02810000  02817000  02873000  02880000  02881000 *02889000**02934000* 02943000 *02948000**02958500*
          07672000  07673500  07676000  07677000  07677500 *10140000**10167300**12165000* 13302000  13306000 *13911500*
          14259010  14507050 *14507100* 14507105 *14507160**16384800* 16384900  16392000
COUNT  -- STREAM VARIABLE -- VALUE PARAMETER -- DECLARED IN SEGMENT 3 AT 02062500
          02064000
COUNT  -- STREAM VARIABLE -- NAME PARAMETER -- DECLARED IN SEGMENT 10 AT 02090000
          02121500
COUNT  -- STREAM VARIABLE -- VALUE PARAMETER -- DECLARED IN SEGMENT 45 AT 05016000
          05017000  05028000
COUNT  -- STREAM VARIABLE -- DECLARED IN SEGMENT 3 AT 05238000
          *05240000*
COUNT  -- STREAM VARIABLE -- DECLARED IN SEGMENT 54 AT 05325070
          *05325150* 05325160 *05325210* 05325320  05325330
COUNT  -- STREAM VARIABLE -- DECLARED IN SEGMENT 104 AT 09012000
          *09016000* 09018000  09019000
COUNT  -- STREAM VARIABLE -- VALUE PARAMETER -- DECLARED IN SEGMENT 121 AT 10236000
          10237000  10240000
COUNT  -- INTEGER -- DECLARED IN SEGMENT 121 AT 10241000
          *10242000* 10245000 *10247000* 10249000  10252000  10255000  10256000
COUNTV -- STREAM VARIABLE -- VALUE PARAMETER -- DECLARED IN SEGMENT 10 AT 02089500
          02090500  02121000  02123500
CPLUS1 -- DEFINE -- DECLARED IN SEGMENT 3 AT 01713000
          15033000
CPLUS2 -- DEFINE -- DECLARED IN SEGMENT 3 AT 01714000
          15032000
CREL -- BOOLEAN -- DECLARED IN SEGMENT 39 AT 04167000
          *04173000**04183000* 04187000
CRF -- DEFINE -- DECLARED IN SEGMENT 157 AT 16010000
          16121000  16200000  16212600  16370000  16408000
CROSSHATCH -- DEFINE -- DECLARED IN SEGMENT 3 AT 01262000
          10279000
CROSSHATCH -- LABEL -- DECLARED IN SEGMENT 27 AT 02640000 -- OCCURS AT 02714000
          02642000
CROSSREFDUMP -- PROCEDURE -- DECLARED IN SEGMENT 3 AT 02001740
          02001830  13283500  14445100
CROSSREFIT -- PROCEDURE -- DECLARED IN SEGMENT 3 AT 02001640
          02001730  02882200  13310580  14469100  15116000  15301100  15342300
CSZ -- LABEL -- DECLARED IN SEGMENT 138 AT 13452000 -- OCCURS AT 13527000
          13518000
CT -- STREAM VARIABLE -- VALUE PARAMETER -- DECLARED IN SEGMENT 2 AT 00512100
          00512300
CTR -- INTEGER -- DECLARED IN SEGMENT 36 AT 04046000
          *04059000**04063000**04067000*
CTR -- STREAM VARIABLE -- VALUE PARAMETER -- DECLARED IN SEGMENT 119 AT 10100000
          10101000  10103000
CURRENT -- INTEGER -- DECLARED IN SEGMENT 131 AT 13039550
          *13113000**13114000* 13184000
D  -- INTEGER -- VALUE PARAMETER -- DECLARED IN SEGMENT 3 AT 01717950
D  -- STREAM VARIABLE -- NAME PARAMETER -- DECLARED IN SEGMENT 7 AT 02001760
          02001775  02001780
D  -- STREAM VARIABLE -- NAME PARAMETER -- DECLARED IN SEGMENT 3 AT 02001838
          02001842
D  -- STREAM VARIABLE -- NAME PARAMETER -- DECLARED IN SEGMENT 3 AT 02001858
          02001890
D  -- STREAM VARIABLE -- NAME PARAMETER -- DECLARED IN SEGMENT 15 AT 02238180
          02238200
D  -- STREAM VARIABLE -- NAME PARAMETER -- DECLARED IN SEGMENT 17 AT 02264400
```

```
DECLRFF  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01007490
        13310525  13310580  14310500  17092900
DECLSW  --  SWITCH LABEL  --  DECLARED IN SEGMENT 143 AT 14018000
        14134000
DEFINEARRAY  --  ALPHA ARRAY  --  DECLARED IN SEGMENT 3 AT 01491000
       *02202030**02202040* 02202050  02202060  02212500 *02217250**02217500* 02217750  02218000  02366100  02602600
        02719000  02721000 *02900000**02902000* 02903000
DEFINECTR  --  INTEGER  --  DECLARED IN SEGMENT 3 AT 01481000
        02715000  02760000 *10262000**10278000* 10280000 *10283000**10284000*
DEFINEDEC  --  LABEL  --  DECLARED IN SEGMENT 143 AT 14017000  --  OCCURS AT 14254000
        14021000
DEFINEDID  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01180000
        02894000  13310075  13339000  14259010  14259090
DEFINEGEN  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 10228000
        10286000  12124000  14266000
DEFINEINDEX  --  INTEGER  --  DECLARED IN SEGMENT 3 AT 01481000
        02212500  02718000  02719000 *02721000* 02898000  02900000  02902000  02903000 *02905000*
DEFINEPARAM  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 12150000  --  FORWARD AT 01717950
        10264810  12166000  14259085
DEFINEV  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01298000
        10278000
DEFINFO  --  ALPHA ARRAY  --  DECLARED IN SEGMENT 3 AT 01481100
        10264760  14259080
DEFINING  --  BOOLEAN  --  DECLARED IN SEGMENT 3 AT 01007650
       *02419110* 02910100  13283500  13310900 *14255500**14268500* 14445100  17001000
DEFSTACKHEAD  --  INTEGER  --  DECLARED IN SEGMENT 3 AT 01481300
        02723500  02724000 *02726000* 02873000  12122000 *12123000*
DEL  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01642000
        04066000  04079000  05186000  06072090  06072280  06073750  06073800  06073850  06335500  07463750  07465500
        07466750  07467000  07471250  07641000  07691500  07693000  07807000  07817000  07840000  07856000  07857000
        07912000  07951000  12041140  12061800
DELAY  --  LABEL  --  DECLARED IN SEGMENT 57 AT 06060100  --  OCCURS AT 06072200
        06061100
DESC  --  DEFINE  --  DECLARED IN SEGMENT 89 AT 08016000
        08098000  08108000
DESCRP  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 02013060
        02013065  02013078
DEST  --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 2 AT 00515000
        00519000
DEST  --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 3 AT 01737000
        01737100
DEST  --  STREAM VARIABLE  --  DECLARED IN SEGMENT 104 AT 09012000
       *09017000* 09019000
DEXP  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 06388000  --  FORWARD AT 03017000
        06253000  06305000  06408000  07374000  07528000  07925100  08243000  08245000  08493650  08698200  08698400
        10992000
DIA  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01680000
        04315000  04550000
DIALA  --  INTEGER  --  DECLARED IN SEGMENT 3 AT 01502000
       *04085000**04123000**04169000* 04314000 *04315000**04319000**07484000**07492000**07579000**08100000**08101000*
       *08152000**10293000**10364000*
DIALB  --  INTEGER  --  DECLARED IN SEGMENT 3 AT 01502000
       *04085000**04123000**04169000* 04316000 *04317000**04319000**07484000**07492000**07579000**08100000**08101000*
       *08152000**10293000**10364000*
DIALER  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01579200
        06334500  06335200
DIB  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01681000
        04317000
```

```
DIMCTR  --  DEFINE  --  DECLARED IN SEGMENT 125 AT 10731000
        10808000  10823000  10848000
DINFO  --  REAL  --  DECLARED IN SEGMENT 120 AT 10259200
       *10259600* 10264810
DINFO  --  INTEGER  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 12151000
        12150000  12151000  12162000  12163000
DINFO  --  REAL  --  DECLARED IN SEGMENT 147 AT 14254100
       *14259017* 14259085  14259150  14265930  14265950  14266000
DIQ  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01237000
DIR  --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 3 AT 02001020
        02001090
DIRECTORY  --  REAL ARRAY  --  DECLARED IN SEGMENT 3 AT 01561240
        02013235  02013250 *02013255* 02013275  02013295 *02013505**02013568**02013570* 02013665
DIVIDE  --  INTEGER PROCEDURE  --  DECLARED IN SEGMENT 3 AT 10184000  --  FORWARD AT 03032000
        10153000  10160000  10161500  10167900 *10192000*
DK  --  DEFINE  --  DECLARED IN SEGMENT 2 AT 00514500
        00519000  00521000
DMUP  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 10681000
        10939000  14155000
DOITANYWAY  --  LABEL  --  DECLARED IN SEGMENT 149 AT 14273000  --  OCCURS AT 14318000
        14308000
DOLLAR  --  BOOLEAN  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 02013165
        02013185
DOLLAR  --  LABEL  --  DECLARED IN SEGMENT 27 AT 02639000  --  OCCURS AT 02729000
        02641000
DOLLARCARD  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 02319000  --  FORWARD AT 02065500
        02229000  02231000  02604000  02730000
DOLLARTOG  --  BOOLEAN  --  DECLARED IN SEGMENT 3 AT 01001540
       *02366000**02603000*
DOLLAR2TOG  --  BOOLEAN  --  DECLARED IN SEGMENT 13 AT 02211250
       *02230100* 02234600 *02234650*
DONEMORE  --  LABEL  --  DECLARED IN SEGMENT 95 AT 08493510  --  OCCURS AT 08493710
        08493550
DONESOME  --  LABEL  --  DECLARED IN SEGMENT 95 AT 08493510  --  OCCURS AT 08493600
        08493520
DONTSCAN  --  LABEL  --  DECLARED IN SEGMENT 15 AT 02238120  --  OCCURS AT 02238342
        02238450
DOSTMT  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 07482000
        07488000  07749000
DOT  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01841000
        06127000  06266000  06277000
DOT  --  LABEL  --  DECLARED IN SEGMENT 27 AT 02637000  --  OCCURS AT 02668000
        02641000
DOTIT  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 06338000  --  FORWARD AT 03015000
        06127000  06266000  06277000  06346000
DOTSYNTAX  --  BOOLEAN PROCEDURE  --  DECLARED IN SEGMENT 3 AT 05270000
       *05286000* 06341000  15112000  15341000
DOUBLEV  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01222000
DOV  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01225000
        07494000  08159000  08186000
DPPOS  --  STREAM VARIABLE  --  DECLARED IN SEGMENT 3 AT 02001070
       *02001090* 02001270 *02001290**02001490*
DPTOG  --  BOOLEAN  --  DECLARED IN SEGMENT 3 AT 01690000
        02253000  02784000  02794000  02829000 *12012000**12016000*
DSK  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 2 AT 00515000
        00516000  00519000  00521000
DSK1  --  FILE  --  DECLARED IN SEGMENT 3 AT 01561085
```

```
        02001821  17010000  17013000  17022100  17026000  17044000  17093950
DSK2  --  FILE  --  DECLARED IN SEGMENT 3 AT 01561087
        02001680  17004000  17076000  17118000
DSS  --  PROCEDURE  --  DECLARED IN SEGMENT 157 AT 16362000
        16398000  16491000
DSV  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01240000
DTYPE  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01452000
        06253000  06263000  06305000  06407000  10327000
DUMMY  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01001470
DUMMY  --  INTEGER  --  DECLARED IN SEGMENT 139 AT 13637100
DUMPDEC  --  LABEL  --  DECLARED IN SEGMENT 143 AT 14014000  --  OCCURS AT 14149000
        14019000
DUMPE  --  DEFINE  --  DECLARED IN SEGMENT 125 AT 10750000
        10883000  10916000
DUMPEE  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01547000
        07622000  10881000  10914000
DUMPETEMP  --  DEFINE  --  DECLARED IN SEGMENT 125 AT 10727000
        10881000  10883000  10914000  10915000  10916000  10927000
DUMPINFO  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 02264000
        02266100  02468000
DUMPLOC  --  DEFINE  --  DECLARED IN SEGMENT 125 AT 10777000
        10800000  10929000
DUMPOR  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01552000
        07627000
DUMPR  --  DEFINE  --  DECLARED IN SEGMENT 125 AT 10771000
        10930000
DUMPV  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01287000
        14130000  04062000  04063000  05186000  06069000  06072060  06329500  07468500  07469750  07470500  07859060
        08728500  08977000  10812000  10961000  13509000  15095000  15306000
DWA  --  INTEGER  --  DECLARED IN SEGMENT 17 AT 02264100
        *02265500*  02265600  02265700
DYNAM  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01154100
        02723500  02901000
D1  --  REAL  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 119 AT 10088000
        10086000  10087000  10095000
D1  --  INTEGER  --  DECLARED IN SEGMENT 119 AT 10106000
        *10157000*  10160000  10167900  10171000
D2  --  REAL  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 119 AT 10088000
        10086000  10087000  10096000
D2  --  INTEGER  --  DECLARED IN SEGMENT 119 AT 10106000
        10160000  10167900  10171000
E  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 13293000  --  FORWARD AT 03054000
        10576000  10582000  10656000  13349000  13917000  14259096  14507140
E  --  LABEL  --  DECLARED IN SEGMENT 36 AT 04047000  --  OCCURS AT 04074000
        04048000  04057000
E  --  LABEL  --  DECLARED IN SEGMENT 37 AT 04099000  --  OCCURS AT 04112000
        04100000  04102000
E  --  REAL  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 04500000
        04526200  04530000  04536000  04543000  04549000  04555000
E  --  REAL  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 43 AT 04504000
        04503000  04506000  04508000 *04509000*  04522000
E  --  LABEL  --  DECLARED IN SEGMENT 86 AT 07801000  --  OCCURS AT 07906000
        07814500  07816000  07832000  07833000  07835000  07837000  07847000
E  --  REAL  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 157 AT 16065000
        16070000  16072000  16074000
E  --  REAL  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 157 AT 16085000
        16089000  16093000
EDITLINE  --  STREAM PROCEDURE  --  DECLARED IN SEGMENT 3 AT 02183500
```

```
        02187000  02195000  02195750  02214200  02227500  02228750  05038000  07025020
EDOC  -- REAL ARRAY -- DECLARED IN SEGMENT 3 AT 01371000
        04147000  04296000  05312000  07646630  07646690  07694000  07697000  07698500  08860150  08860450  09314000
        09395000  09396000  10071000  10127100  13640000  13777600  13818500  14049000  14595000
EDOC  -- STREAM VARIABLE -- NAME PARAMETER -- DECLARED IN SEGMENT 3 AT 01688010
        01688030  01688050
EDOCINDEX -- DEFINE -- DECLARED IN SEGMENT 3 AT 01710000
EEXIT -- LABEL -- DECLARED IN SEGMENT 15 AT 02238120 -- OCCURS AT 02238770
        02238280  02238430  02238520  02238522  02238552  02238602  02238652  02238680
EL  -- LABEL -- DECLARED IN SEGMENT 119 AT 10085000 -- OCCURS AT 10119000
        10124100
EL  -- REAL -- DECLARED IN SEGMENT 129 AT 12102000
       *12112000* 12114000 *12125000* 12126000
ELBAT -- REAL ARRAY -- DECLARED IN SEGMENT 3 AT 01319000
        02265100  02265250 *02910000* 02916000  02920000  02923000 *02924000* 05139000  05140000  05145000  05168000
        05187560  05187570  05189000  05216000  05217000  05218000  05222000  05273300  05273400  05274100  05274200
        05280000  05281000  06016000  06044000  06062000  06062150  06062350  06062380  06110000  06117000  06120100
        06184000  06232000  06256000  06284400  06286200  06287000  06287800  06313650  06313700  06314150  06314200
        06395000 *07031000* 07079000  07396000  07430000  07442000  07463500  07468250  07508000  07509000  07511000
        07554000  07566000  07572000  07583000  07599000 *07646515* 07663500  07670500  07923000 *08017900* 08029000
       *08034000* 08037000  08176000  08484500  08488000  08489550  08688500  08692000  08696520 *08732100* 08739000
       *08795100* 08809000 *08860600**08860650**08860700* 08909000  08910000  08914000  08915000  08916000  08917000
        08969000  08970000  08975000  08976000  08988000  08989000  08991000  08992000  10277000  10308100  10309000
        10568000  10569000  10573000  10585000  10598000  10644000  10649000  10656010  10658000  10795000  10796000
        10804000  10911000  10929000  10930000  10965000  10967000  12015000  12024000  12036000  12036200  12044000
        12061000  12061725  12062600  12062800  12064400  12125000  13086000  13200000  13310350  13348200  13361000
        13372000  13476000  13483000  13517000  13575000 *13816000* 13819400  13911400 *14062000* 14129000  14269446
        14269448  14269520  14269540  14314000  14349000  14352000  14353000  14399000 *14507090**14519000* 15077000
        16121000  16124000  16159000  16200000  16204000  16209000  16211100  16211300  16212200  16212500  16212600
        16212700  16214000  16254000  16283000  16321000  16329000  16342000  16344000  16370000  16376000  16379500
        16408000  16412000  16435000 *16449000*
ELBATCLASS -- REAL -- VALUE PARAMETER -- DECLARED IN SEGMENT 3 AT 05289000
        05287000  05288000  05294000
ELBATWORD -- REAL -- VALUE PARAMETER -- DECLARED IN SEGMENT 3 AT 03016000
ELBATWORD -- REAL -- VALUE PARAMETER -- DECLARED IN SEGMENT 3 AT 05113000
        05116000  05118000
ELBATWORD -- DEFINE -- DECLARED IN SEGMENT 49 AT 05160000
        05168000  05170000  05175000
ELBATWORD -- REAL -- VALUE PARAMETER -- DECLARED IN SEGMENT 3 AT 05207000
        05206000  05207000  05208000  05209000
ELBATWORD -- REAL -- VALUE PARAMETER -- DECLARED IN SEGMENT 3 AT 06356000
        06359000
ELBATWORD -- DEFINE -- DECLARED IN SEGMENT 79 AT 07595000
        07599000  07601000  07611000  07617000
ELBATWORD -- DEFINE -- DECLARED IN SEGMENT 124 AT 10519000
        10585000  10644000  10651000  10652000  10658000  10659000
ELBATWORD -- DEFINE -- DECLARED IN SEGMENT 125 AT 10781000
        10803000  10814000  10826000  10835000  10839000  10868000  10874000  10883000  10885000  10886000  10888000
        10889000  10911000  10912000  10912100  10916000
ELBATWORD -- REAL -- VALUE PARAMETER -- DECLARED IN SEGMENT 3 AT 15003000
        15001000  15002000  15008000  15009000  15010010  15010020
ELBATWORD -- REAL -- VALUE PARAMETER -- DECLARED IN SEGMENT 3 AT 15016000
        15014000  15015000  15019000
ELBATWORD -- REAL -- VALUE PARAMETER -- DECLARED IN SEGMENT 3 AT 15022000
        15020000  15021000  15034000
ELBW  -- REAL -- VALUE PARAMETER -- DECLARED IN SEGMENT 3 AT 05389000
        05390000  05390100  05391000
```

```
ELBW  -- REAL  -- VALUE PARAMETER  -- DECLARED IN SEGMENT 3 AT 05395000
      05396000  05396100  05397000
ELBW  -- REAL  -- VALUE PARAMETER  -- DECLARED IN SEGMENT 3 AT 05402000
      05403000  05403100  05404000
ELBW  -- REAL  -- DECLARED IN SEGMENT 67 AT 06392000
      *06395000* 06400000  06405000  06406000
ELBW  -- REAL  -- DECLARED IN SEGMENT 76 AT 07503000
      *07511000* 07512000  07514000  07515000  07520000  07521000  07526000
ELBW  -- REAL  -- DECLARED IN SEGMENT 87 AT 07922000
      *07923000* 07926000  07926100
ELBW  -- REAL  -- VALUE PARAMETER  -- DECLARED IN SEGMENT 103 AT 08940000
      08941000  08941100  08942000
ELCLASS  -- INTEGER  -- DECLARED IN SEGMENT 3 AT 01328000
      02934000 *02944000**02952000**05002000**05003000* 05109000  05141000  05169000  05187580  05273800  05275000
      05304000  06004000  06013000  06019000  06034000  06036000  06048000  06052000  06055000  06062140  06062165
      06062190  06062210  06062220  06062290  06062295  06062340  06062370  06062390  06062430  06072050  06072110
      06072220  06072240  06072260  06073250  06073450  06073650  06073810  06073860  06091000  06092000  06092005
      06094950  06125000  06127000  06128500  06141000  06144000  06152000  06170000  06175000  06188000  06192000
      06193000  06209000  06225000  06226000  06227000  06227500  06239000  06261000  06266000  06267500  06277000
      06286400  06286600  06286800  06288800  06290000  06302000  06313550  06314000  06314650  06320000  06329100
      06329400  06332300  06393000  06411000  06416000  06422000  06504000  06510000  07012000  07013000  07018000
      07021000  07026000 *07032000* 07111200  07111255  07111275  07112000  07205000  07211000  07237000  07238000
     *07260000**07374000* 07399000  07404000  07409000  07441000  07452000  07463000  07467750  07468500  07470000
      07471750  07485000  07494000  07506000  07507000  07550000  07553000 *07556000* 07578000 *07646515* 07646525
      07646538  07646555  07646595  07662000  07668500  07669000  07669500  07671000  07683500  07684500  07685500
      07688000  07728000  07728500  07729100  07729190 *07733000**07735000* 07814000  07814200  07814300  07831000
      07833000  07834000  07838000  07846000  07849000  07852000  07859080  07905000  07909000  07925100  07938000
      07939000  07943000  08017700  08031000  08032000  08038000  08103000  08113000  08128000  08133000  08139000
      08145000  08150000  08153000  08159000  08229000  08244000  08390000  08401000  08409000  08413120  08416000
      08418100  08418200  08422000  08435000  08479000  08485000  08493580  08493680  08597100  08606000  08613100
      08613300  08622000  08628000  08653100  08653120  08653130  08653180  08653200  08654000  08687000  08689000
      08698000  08698300  08729000  08732100  08738000  08739000  08742000  08795100  08805000  08809000 *08860350*
      08860600  08860650  08908000  08919000  08967000  08973000  08994000  08994040  08994080  08995000  08995500
      08996000  10112000  10112100  10113000  10113100  10114000  10120000  10121000  10124000  10128000  10130000
      10132000  10133000  10134000  10134500  10134530  10134540  10134545  10134550  10134580  10135000 *10141000*
      10146000  10150000  10150130  10150140  10150145  10150155  10150170  10150185  10150190 *10150290* 10159000
     *10159100* 10160000  10161500  10162000  10167900 *10167910**10178000* 10178100 *10179000* 10294000  10295000
      10297000  10298000  10307000  10310000  10311000  10317000  10319000  10320000  10321000  10325000  10358000
      10368000  10565000  10628000  10642000  10863000  10931000  10993000  12022000  12028000  12033000  12034100
      12034110  12035000  12036100  12036400  12039000  12041100  12042000  12052000  12056000  12060800  12061500
      12062008  12062250  12062255  12062400  12062500  12062700  12063000  12063600  12063800  12064200  12064800
      12065000  13051000  13071000  13083000  13088010  13088085  13088090  13088105  13089000  13104000  13145000
      13149000  13162000  13182000  13196380  13196520  13196590  13331000  13337000  13339000  13473000  13503000
     *13511000**13572000* 13582000  13787000 *13791050* 13794000 *13806000* 13810000  13815000  13816000  13918000
      14174000  14182000  14197000  14219000  14238000  14259020  14259120  14260000  14269280  14269442  14269460
      14269680  14269840  14269920  14325000  14326000  14336000  14357000  14370000  14396000  14404000  14479000
     *14516000* 14601100  15078000  15083000  15089010  15111000  15287000  15294000  15339000  15349000  15369400
      16119000  16131000  16132000  16198000  16204000  16210000  16212000  16221000  16222000  16223000  16224000
      16225000  16228500  16230000  16315000  16319000  16320000  16326100  16330000  16335000  16336000  16337000
      16338000  16339000  16341000  16374000  16379000  16381000  16384000  16384100  16384700  16410000  16415000
      16417000  16437000  16477000  16478000  16479000  16481000  16482000
ELCLASS  -- REAL  -- DECLARED IN SEGMENT 120 AT 10258100
     *10263400* 10263700  10264100  10264300  10264410  10277000  10279000 *10282000*
ELCLASS  -- DEFINE  -- DECLARED IN SEGMENT 134 AT 13220000
      13226000  13246000  13253000  13278000
ELSEBRANCH  -- INTEGER  -- DECLARED IN SEGMENT 63 AT 06295000
     *06300000* 06307000
ELSEV  -- DEFINE  -- DECLARED IN SEGMENT 3 AT 01227000
```

```
        06302000  07567000  07578000  16228500  16230000
EMIT  -- PROCEDURE --  DECLARED IN SEGMENT 3 AT 04291000  --  FORWARD AT 03028000
        04004000  04007000  04014000  04019000  04024000  04086000  04221000  04302000  04315000  04317000  04318000
        04531000  04535000  04550000  04551000  04554000  04613000  06054000  06073300  06073400  06073700  06110000
        06154000  06206000  06208000  06256000  06289200  06289600  06308000  06334600  06368000  06508000  06512100
        07365000  07441000  07449000  07544000  07646533  07646543  07804000  07805000  07817000  07925000  08049000
        08050000  08122000  08148000  08151000  08227000  08243000  08244000  08287000  08401050  08597500  08613500
        08698200  08698300  08698800  08698850  08906000  08907000  08908000  08911000  08913000  08921000  08925000
        08954000  08957000  08958000  08964000  08965000  08968000  08979000  08981000  08984000  08993500  08994510
        08995500  10819000  10821000  10844000  10846000  10959000  10974000  10975000  10995000  13767500  13916000
        14459000  15273200  15283000  15285000  16038000  16080000  16122000  16133000  16219000  16227000  16232000
        16266000  16287000  16387000  16394000  16453000
EMITB  -- PROCEDURE --  DECLARED IN SEGMENT 3 AT 04114000  --  FORWARD AT 03036000
        04125000  04218000  06301000  06307000  07111320  07111325  07138000  07201000  07272000  07312000  07487000
        07495000  07496000  07523000  07524000  07540000  07579000  07586000  07608000  07609700  07646730  07646800
        07727080  08047000  08078000  08086000  08127000  08146000  08168000  08215000  08221000  08918000  08923000
        10366000  10928000  10994000  10997000  13606000  13769000  14571000  15037000
EMITC  -- PROCEDURE --  DECLARED IN SEGMENT 3 AT 04010000
        04014000  14451400  14454000  14455000  14456000  16040000  16044000  16045000  16046000  16077000  16079000
        16121000  16124000  16200000  16209000  16211300  16212500  16212600  16212700  16214000  16216000  16286000
        16342000  16352000  16370000  16379500  16383000  16408000  16415000  16417000
EMITCALL  -- LABEL --  DECLARED IN SEGMENT 97 AT 08589000  --  OCCURS AT 08698000
        08612000  08650000  08678000  08688000
EMITD  -- PROCEDURE --  DECLARED IN SEGMENT 3 AT 04311000  --  FORWARD AT 03050000
        04322000  04556000  05203000  06062195  06062560  06070000  06336300  07470750  07859070  08413016  08414016
        08416016  08418500  08418650  08613500  08621004  08624004  08629004  08728600  15100000  15308000
EMITFORMAT  -- PROCEDURE --  DECLARED IN SEGMENT 119 AT 10086000
        10099000  10117000  10122000  10126000  10134590  10150225  10171000
EMITI  -- PROCEDURE --  DECLARED IN SEGMENT 3 AT 04500000
        06343000  15124000  15369000
EMITJUMP  -- PROCEDURE --  DECLARED IN SEGMENT 157 AT 16065000
        16093000  16255000
EMITL  -- PROCEDURE --  DECLARED IN SEGMENT 3 AT 04003000
        04032000  04066000  04070000  04205000  04544000  05145000  05154000  05167000  05168000  05180000  05186000
        05187620  05222000  06062185  06062385  06069000  06073000  06328000  06329550  06331700  06331800  06333500
        06334200  06335100  06369000  06376000  06405000  06512200  07256000  07377000  07380000  07418000  07419000
        07466500  07466750  07529100  07625000  07848000  07859060  07911000  08046000  08067000  08077000  08285000
        08401070  08412080  08413016  08413030  08413060  08413080  08413100  08413110  08414010  08414016  08414022
        08414030  08415010  08416016  08416030  08418350  08418650  08418800  08419000  08424000  08433000  08437000
        08437050  08437075  08449000  08451000  08493540  08493670  08493675  08493780  08493840  08515100  08524000
        08608000  08611000  08612000  08621000  08621004  08621010  08621020  08624000  08624004  08625000  08626000
        08629004  08631000  08650000  08653150  08653190  08653220  08656000  08659000  08698600  08728500  08740000
        08803000  08807000  08811000  08812200  08858000  08860750  08906000  08913000  08923000  08956000  08957000
        08958000  08959000  08965000  08970000  08977000  08979000  08985000  08986000  09272000  09307000  10359000
        10827000  10837000  10868000  10887000  10888000  10906000  10917000  10982000  11001000  12036300  12041130
        12061725  12062900  12064000  13056000  13057000  13058000  13067000  13072000  13082500  13108000  13109000
        13111000  13152000  13153000  13157040  13159000  13166000  13196430  13196440  13196450  13196550  13196600
        13264000  13267000  13347500  13486000  13508000  13517000  13525000  13531100  13545000  13552000  13553000
        13757000  13763000  13766000  13914000  13919000  14488500  14543000  14550000  14564100  14567000  15008000
        15248000  15321000
EMITLNG  -- PROCEDURE --  DECLARED IN SEGMENT 3 AT 04098000
        04112000  06154000  06203000  07573000  07575000  08150000  08220000
EMITN  -- PROCEDURE --  DECLARED IN SEGMENT 3 AT 04022000
        04024000  04212000  04524000  04616000  05143000  05147000  05155000  05175000  05187600  05202000  06062350
        06371000  07111310  07134000  07348000  08075000  08179000  08212000  08413040  08414022  08416022  08621010
        08625000  08630000  08911100  08955000  08957000  08970000  08978000  08980000  08985000  10309000  10328000
        10835000  12036000  12062600  13068000  13160000  13166000  13196560  13347500  13915000  15094000  15224000
```

```
        15271100  15353000  15362000  16261000
EMITNO  -- DEFINE --  DECLARED IN SEGMENT 3 AT 08289010
        08412070  08413100  08414010  08414030  08621000  08621020  08624000  08626000
EMITNUM  -- PROCEDURE --  DECLARED IN SEGMENT 3 AT 04203000  --  FORWARD AT 03049000
        04121000  04225000  06071000  06113000  06283400  07384000  07422100  07516100  07619100  07693000  08021000
        08089000  08493790  10330000  10891100  12014000  12015000  13229000  15011000  15271000  15326000  15359000
        15374000
EMITO  -- PROCEDURE --  DECLARED IN SEGMENT 3 AT 04006000
        04018000  04023000  04033000  04034000  04062000  04063000  04064000  04066000  04067000  04071000  04072000
        04075000  04079000  04109000  04112000  04122000  04311090  04507000  04531000  04537000  04545000  05143000
        05167000  05171000  05177000  05179000  05186000  05187600  05226000  05316000  06021000  06062110  06062170
        06062435  06062560  06064000  06067000  06069000  06071000  06072060  06072080  06072090  06072280  06073000
        06073750  06073800  06073850  06204000  06205000  06283800  06289400  06318500  06328000  06329500  06329550
        06329600  06331700  06335500  06358000  06377000  06503000  06508000  06512100  07108500  07108505  07111230
        07111315  07111320  07116000  07122520  07122530  07138000  07249000  07272000  07311000  07366000  07408000
        07418000  07441000  07463750  07465500  07466500  07466750  07467000  07468500  07469750  07470500  07470750
        07471250  07514000  07525000  07526000  07529100  07529200  07617000  07625000  07634000  07641000  07646460
        07691500  07693000  07806000  07807000  07817000  07840000  07851000  07856000  07857000  07859060  07859070
        07911000  07912000  07944000  07946000  07951000  08044000  08049000  08050000  08065000  08068000  08070000
        08079000  08094000  08123000  08125000  08126000  08166000  08189000  08214000  08219000  08228000  08400000
        08401050  08401070  08413040  08413060  08414022  08415010  08415020  08416022  08418600  08421000  08489510
        08489520  08493570  08513000  08597000  08597500  08597700  08613200  08621010  08625000  08627000  08630000
        08653110  08653125  08653140  08653195  08653210  08694500  08695000  08698100  08728500  08728600  08906000
        08924000  08953000  08956000  08957000  08976000  08977000  08980000  08985000  08993000  09272000  09307000
        10329000  10350000  10359000  10361000  10616000  10802000  10803000  10812000  10836000  10837000  10850000
        10851000  10854000  10906000  10922000  10928000  10959000  10960000  10961000  10971000  10985000  10992000
        10994000  10995000  12024000  12038000  12041130  12041140  12054000  12061100  12061200  12061700  12061800
        12063300  12064000  12064800  13053000  13069000  13082500  13159000  13166000  13196400  13196570  13263000
        13264000  13268000  13470000  13489000  13509000  13526000  13544000  13545000  13565000  13744000  13747000
        13758000  13767300  13904000  14046000  14488500  14544000  14551000  14557500  14558500  14565000  14569200
        15007000  15095000  15106000  15222000  15227000  15230000  15241000  15273300  15291000  15306000  15309000
        15315000  15333000  15344000  15354000
EMITPAIR  -- PROCEDURE --  DECLARED IN SEGMENT 3 AT 04028000
        04034000  04076000  04077000  04523000  04614000  05201000  05227000  06062190  06072000  06072070  06072090
        06072280  06073350  06073600  06073700  06073830  06120200  06288800  06399000  07111310  07134000  07217510
        07225000  07306000  07311000  07358000  07420000  07421000  07463500  07465500  07466750  07514000  07519000
        07618000  07626000  07691500  07693000  07840000  07856000  07926000  07946000  08090000  08401060  08402000
        08413040  08414022  08415030  08416022  08418300  08418400  08489550  08493860  08514000  08597600  08604000
        08613400  08621010  08625000  08627000  08630000  08653110  08653140  08653210  08696520  08860750  08910000
        08915000  08917000  08953000  08954000  08955000  08970000  08974000  08976000  08981000  08989000  08992000
        08993500  10302000  10328000  10360000  10621000  10868000  10888000  10890000  10919000  12061100  12061200
        13230000  13501000  13761000  13767300  13915000  14181000  14233000  14545000  14565000  14569200  15007000
        15008000  15010000  15107000  15267000  15279000  15320000  15322000  15323000
EMITREST  -- LABEL --  DECLARED IN SEGMENT 99 AT 08780000  --  OCCURS AT 08816000
        08803000  08807000  08811000  08812200
EMITSTORE  -- DEFINE --  DECLARED IN SEGMENT 3 AT 13214000
        13230000  13501000  13761000  14181000  14233000  14545000  14565000
EMITTIME  -- DEFINE --  DECLARED IN SEGMENT 3 AT 08289020
        08413040  08414022  08416022  08621010  08625000  08630000
EMITUP  -- PROCEDURE --  DECLARED IN SEGMENT 3 AT 04045000
        04079000  06020000  06050000
EMITV  -- PROCEDURE --  DECLARED IN SEGMENT 3 AT 04017000
        04019000  04078000  04184000  04521000  05180000  05219000  05226000  05316000  06062155  06062285  06062435
        06062540  06062560  06066000  06072090  06283600  06285800  06334500  06335200  06372000  06380000  06400000
        06404000  06513000  07122510  07255000  07270000  07306000  07335000  07346000  07365000  07413000  07419000
        07423000  07445000  07454000  07516200  07529100  07620000  07625000  07646450  07946000  07953000  08021200
        08044000  08075000  08166000  08212000  08287100  08287200  08489500  08493860  08694000  08698600  08698850
        08698900  08926000  08953000  08959000  08986000  08997000  10350000  10820000  10826000  10845000  10850000
```

```
        10854000  10874000  10884000  10891200  10927000  10959000  10960000  12041130  12047000  12061100  12061200
        12061750  12064000  12064500  13059000  13153000  13157040  13159000  13196460  13236000  13554000  13743000
        13756000  13919000  14045000  14542000  14558000  14564100  15007000  15012000  15031000  15097000  15222000
        15230000  15249000  15273200  15284000  15328000  15330000  15344000  15352000  15361000  15374100
EMITWORD  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 04142000
        04152000  04187000  15033000
EMIT21  --  PROCEDURE  --  DECLARED IN SEGMENT 43 AT 04503000
        04526200  04530000  04536000  04543000  04549000  04555000
ENDOFITALL  --  LABEL  --  DECLARED IN SEGMENT 3 AT 05101100  --  OCCURS AT 17000100
        05107100  05107200  13312500
ENDREADTAPE  --  LABEL  --  DECLARED IN SEGMENT 14 AT 02198760  --  OCCURS AT 02202080
        02202010
ENDTOG  --  BOOLEAN  --  DECLARED IN SEGMENT 3 AT 01415000
       *02653000* 02680000  02884000 *06021000**06022000**07016000* 07019000 *07020000**07646585* 07646600 *07646610*
ENDV  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01228000
        02910400  05109000  07013000  07018000  07646538  07646555  07646595  07732000  07733000  16477000
ENIL  --  REAL ARRAY  --  DECLARED IN SEGMENT 2 AT 00504300
       *02236750**09307100* 13654000  14049000 *14049300**14576100* 14594000
ENILPTR  --  INTEGER  --  DECLARED IN SEGMENT 2 AT 00504400
        02236750 *02237000**02237250**09307100* 13654000 *14049200**14049300* 14576100 *14593000*
ENILSPOT  --  DEFINE  --  DECLARED IN SEGMENT 2 AT 00504500
        02236750  14049300  14576100  16133000
ENTER  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 13714000  --  FORWARD AT 03069000
        13081000  13471000  13727000  14138000  14139000  14140000  14141000
ENTERID  --  STREAM PROCEDURE  --  DECLARED IN SEGMENT 131 AT 13020000
        13102000  06061000
ENTRY  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 13314000  --  FORWARD AT 03055000  --  OCCURS AT 03055000
        13049000  13193000  13196360  13567000  13726000  13787000  13806000  14161000  14170000  14174000  14194000
        14217000  14259010  14269240  14301000  14319000  14333000  14403000  14409000
EOF  --  LABEL  --  DECLARED IN SEGMENT 13 AT 02210250  --  OCCURS AT 02217000
        02216250  02220750
EOF  --  LABEL  --  DECLARED IN SEGMENT 15 AT 02238120
EOF  --  LABEL  --  DECLARED IN SEGMENT 171 AT 17009000  --  OCCURS AT 17012000
        17010000
EOF  --  LABEL  --  DECLARED IN SEGMENT 175 AT 17073000  --  OCCURS AT 17082000
        17076000  17079000
EOFT  --  LABEL  --  DECLARED IN SEGMENT 14 AT 02198760  --  OCCURS AT 02202020
        02201750
EOF2  --  LABEL  --  DECLARED IN SEGMENT 177 AT 17091100  --  OCCURS AT 17096800
        17093950
EPART  --  LABEL  --  DECLARED IN SEGMENT 27 AT 02639000  --  OCCURS AT 02810000
        02688000
EQL  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01644000
        10928000
EQLESCHECK  --  BOOLEAN PROCEDURE  --  DECLARED IN SEGMENT 3 AT 05402000
       *05403100**05407000* 08916000  08991000
EQUAL  --  BOOLEAN STREAM PROCEDURE  --  DECLARED IN SEGMENT 3 AT 02062500
       *02064000* 02064500  02881000  10264750
EQUALV  --  DEFINE  --  DECLARED IN SEGMENT 162 AT 16192000
        16214000
EQVOP  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01271000
        02965500  02971000  06175000
ERR  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 05105000  --  FORWARD AT 01718000
        02211500  02768000  04145000  04300000  05109000  05285000  05295000  05385000  05390000  05391000  05392000
        05396000  05397000  05398000  05399000  05403000  05404000  05405000  05406000  06005000  06062565  06072030
        06072110  06072210  06072230  06072250  06072270  06073200  06073820  06073860  06092000  06095000  06119000
        06121000  06125000  06133000  06144000  06157000  06173000  06191000  06194000  06227000  06231000  06234000
```

```
          06261000  06272000  06302000  06314000  06314350  06320000  06329900  06332300  06407000  06411000  06415000
          06417000  06421000  06423000  06505000  07015000  07118000  07158000  07239000  07243000  07405000  07412000
          07452000  07462250  07462750  07471750  07485000  07494000  07598000  07646440  07646555  07666000  07670000
          07672500  07679000  07681500  07686000  07688500  07695500  07730000  07830000  07839000  07845000  07850000
          07853000  07855000  07859030  07859080  07900000  07902000  07906000  07910000  07924000  07938000  07940000
          07942000  07943000  08059000  08134000  08160000  08178000  08229000  08392000  08401040  08406000  08463000
          08472000  08493580  08493610  08493770  08511000  08597400  08601000  08653120  08671000  08681000  08728200
          08729000  08745000  08793000  08813000  08850000  08860500  08922000  08941000  08942000  08943000  08994530
          08995500  08996000  10134550  10173000  10298000  10318000  10327000  10368000  10565300  10638000  10647000
          10664000  10866000  10898000  10969000  10991000  12007000  12033000  12036500  12052000  12056000  12065200
          13164000  13196530  13911000  13911300  13912000  14259040  14259075  14259120  15081000  15115100  15119000
          15233000  15254800  15256000  15261000  15263100  15291200  15294000  15297000  15343000  15364000  15367000
          16125000  16132000  16160000  16205000  16206000  16208000  16211200  16212300  16213100  16217000  16253000
          16284000  16316000  16326000  16327000  16347000  16367000  16384500  16397000  16418000  16483000
ERR  --  DEFINE  --  DECLARED IN SEGMENT 15 AT 02238270
          02238280  02238430  02238520  02238522  02238552  02238602
ERRMAX  --  INTEGER  --  DECLARED IN SEGMENT 3 AT 01001550
          02211500 *02502000**09028910*
ERRNUM  --  INTEGER  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 01716000
ERRNUM  --  INTEGER  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 01718000
ERRNUM  --  INTEGER  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 05012000
          05044000
ERRNUM  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 45 AT 05016000
          05017000  05026000
ERRNUM  --  INTEGER  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 05105000
          05106000  05107100  05107200
ERROR  --  STREAM LABEL  --  DECLARED IN SEGMENT 10 AT 02091500  --  OCCURS AT 02114000
          02119000
ERROR  --  DEFINE  --  DECLARED IN SEGMENT 32 AT 02958000
          02959000
ERROR  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 05110000
          07239000  07666000  07670000  07672500  07679000  07681500  07686000  07688500  07695500  07900000  08463000
          08472000  08511000  08671000  08681000  08745000  08793000  08813000  08850000  10567000  10638000  10647000
          10664000  13767400
ERROR  --  LABEL  --  DECLARED IN SEGMENT 57 AT 06060110  --  OCCURS AT 06062565
          06062120  06062165  06062190  06062210  06062220  06062420  06062430
ERROR  --  LABEL  --  DECLARED IN SEGMENT 128 AT 12060300  --  OCCURS AT 12065100
          12060500  12062400  12063200  12064800
ERRORCOUNT  --  INTEGER  --  DECLARED IN SEGMENT 2 AT 00501000
          02211500 *05035000* 09028000  09028150 *09028900* 09407050  09417000
ERRORNUMBER  --  REAL  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 05289000
          05287000  05288000  05295000
ERRORTOG  --  BOOLEAN  --  DECLARED IN SEGMENT 3 AT 01412000
          05032000 *05047000**05119000**07009000**07221000**07645000**07646475**07734000**09275000**10187000**13282000*
          13330600 *13330650**14130000**14443000* 14511000 *16130000**16214100* 06062120  06062165  06062190  06062210
          06062220  06062420  06062430
ERRX  --  DEFINE  --  DECLARED IN SEGMENT 128 AT 12060400
          12060500  12062400  12063200  12064800
EX  --  LABEL  --  DECLARED IN SEGMENT 119 AT 10085000  --  OCCURS AT 10173000
          10112100  10119000  10124200  10134100  10150205  10159000  10160000  10162000  10170000
EXAMIN  --  REAL STREAM PROCEDURE  --  DECLARED IN SEGMENT 2 AT 00511000
          00512000  00529000  00531000  00533000  00536000  00539000  02013317  02013516  02214200  02225750  02228250
          02229000  02230000  02230100  02230750  02238512  02238572  02665000  02680000  02696000  02697000  02730000
          02745000  02750000  02753000  02761000  02769000  02786000  02791000  02807000  02813000  02888000  02936000
          02940000  02946000  10167400  10167600
EXAMINELAST  --  REAL STREAM PROCEDURE  --  DECLARED IN SEGMENT 2 AT 00512100
          00512400  00512600  02873000
EXIT  --  STREAM LABEL  --  DECLARED IN SEGMENT 3 AT 02001080  --  OCCURS AT 02001590
```

```
          02001380  02001550
EXIT   -- LABEL  -- DECLARED IN SEGMENT 8 AT 02013175  --  OCCURS AT 02013720
          02013222  02013305  02013350  02013380  02013405  02013435  02013702
EXIT   -- STREAM LABEL  -- DECLARED IN SEGMENT 10 AT 02091500  --  OCCURS AT 02119500
          02104500  02112000  02114500  02115500  02118500
EXIT   -- LABEL  -- DECLARED IN SEGMENT 13 AT 02210000  --  OCCURS AT 02225000
          02215500  02215750  02216250  02224060
EXIT   -- LABEL  -- DECLARED IN SEGMENT 15 AT 02238120  --  OCCURS AT 02238830
          02238760
EXIT   -- LABEL  -- DECLARED IN SEGMENT 21 AT 02360000  --  OCCURS AT 02602000
          02375000  02413000  02488000  02571000
EXIT   -- LABEL  -- DECLARED IN SEGMENT 32 AT 02957000  --  OCCURS AT 02967500
          02959000  02962500
EXIT   -- LABEL  -- DECLARED IN SEGMENT 42 AT 04311010  --  OCCURS AT 04322100
          04311100
EXIT   -- LABEL  -- DECLARED IN SEGMENT 43 AT 04501000  --  OCCURS AT 04558000
          04526200  04532000  04538000  04546000  04552000
EXIT   -- LABEL  -- DECLARED IN SEGMENT 52 AT 05273000  --  OCCURS AT 05286000
          05273500  05274400  05284000
EXIT   -- LABEL  -- DECLARED IN SEGMENT 54 AT 05325030
EXIT   -- LABEL  -- DECLARED IN SEGMENT 55 AT 05334000  --  OCCURS AT 05375000
          05372000
EXIT   -- LABEL  -- DECLARED IN SEGMENT 57 AT 06060100  --  OCCURS AT 06074000
          06062435  06062540  06062560  06062565  06066000  06067000  06070000  06072000  06072130  06072210  06072230
          06072250  06072270  06072290  06073100  06073200  06073820  06073870
EXIT   -- LABEL  -- DECLARED IN SEGMENT 58 AT 06090000  --  OCCURS AT 06131000
          06095000  06119000  06121000  06125000
EXIT   -- LABEL  -- DECLARED IN SEGMENT 60 AT 06182000  --  OCCURS AT 06210000
          06191000  06194000
EXIT   -- LABEL  -- DECLARED IN SEGMENT 61 AT 06224000  --  OCCURS AT 06278000
          06231000  06234000  06239000  06241000  06253000  06261000  06263000  06272000  06274000
EXIT   -- LABEL  -- DECLARED IN SEGMENT 64 AT 06313500  --  OCCURS AT 06336600
          06314000  06314400  06320000  06329900  06332300
EXIT   -- LABEL  -- DECLARED IN SEGMENT 65 AT 06339000  --  OCCURS AT 06346000
          06341000
EXIT   -- LABEL  -- DECLARED IN SEGMENT 67 AT 06390000  --  OCCURS AT 06408000
          06401000  06404000
EXIT   -- LABEL  -- DECLARED IN SEGMENT 68 AT 06501000  --  OCCURS AT 06512000
          06505000  06510000
EXIT   -- LABEL  -- DECLARED IN SEGMENT 70 AT 07066000  --  OCCURS AT 07385000
          07158000  07239000  07243000  07244000
EXIT   -- LABEL  -- DECLARED IN SEGMENT 71 AT 07394000  --  OCCURS AT 07425000
          07402000  07405000
EXIT   -- LABEL  -- DECLARED IN SEGMENT 73 AT 07461750  --  OCCURS AT 07472000
          07462250  07462750
EXIT   -- LABEL  -- DECLARED IN SEGMENT 76 AT 07504000  --  OCCURS AT 07530000
          07510000  07527000
EXIT   -- LABEL  -- DECLARED IN SEGMENT 77 AT 07549000  --  OCCURS AT 07557000
          07555000
EXIT   -- LABEL  -- DECLARED IN SEGMENT 78 AT 07562000  --  OCCURS AT 07587000
          07573000  07574000  07576000  07579000  07584000
EXIT   -- LABEL  -- DECLARED IN SEGMENT 79 AT 07594000  --  OCCURS AT 07646000
          07598000
EXIT   -- LABEL  -- DECLARED IN SEGMENT 83 AT 07654500  --  OCCURS AT 07699500
          07686500  07688500  07692000
EXIT   -- LABEL  -- DECLARED IN SEGMENT 84 AT 07659000  --  OCCURS AT 07683000
          07680500
```

```
EXIT  -- LABEL  -- DECLARED IN SEGMENT 85 AT 07718000  -- OCCURS AT 07757000
        07726000  07729000  07729100  07729200  07730000  07733000  07735000  07736000  07737000  07738000  07740000
        07741000  07742000  07743000  07744000  07745000  07746000  07747000  07748000  07749000  07750000  07751000
        07752000  07753000
EXIT  -- LABEL  -- DECLARED IN SEGMENT 88 AT 07935000  -- OCCURS AT 07954000
        07938000  07940000  07942000  07943000
EXIT  -- LABEL  -- DECLARED IN SEGMENT 89 AT 08017000  -- OCCURS AT 08232000
        08178000  08222000  08229000
EXIT  -- LABEL  -- DECLARED IN SEGMENT 90 AT 08092000  -- OCCURS AT 08172000
        08134000  08160000
EXIT  -- LABEL  -- DECLARED IN SEGMENT 91 AT 08238000  -- OCCURS AT 08288000
        08278000
EXIT  -- LABEL  -- DECLARED IN SEGMENT 92 AT 08378000  -- OCCURS AT 08492000
        08393000  08396000  08401040  08406000  08417010  08418700  08428000  08446000  08450300  08453000  08463000
        08466000  08472000  08476000  08489580
EXIT  -- LABEL  -- DECLARED IN SEGMENT 93 AT 08493040  -- OCCURS AT 08493180
        08493170
EXIT  -- LABEL  -- DECLARED IN SEGMENT 95 AT 08493510  -- OCCURS AT 08493920
        08493650  08493770
EXIT  -- LABEL  -- DECLARED IN SEGMENT 96 AT 08500000  -- OCCURS AT 08526000
        08504000  08511000  08517000  08521000
EXIT  -- LABEL  -- DECLARED IN SEGMENT 97 AT 08571000  -- OCCURS AT 08699000
        08594000  08597400  08601000  08613600  08615000  08634000  08640000  08658300  08661000  08671000  08674000
        08681000  08684000  08696550  08698500
EXIT  -- LABEL  -- DECLARED IN SEGMENT 98 AT 08713000  -- OCCURS AT 08754000
        08723000  08728200  08730000  08734000  08745000  08750000
EXIT  -- LABEL  -- DECLARED IN SEGMENT 99 AT 08772000  -- OCCURS AT 08823000
        08786000  08793000  08797000  08813000  08819000
EXIT  -- LABEL  -- DECLARED IN SEGMENT 100 AT 08833000  -- OCCURS AT 08859000
        08843000  08850000  08854000
EXIT  -- LABEL  -- DECLARED IN SEGMENT 119 AT 10085000  -- OCCURS AT 10174000
        10133000  10134100  10134560  10150260  10172000
EXIT  -- LABEL  -- DECLARED IN SEGMENT 120 AT 10259000  -- OCCURS AT 10284000
        10263900  10275000
EXIT  -- LABEL  -- DECLARED IN SEGMENT 124 AT 10564000  -- OCCURS AT 10679000
        10565300  10571000  10601000  10631000  10638000  10647000  10664000  10669000  10673000  10675000
EXIT  -- LABEL  -- DECLARED IN SEGMENT 125 AT 10709000  -- OCCURS AT 10938000
        10793000  10798000  10829000  10856000  10866000  10877000  10898000  10903000  10909000  10924000  10926550
EXIT  -- LABEL  -- DECLARED IN SEGMENT 126 AT 10956000  -- OCCURS AT 12001000
        10969000  10991000
EXIT  -- LABEL  -- DECLARED IN SEGMENT 127 AT 12005000  -- OCCURS AT 12058000
        12033000  12040000  12052000  12056000
EXIT  -- LABEL  -- DECLARED IN SEGMENT 128 AT 12060300  -- OCCURS AT 12065300
        12063610
EXIT  -- STREAM LABEL  -- DECLARED IN SEGMENT 131 AT 13007000  -- OCCURS AT 13018000
        13015000
EXIT  -- LABEL  -- DECLARED IN SEGMENT 148 AT 14269100  -- OCCURS AT 14269860
        14269710  14269760
EXIT  -- LABEL  -- DECLARED IN SEGMENT 154 AT 15076000  -- OCCURS AT 15377000
        15081000
EXIT  -- LABEL  -- DECLARED IN SEGMENT 155 AT 15088000  -- OCCURS AT 15127000
        15112000  15115200  15119000
EXIT  -- LABEL  -- DECLARED IN SEGMENT 156 AT 15184000  -- OCCURS AT 15376000
        15233000  15254800  15256000  15261000  15263100  15275000  15291200  15294000  15297000  15305000  15331000
        15337000  15341000  15343000  15365000  15367000
EXIT  -- LABEL  -- DECLARED IN SEGMENT 161 AT 16117000  -- OCCURS AT 16144000
        16125000  16132000
EXIT  -- LABEL  -- DECLARED IN SEGMENT 162 AT 16193000  -- OCCURS AT 16239000
```

```
        16205000  16206000  16208000  16211200  16212300  16213100  16217000
EXIT  -- LABEL  -- DECLARED IN SEGMENT 163 AT 16251000  --  OCCURS AT 16270000
        16253000
EXIT  -- LABEL  -- DECLARED IN SEGMENT 164 AT 16313000  --  OCCURS AT 16355000
        16316000  16326000  16327000  16342000  16347000
EXIT  -- LABEL  -- DECLARED IN SEGMENT 165 AT 16365000  --  OCCURS AT 16398000
        16367000  16384500
EXIT  -- LABEL  -- DECLARED IN SEGMENT 157 AT 16475000  --  OCCURS AT 16494000
        16484000  16487000  16488000  16490000  16491000  16492000  16493000
EXITOUT  -- LABEL  -- DECLARED IN SEGMENT 8 AT 02013175  --  OCCURS AT 02013325
        02013317
EXPI  -- DEFINE  -- DECLARED IN SEGMENT 3 AT 01570000
        04077000
EXPLICITFORMAT  -- PROCEDURE  -- DECLARED IN SEGMENT 3 AT 08860050  --  FORWARD AT 03056100
        08450020  08658020  08860800
EXPRSS  -- INTEGER PROCEDURE  -- DECLARED IN SEGMENT 3 AT 06139000  --  FORWARD AT 03007000
        06133000 *06143000**06145000* 06146000  06259000  06299000  06407000  07125000  07133000  10327000
F  -- INTEGER  -- DECLARED IN SEGMENT 3 AT 01688000
       *08860250* 08860400  10071000  10074000 *10076000* 10116000 *10127000* 10127100 *10127200**13783000* 13791000
        13795000  13798000  13807000  13810000  13814000 *14077000**14078000**14092000**14100000* 14108000 *14115000*
       *14177000* 14178000 *14179000* 14181000
F  -- STREAM VARIABLE  -- VALUE PARAMETER  -- DECLARED IN SEGMENT 3 AT 02262000
        02263000
F  -- STREAM VARIABLE  -- NAME PARAMETER  -- DECLARED IN SEGMENT 23 AT 02441000
        02443000
F  -- STREAM VARIABLE  -- NAME PARAMETER  -- DECLARED IN SEGMENT 23 AT 02446000
        02448000
F  -- BOOLEAN  -- DECLARED IN SEGMENT 67 AT 06391000
       *06397000* 06397200  06401000  06404000
F  -- REAL  -- DECLARED IN SEGMENT 158 AT 16031000
       *16033000* 16036000  16040000  16043000  16044000
FA  -- DEFINE  -- DECLARED IN SEGMENT 3 AT 01420000
        07111240  07157000  07197000  08493520  08493840  15262500  15272900
FACTOP  -- DEFINE  -- DECLARED IN SEGMENT 3 AT 01278000
        06019000  06036000  06050000  06507000  07157000  07198000  07465000  07687500  07816000  07837000  07859050
        07941000  08401030  08412050  08413120  08414000  08416010  08418250  08431000  08460000  08597300  08620020
        08622000  08623000  08629000  08654000  08668000  08728400  08739000  08742000  08805000  10318000  12041100
        12063800  13337000  13582000  14601100  15253000  15254700
FAH  -- LABEL  -- DECLARED IN SEGMENT 61 AT 06224500  --  OCCURS AT 06262500
        06227530
FAULTDEC  -- PROCEDURE  -- DECLARED IN SEGMENT 3 AT 13900000  --  FORWARD AT 03080000
        13920000  14148000
FAULTID  -- DEFINE  -- DECLARED IN SEGMENT 3 AT 01278100
        07729100  07925100  13911300  13913000
FAULTLEVEL  -- INTEGER  -- DECLARED IN SEGMENT 3 AT 01750000
       *06397200**06397300**09038000* 13767100 *13767200**13905000**13906000* 14569100  14611000
FAULTOG  -- BOOLEAN  -- DECLARED IN SEGMENT 3 AT 01753000
        06397100  07923000 *07923100**07926100* 13750000  13753000  13759000 *14058100* 14488500  14537000  14539000
        14561000  14564000 *14611000*
FAULTOGO  -- BOOLEAN  -- DECLARED IN SEGMENT 143 AT 14034000
       *14058100* 14611000
FAULTSTMT  -- PROCEDURE  -- DECLARED IN SEGMENT 3 AT 07920000  --  FORWARD AT 03079000
        07729100  07925100  07927000
FBIT  -- BOOLEAN  -- VALUE PARAMETER  -- DECLARED IN SEGMENT 3 AT 07047000
        07045000  07046000  07081000  07240000  07254000  07281900  07368000
FCR  -- INTEGER  -- DECLARED IN SEGMENT 3 AT 01330000
        02194000  02195000  02195750 *02214000* 02214200 *02225250* 02225750  02227500  02228250  02228750  02230000
```

```
        02230100  02238572  02238580  02731000  05038000  05057000 *07025010* 07025020 *07025030*
FCR  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 01742200
        01742700
FCR  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 02016000
        02017000
FEIL  --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 3 AT 04130000
        04132000
FEJ  --  INTEGER STREAM PROCEDURE  --  DECLARED IN SEGMENT 23 AT 02441000
        02444000  02445000  02452000
FETCH  --  REAL STREAM PROCEDURE  --  DECLARED IN SEGMENT 3 AT 02262000
        02263000
FF  --  DEFINE  --  DECLARED IN SEGMENT 167 AT 17002008
        17025100  17080000
FGEN  --  LABEL  --  DECLARED IN SEGMENT 70 AT 07067000  --  OCCURS AT 07248000
        07106000
FI  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01420500
        06073500  06073550  15263000  15364500  15369100  15369300  15369600
FIELDDEC  --  LABEL  --  DECLARED IN SEGMENT 143 AT 14016000  --  OCCURS AT 14269020
        14021000
FIELDID  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01278090
        05273100  05273900  06313550  06314050  14269240  14269442  17095000  17095100
FIELDV  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01298600
FIL  --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 104 AT 09010000
        09013000
FILATTINT  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01579300
        08493860
FILEATTRIBUTEHANDLER  --  INTEGER PROCEDURE  --  DECLARED IN SEGMENT 3 AT 08493470  --  FORWARD AT 03061010
        06092015  06227520  07111240  07729200 *08493870* 08493930  13160000  13167000
FILEATTRIBUTEINDX  --  REAL PROCEDURE  --  DECLARED IN SEGMENT 3 AT 08493010
       *08493170* 08493190  08493610  13157000  13163000
FILEATTRIBUTES  --  REAL ARRAY  --  DECLARED IN SEGMENT 3 AT 01303500
        08493070 *08493090* 08493140  08493160  08493710  08493740  08493820  08493830  08493880
FILEDEC  --  LABEL  --  DECLARED IN SEGMENT 143 AT 14015000  --  OCCURS AT 14157000
        14020000
FILEID  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01184000
        02761500  06073250  06092005  06227500  07107000  07109000  07117000  07260000  07467750  07685500  07729190
        07814200  07831000  07846000  08403000  08508000  08598000  08727000  08790000  08847000  08908000  08967000
        08973000  10565000  10792000  13064000  13081000  14430000
FILEIDENT  --  DEFINE  --  DECLARED IN SEGMENT 124 AT 10505000
        10569000  10576000  10583000  10649000  10657000
FILEIDENT  --  DEFINE  --  DECLARED IN SEGMENT 125 AT 10712000
        10796000  10891000
FILEINDEX  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01561090
        02013490  02013505  02013580
FILEINX  --  INTEGER  --  DECLARED IN SEGMENT 3 AT 01561120
       *02013210**02013230* 02013235  02013245  02013250  02013255  02013270  02013275  02013295  02013490  02013528
        02013568  02013570 *02013580* 02013665  02013680  02013685  02013697  02013705  02013710  02013715  02200250
        02200500  02204250
FILEND  --  INTEGER  --  DECLARED IN SEGMENT 102 AT 08902000
       *08918000* 08923000
FILENO  --  INTEGER  --  DECLARED IN SEGMENT 3 AT 01584000
        08946000  08949000 *08951000**09031000* 09361140  09405200  13102000 *13109000*
FILENO  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 131 AT 13020000
        13021000  13024000
FILER  --  INTEGER  --  DECLARED IN SEGMENT 102 AT 08902000
       *08914000* 08918000  08923000
FILETHING  --  REAL  --  DECLARED IN SEGMENT 3 AT 01625100
```

```
      *09031100* 14451100 *14451300* 14451500 *14451700**16287000*
FILEV  -- DEFINE  --  DECLARED IN SEGMENT 3 AT 01296000
       13042000
FILID  -- STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 131 AT 13020000
       13021000  13029000
FILID  -- REAL  --  DECLARED IN SEGMENT 131 AT 13039000
       13080000  13095000 *13100000* 13102000
FILID1  -- STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 131 AT 13020000
       13033000
FILLIT  -- STREAM PROCEDURE  --  DECLARED IN SEGMENT 54 AT 05325040
       05325440  05325460  05325470  05325480
FILLIT  -- BOOLEAN PROCEDURE  --  DECLARED IN SEGMENT 83 AT 07657000
       07664000 *07682500* 07684000  07695000
FILLSTMT  -- PROCEDURE  --  DECLARED IN SEGMENT 3 AT 07653500
       07700000  07750000
FILLTEMP  -- REAL ARRAY  --  DECLARED IN SEGMENT 83 AT 07656500
       07695000  07697000
FILLV  -- DEFINE  --  DECLARED IN SEGMENT 3 AT 01229000
       16379000
FINAL  -- LABEL  --  DECLARED IN SEGMENT 120 AT 10258300  --  OCCURS AT 10263900
       10264400  10264420
FINALL  -- DEFINE  --  DECLARED IN SEGMENT 125 AT 10762000
       10926000  10929000
FINDOPTION  -- BOOLEAN PROCEDURE  --  DECLARED IN SEGMENT 3 AT 02307000
      *02317000* 02318000  02327000  02967000
FINI  -- LABEL  --  DECLARED IN SEGMENT 157 AT 16475000  --  OCCURS AT 16495000
       16476000  16478000  16483000  16486000  16489000
FINIS  -- STREAM LABEL  --  DECLARED IN SEGMENT 10 AT 02091500  --  OCCURS AT 02125500
       02095500  02096500  02099500  02110000
FINISHBOO  -- LABEL  --  DECLARED IN SEGMENT 70 AT 07065000  --  OCCURS AT 07212000
       07210000
FINISHED  -- LABEL  --  DECLARED IN SEGMENT 40 AT 04204000  --  OCCURS AT 04225000
       04220000
FINISHNUMBER  -- LABEL  --  DECLARED IN SEGMENT 27 AT 02638000  --  OCCURS AT 02847000
FINISHPT  -- REAL  --  DECLARED IN SEGMENT 3 AT 01561200
       02013495 *02013500**02013600* 02132000  02207750  02224000  02733000
FIO  -- DEFINE  --  DECLARED IN SEGMENT 3 AT 01420500
       08493520  13160000  13167000
FIRST  -- INTEGER  --  NAME PARAMETER  --  DECLARED IN SEGMENT 3 AT 05271000
       05270000 *05273300**05274100**05280000* 05282000
FIRST  -- INTEGER  --  DECLARED IN SEGMENT 64 AT 06312500
      *06313650**06314150**06315000* 06317000  06328000  06331700  06336200  06336300
FIRST  -- INTEGER  --  DECLARED IN SEGMENT 65 AT 06339000
       06341000  06343000
FIRSTIMEX  -- BOOLEAN  --  DECLARED IN SEGMENT 3 AT 01561202
      *02013540**02013541**02013628* 02220250 *02220500* 02223250 *02223500*
FIRSTTIME  -- LABEL  --  DECLARED IN SEGMENT 13 AT 02210000  --  OCCURS AT 02213500
       02210750
FIRSTTIME  -- BOOLEAN  --  DECLARED IN SEGMENT 172 AT 17069400
       17091155 *17091162**17117920*
FIRSTX  -- REAL  --  DECLARED IN SEGMENT 3 AT 01621000
       13608000 *13612000* 13623000  14051000 *14052000* 14534000  14555000  14571000 *14608000*
FIRSTXO  -- INTEGER  --  DECLARED IN SEGMENT 143 AT 14033000
      *14051000* 14608000
FIX  -- STREAM PROCEDURE  --  DECLARED IN SEGMENT 23 AT 02446000
       02451000  02454000
FIX  -- PROCEDURE  --  DECLARED IN SEGMENT 90 AT 08082000
```

```
        08090000  08157000  08171000
FIXC  --  PROCEDURE  --  DECLARED IN SEGMENT 157 AT 16029000
        16057000  16141000  16233000  16236000  16238000
FIXDEFINEINFO  --  REAL PROCEDURE  --  DECLARED IN SEGMENT 3 AT 12101000  --  FORWARD AT 01717900
        02897000 *12108000*
FIX1  --  REAL  --  DECLARED IN SEGMENT 162 AT 16195000
       *16218000* 16233000  16238000
FIX2  --  REAL  --  DECLARED IN SEGMENT 162 AT 16195000
       *16232000* 16236000
FL  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01420000
        06062400  07173000  07191000  07687500  07815000  07836000  07859040  07941000  08401020  08459000  08597200
        08667000  08728300  10317000  12037000  12063100  15263000
FL  --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 46 AT 05050000
        05053000
FLAG  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 05012000  --  FORWARD AT 01716000
        02013180  02013702  02226250  02237250  02238280  02238430  02238520  02238522  02238552  02238602  02316000
        02501000  02593000  02599000  02650000  02697600  02773000  02775000  02776800  02819000  02821000  02824000
        02828000  02899000  02943000  02944000  02959000  02962000  02976500  02977500  04013000  05101000  05106000
        05119000  05347000  05361000  05370000  06062380  06092015  06317000  06323500  06331100  06336200  07024000
        07025000  07127000  07165000  07175000  07217000  07217520  07221000  07226000  07233000  07369000  07382000
        07600000  07604000  07678000  07687500  07729500  07731000  07859050  08493705  08493720  08728400  08948000
        10076000  10090000  10136500  10165500  10179000  10187000  10243000  10264420  10912100  10926550  12025000
        12030000  12036200  12039100  12061300  12062200  12062800  12063610  12115000  12126100  13052000  13065000
        13088085  13088100  13088105  13101200  13105000  13145000  13182000  13192000  13196380  13210000  13282000
        13312500  13330600  13332000  13338000  13339200  13340000  13457000  13462000  13467000  13473000  13504000
        13531000  13549000  13552500  13587000  13726000  13787000  13791000  13798000  13806000  13819410  14053100
        14066000  14116400  14116500  14135000  14136000  14136100  14137000  14146000  14153000  14174000  14187000
        14211000  14238000  14263000  14269820  14286000  14295000  14308000  14310000  14311100  14323100  14326000
        14333200  14337000  14339000  14346000  14351000  14358000  14371000  14442000  14487010  14507180  14508000
        15262600  16163000  16171000  16214100  16265000  16434000
FLAGIT  --  PROCEDURE  --  DECLARED IN SEGMENT 8 AT 02013176
        02013183  02013222  02013305  02013350  02013380  02013405  02013435
FMT  --  SWITCH FORMAT  --  DECLARED IN SEGMENT 30 AT 02927150
        02927440  02927460
FN  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 103 AT 08932000
        08933000  08935000
FOOT  --  INTEGER  --  DECLARED IN SEGMENT 90 AT 08091000
       *08162000* 08166000
FORCLASS  --  BOOLEAN PROCEDURE  --  DECLARED IN SEGMENT 89 AT 08017500
       *08017600* 08017950  08182000  08184000  08186000
FORLIST  --  PROCEDURE  --  DECLARED IN SEGMENT 89 AT 08080000
        08156000  08172000  08231000
FORMAL  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01149000
        05145000  05170000  05209000  05218000  05236000  05390100  05396100  05403100  07082000  07083000  07166000
        08941100  13202000  13244000  13310080  14259090  14427000  16283000
FORMALF  --  BOOLEAN  --  DECLARED IN SEGMENT 3 AT 01591000
        06397000  07264000  07275000  07327000  07347000  07400000  07410000  07411000 *13202000**13328000**13342000*
        13372000 *13913100**14223000**14243000* 14245000 *14507120*
FORMALNAME  --  DEFINE  --  DECLARED IN SEGMENT 62 AT 06282800
        06287000
FORMALNAME  --  DEFINE  --  DECLARED IN SEGMENT 155 AT 15089000
        15092020  15105000
FORMALV  --  DEFINE  --  DECLARED IN SEGMENT 89 AT 08014000
        08056000  08065000  08098000  08108000  08130000  08179000  08212000  08217000  08227000
FORMATBIT  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01000950
        02567000  14461500  14493500
FORMATDEC  --  LABEL  --  DECLARED IN SEGMENT 143 AT 14015000  --  OCCURS AT 14161000
        14020000
```

```
FORMATGEN  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 13774000  --  FORWARD AT 03056000
        13194000  13819000  14162000
FORMATOG  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01001240
        14461500  14493500
FORMATPHRASE  --  BOOLEAN PROCEDURE  --  DECLARED IN SEGMENT 3 AT 10083000  --  FORWARD AT 03056200
        08860350  10119000 *10173000* 10174000  13792000  13808000
FORMATTYPE  --  DEFINE  --  DECLARED IN SEGMENT 125 AT 10758000
        10833000  10869000  10871000  10891100
FORMATV  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01295000
        13192000  14158000  14159000
FORSTMT  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 08008000  --  FORWARD AT 03052000
        07747000  08232000  10294000
FORV  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01223000
        10294000
FORWARDBRANCH  --  INTEGER  --  DECLARED IN SEGMENT 90 AT 08091000
       *08152000* 08157000  08169000  08171000
FORWARDREF  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01007485
        14469100  17092000
FORWARDV  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01254000
        14238000  14467000
FORWART  --  REAL  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 90 AT 08084000
        08082000  08083000  08086000
FOUND  --  LABEL  --  DECLARED IN SEGMENT 20 AT 02309000  --  OCCURS AT 02315000
        02313000
FOUND  --  LABEL  --  DECLARED IN SEGMENT 40 AT 04204000  --  OCCURS AT 04221000
        04209000
FOUNDLB  --  BOOLEAN  --  DECLARED IN SEGMENT 148 AT 14269080
      *14269360**14269440* 14269660
FP  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01420000
        06092015  06107000  06227520  06250000  08493840  15119000  15236000  15263050  15349000
FPART  --  LABEL  --  DECLARED IN SEGMENT 27 AT 02639000  --  OCCURS AT 02790000
        02682000
FR  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01420000
        08228000  15305000
FRMTID  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01182000
        07106000  08439000  08644000  13193000  13806000  14091000  14161000
FROM  --  REAL ARRAY  --  NAME PARAMETER  --  DECLARED IN SEGMENT 3 AT 02927110
        02927100  02927530  02927540  02927570
FROM  --  BOOLEAN  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 03029000
FROM  --  INTEGER  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 03037000
        03036000
FROM  --  REAL  --  NAME PARAMETER  --  DECLARED IN SEGMENT 3 AT 03038000
FROM  --  INTEGER  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 04116000
        04114000  04115000  04120000  04120100  04120500
FROM  --  BOOLEAN  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 07391000
        07402000  07404000  07411100
FRONT  --  INTEGER  --  DECLARED IN SEGMENT 75 AT 07491000
       *07493000* 07496000
FRSTLEVEL  --  INTEGER  --  DECLARED IN SEGMENT 3 AT 01402000
        05116000  06365000 *14478030*
FS  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01420000
        07729200  07738000  08493705  08493770  15102000  15104000  15114000  15319000  15329000  15334000  15335000
        15342100  15364000  15367000
FSAVE  --  REAL  --  DECLARED IN SEGMENT 3 AT 01340500
      *02682000**02688000**02758000**02803000* 02826000  02845000
FUNCMONFILE  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01543000
        07422000
```

```
FUNCTOG  --  BOOLEAN  --  DECLARED IN SEGMENT 3 AT 01586000
        13740000  14054000  14187000 *14275000**14288000**14296000* 14389000  14450000  14452000  14478040 *14553000*
       *14605000*
FUNCTOGO  --  BOOLEAN  --  DECLARED IN SEGMENT 143 AT 14034000
       *14054000* 14553000  14605000
FWDSEQNO  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 172 AT 17047300
        17047400  17050700  17051000
FWDSEQNO  --  OWN REAL  --  DECLARED IN SEGMENT 177 AT 17091150
       *17092600**17093400* 17094900
FWDTOG  --  BOOLEAN  --  DECLARED IN SEGMENT 143 AT 14025100
        14116100 *14276000**14312000* 14333100  14373000
FWDTOG  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 172 AT 17047200
        17047400  17050600
FWDTOG  --  OWN BOOLEAN  --  DECLARED IN SEGMENT 177 AT 17091110
       *17093300* 17094800 *17095400*
FZERO  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01692000
        14078000
G  --  REAL  --  DECLARED IN SEGMENT 3 AT 01610000
       *13042000**13045000**13055000* 13072000  13075000  13091000  13097000  13100000  13192000 *13196420* 13196600
       *13231000* 13236000 *13716000**13720000**13722000**13724000* 13726000 *14084000* 14085000  14091000 *14092000*
        14094000  14098000 *14100000**14115000**14116000* 14116300  14116500  14117000  14158000  14159000 *14167000*
       *14222000**14278000**14281000* 14284000  14292000  14294000 *14310000* 14311100 *14349000**14352000* 14353000
       *14387000**14446000* 14449000 *14470000**14567000**14575000**14601000*
GENERATE  --  LABEL  --  DECLARED IN SEGMENT 164 AT 16313000  --  OCCURS AT 16345000
        16322000
GENGO  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 03016000
GENGO  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 06356000
        06381000  06406000  07271000  10983000
GEQ  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01647000
        08050000  10959000
GET  --  INTEGER PROCEDURE  --  DECLARED IN SEGMENT 3 AT 05307000  --  FORWARD AT 03019000
        04050000  04053000  04057000  04102000  04105000  04182000 *05312000* 07607000  07608000  07646720  07727060
        08086000  13113000  13148200  13157020  13235000  13258000  13267000  13273000  14451300  16033000  16038000
        16092000
GETALPHA  --  REAL STREAM PROCEDURE  --  DECLARED IN SEGMENT 3 AT 05317000
        05323000  05325000  15033000
GETEM  --  LABEL  --  DECLARED IN SEGMENT 15 AT 02238120  --  OCCURS AT 02238560
        02238350
GETF  --  INTEGER STREAM PROCEDURE  --  DECLARED IN SEGMENT 2 AT 00523000
        00524000 *00526000* 00527000  00529000
GETINT  --  BOOLEAN PROCEDURE  --  DECLARED IN SEGMENT 3 AT 10177000  --  FORWARD AT 03031000
        10150280  10159100  10167800 *10178100* 10180000
GETLP  --  LABEL  --  DECLARED IN SEGMENT 144 AT 14076000  --  OCCURS AT 14084000
        14089000
GETSPACE  --  INTEGER PROCEDURE  --  DECLARED IN SEGMENT 3 AT 05331000  --  FORWARD AT 03051000
        05132000  05260000 *05375000* 05378000  07646450  08162000  08860250  09274000  10882000  10915000  13345000
        13500000  13783000  13913000  14044000  14179000  14232000  14240000  14241000  14558000
GETSYL  --  INTEGER STREAM PROCEDURE  --  DECLARED IN SEGMENT 53 AT 05309000
        05310000  05312000
GETVOID  --  STREAM PROCEDURE  --  DECLARED IN SEGMENT 3 AT 01756000
        01784000  02410000  02485000
GET7  --  STREAM PROCEDURE  --  DECLARED IN SEGMENT 131 AT 13005000
        13080000  13091000  13095000
GIT  --  INTEGER PROCEDURE  --  DECLARED IN SEGMENT 3 AT 05123000  --  FORWARD AT 01719000
        02897000 *05124000* 05391000  05397000  05404000  07276000  07361000  07410000  07412000  07422000  07442000
        07515000  07520000  07539000  07601000  08942000  10649000  10814000  10839000  10883000  10916000  10929000
        10930000  12108000  12123500  13249000  13257000  14438000  15010010  15010020  15233100  15242000  15244000
        15273100  15281000  15324000  16070000  16089000  16138000  16163000  16254000  16257000
```

```
GLOBALNFOO  --  REAL  --  DECLARED IN SEGMENT 3 AT 01620100
        13339100  14055250 *14055300**14606100*
GNAT  --  INTEGER PROCEDURE  --  DECLARED IN SEGMENT 3 AT 05128000  --  FORWARD AT 03020000
        04076000  04077000  04078000 *05131000**05132000* 05133000  05316000  06062285  06062540  06066000  06334500
        06335200  06380000  06400000  06405000  07270000  07316000  07420000  07421000  07423000  07514000  07516200
        07618000  07620000  07953000  08287200  08401060  08402000  08493860  08514000  08597600  08604000  08698900
        08906000  08911100  08926000  08955000  08965000  08997000  10888000  10890000  10891200  10982000  12041130
        12061750  12064000  13267000  15008000  15010000  15012000  15271100  15322000  15323000  15328000  15361000
        15362000  15374100
GNC  --  STREAM LABEL  --  DECLARED IN SEGMENT 10 AT 02091500  --  OCCURS AT 02104000
        02097000
GOBBLE  --  LABEL  --  DECLARED IN SEGMENT 70 AT 07066000  --  OCCURS AT 07303000
        07319000
GOGEN  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 07535000  --  FORWARD AT 03023000
        07509000  07545000  07572000  07573000  07575000  07583000  13274000
GOGOGO  --  BOOLEAN  --  DECLARED IN SEGMENT 3 AT 01000810
       *07948000* 08287000 *08287300**08424100**08468100**08611100**08677100* 08698750 *08698950*
GOMCP  --  LABEL  --  DECLARED IN SEGMENT 76 AT 07504000  --  OCCURS AT 07528000
        07507000  07508000
GOOFUP  --  LABEL  --  DECLARED IN SEGMENT 84 AT 07659000  --  OCCURS AT 07682000
        07664000  07666000  07670000  07672500  07679000
GOSTMT  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 07501000
        07530000  07752000
GOTOG  --  BOOLEAN  --  DECLARED IN SEGMENT 3 AT 01467000
       *06397000**07528000* 07529000 *10963000*
GOTOG  --  BOOLEAN  --  DECLARED IN SEGMENT 81 AT 07646395
       *07646515* 07646530  07646540
GOTOS  --  PROCEDURE  --  DECLARED IN SEGMENT 157 AT 16249000
        16270000  16454000  16487000
GOTOSOLVER  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01572000
        06380000
GOTSCHK  --  LABEL  --  DECLARED IN SEGMENT 143 AT 14016000  --  OCCURS AT 14160000
        14148000  14156000  14157000  14158000  14168000
GOTSTORAGE  --  BOOLEAN  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 13733000
        13731000  13732000  13750000  13753000  13759000
GOTSTORAGE  --  BOOLEAN  --  DECLARED IN SEGMENT 143 AT 14025000
       *14059000**14160000* 14537000  14539000 *14546000* 14555000  14560000  14564000
GOV  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01232000
        07550000  07556000  07646515
GRABTOG  --  BOOLEAN  --  DECLARED IN SEGMENT 92 AT 08385600
       *08418350* 08418800
GS  --  INTEGER  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 05325010
        05325460  05325470  05325480
GS  --  INTEGER  --  DECLARED IN SEGMENT 55 AT 05344000
       *05348000* 05349000 *05350000**05357000**05361000**05367000**05371000**05373000* 05375000 *05376000* 05376100
GTA1  --  REAL ARRAY  --  DECLARED IN SEGMENT 3 AT 01697000
        13042000  13045000 *13080000* 13193000  13210000 *13351000* 13454000  13551500  13552000  13716000  13720000
        13722000  13724000  13778000 *14069000**14129000* 14130000  14134000  14161000  14167000  14259000  14259010
       *14259015* 14278000  14281000  14295100
GTIX  --  INTEGER  --  DECLARED IN SEGMENT 3 AT 01339050
GTI1  --  INTEGER  --  DECLARED IN SEGMENT 3 AT 01384500
       *02013526**02013528* 02013530  02013535 *02013567* 02013568 *02013665**02013670* 02013697  02013699 *02013705*
       *02192500**02193000* 02193250 *06073845**06073850**09361020**09361040**09361080**09361120**09361140**09361160*
       *09361180**09361182**09361200* 09405200  09419000
GTR  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01650000
        06072080  10960000
GTR  --  DEFINE  --  DECLARED IN SEGMENT 167 AT 17002000
```

```
        17079000
GT1  --  REAL  --  DECLARED IN SEGMENT 3 AT 01384000
       *02013235* 02013250  02013255 *02013275* 02013295 *02013360* 02013410  02013515 *02013665* 02013675  02013716
       *02203500* 02205000 *02452000* 02456000 *02652000* 02653000  02655000 *02658000* 02659000  02716000 *02719000*
        02720000 *02723500* 02726000 *02730000**02823000* 02830000 *02832000* 02833000  02834000  02838000  02839000
       *02867000* 02869000 *02875000* 02876000  02877010  02878000  02879000  02881000  02882000  02882100 *02886000*
        02894000 *02897000* 02900000 *04105000* 04107000  04109000  04110000 *04120100**04120500* 04121000  04122000
       *04508000* 04521000  04523000  04524000 *05116000* 05117000 *05231000* 05233000 *05235000**05391000* 05392000
       *05397000* 05398000  05399000 *05404000* 05405000  05406000 *06232000* 06234000 *06322000* 06325000 *07086000*
        07087000 *07089000* 07090000 *07186000**07442000* 07443000 *07444000* 07445000 *07449000**07520000* 07522000
        07523000 *07539000* 07540000  07544000  07545000 *07727060* 07727065 *07727070* 07727080 *07729200**08058000*
        08060000  08493530 *08860200**08942000* 08943000 *09273000**09274000**09291000**09292000* 09296000  09298000
       *09305000**09311000**09349000* 09350000  09352000  09354000  09357000  09359000  09360000 *09394000* 09395000
        09396000  09397000 *09407000* 09407010  09415000  09417000 *10167900**10250000* 10251000 *10304000* 10305000
        10306000 *10926500**10926510* 10926520 *10974000* 10975000 *10982000* 10983000 *13157000**13160000**13163000*
       *13167000**13226000* 13232000  13240000  13244000  13248000 *13251000**13252000**13256000* 13259000 *13273000*
        13274000 *13280000* 13282000 *13371000* 13373000 *13780000**13911300**14099000* 14116000 *14116300* 14116500
       *14333100**14425000* 14427000  14429000  14430000  14437000  14447000 *14451100* 14451700 *15102000* 15106000
        15107000 *15281000* 15283000  15284000  15285000 *16159000* 16163000  16164000  16166000  16171000  16173000
       *16254000* 16255000  16257000  16258000  16260000  16265000 *16283000* 16286000  16288000 *16452000*
GT1  --  INTEGER  --  DECLARED IN SEGMENT 146 AT 14203000
       *14206000**14213000* 14223000  14228000 *14231000**14232000* 14233000 *14240000**14241000* 14245000
GT2  --  REAL  --  DECLARED IN SEGMENT 3 AT 01384000
       *02013255**02013260* 02013295 *02013410* 02013500 *02013670* 02013675  02013699 *02721500**02722500* 02726000
       *02834000**02839000**02867000* 02869000 *02876000* 02877010  02878000  02879000  02881000  02882000 *05233000*
        05234000 *07442000* 07443000  07448000 *07520000* 07522000 *07539000* 07545000 *07601000* 07605000  07607000
        07608000 *07609000**07727055**07727060**09214520* 09214530 *09292000* 09293000  09294000 *09296000* 09296100
        09301000  09311000 *09313000**09407010* 09416000  09417000 *13072000**13196600**13284000* 13285000  13286000
       *14430000* 14431000 *14446000* 14448000 *16163000* 16167000  16172000 *16254000* 16255000  16266000
GT2  --  INTEGER  --  DECLARED IN SEGMENT 146 AT 14203000
       *14212000* 14231000
GT3  --  REAL  --  DECLARED IN SEGMENT 3 AT 01384000
       *02013245**02013675* 02013680 *02826000* 02827000  02842000 *07442000* 07454000 *07601000* 07610000 *09291000*
       *09300100* 09300200 *09305000**13285000* 13287000 *14438000*
GT4  --  REAL  --  DECLARED IN SEGMENT 3 AT 01384000
        02904000 *04053000**04059000* 04060000  04063000 *04065000* 04066000 *06153000* 06157000 *07443000* 07444000
       *07448000* 07449000 *07523000* 07524000 *07601000* 07603000  07610000  07614000  07622000  07627000 *13235000*
        13238000
GT4  --  INTEGER  --  DECLARED IN SEGMENT 146 AT 14203000
       *14213000* 14249000
GT5  --  REAL  --  DECLARED IN SEGMENT 3 AT 01384000
       *07028000* 07029000 *07607000* 07609000 *08860200* 08860400  08860450 *13261000**13779000* 13814000  13818000
GT5  --  INTEGER  --  DECLARED IN SEGMENT 146 AT 14203000
       *14214000* 14248000 *01828000* 01833000  01835000
H  --  REAL  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 05315000
        05316000
HANDLESWLIST  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 13196300
        13196610  14168000
HANDLFTWETAILENDOFAREADORSPACESTATEMENT  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 08233000  --  FORWARD AT 03084000
        07949000  08491000  08525000
HEXIZF  --  BOOLEAN STREAM PROCEDURE  --  DECLARED IN SEGMENT 3 AT 02001858
       *02001896* 02001898  02774000
HF  --  LABEL  --  DECLARED IN SEGMENT 143 AT 14017000  --  OCCURS AT 14382000
        14123000  14508000
HOLD  --  STREAM VARIABLE  --  DECLARED IN SEGMENT 3 AT 02013073
        02013080  02013085 *02013095* 02013135
HOLD  --  INTEGER  --  DECLARED IN SEGMENT 92 AT 08385800
```

```
       *08401045* 08450200
HOLD  --  INTEGER  --  DECLARED IN SEGMENT 97 AT 08591700
       *08597450* 08658200
HOLE  --  REAL  --  DECLARED IN SEGMENT 71 AT 07393000
       *07396000* 07398000  07410000  07412000  07415000  07419000  07420000  07422000
HTTE0AP  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 13731000
        13773000  14489000  14555000
HVCHECK  --  BOOLEAN PROCEDURE  --  DECLARED IN SEGMENT 3 AT 05389000
       *05390100**05393000* 08914000  08988000
HVS1  --  STREAM PROCEDURE  --  DECLARED IN SEGMENT 170 AT 17036000
        17042600
HV1  --  PROCEDURE  --  DECLARED IN SEGMENT 170 AT 17042400
        17045000
HV2  --  PROCEDURE  --  DECLARED IN SEGMENT 172 AT 17112000
        17119000
I  --  INTEGER  --  DECLARED IN SEGMENT 3 AT 01319000
       *02917000**05002000**05003000**05107000* 05139000  05140000  05145000  05168000  05187560  05187570  05189000
        05216000  05217000  05218000  05221000  05222000  05224000 *05225000* 05273300  05273400  05274100  05274200
        05280000  05281000  05294000  06016000  06044000  06062000  06062135  06062150  06062350  06062380  06110000
        06117000  06120100  06184000  06232000  06256000  06284400  06286200  06287000  06287800 *06288600* 06313650
        06313700  06314150  06314200  06315500  06322000  06330700  06395000 *07008000**07031000* 07079000  07095000
       *07113000* 07157000  07185000  07186000  07198000 *07247000**07250000**07256000**07260000**07283000**07374000*
        07396000  07401000  07430000  07442000  07463500  07465000  07468250  07508000  07509000  07511000  07554000
       *07556000* 07566000  07572000  07583000  07599000 *07646515* 07662500  07663500  07670500  07687500  07689500
       *07691000* 07732000 *07733000**07734000**07735000**07755000* 07812000  07816000  07837000  07859050  07923000
        07941000  07947000  08017900  08029000  08034000  08037000 *08110000* 08111000 *08120000**08174000* 08176000
       *08223000* 08401030 *08437050* 08442000  08460000 *08482000* 08484500  08488000 *08489520* 08489550 *08493530*
        08597300 *08653110**08653140**08653210* 08668000  08688500  08692000 *08695500* 08696520  08728400 *08732100*
        08739000 *08795100* 08809000  08860600  08860650 *08860700* 08909000  08910000 *08911000* 08914000  08915000
        08916000  08917000 *08968000* 08969000  08970000 *08974000* 08975000  08976000 *08985000* 08988000  08989000
        08991000  08992000 *09348000* 09349000  10304000  10308100  10309000  10318000 *10354000* 10568000 *10569000*
        10573000  10585000  10588000  10592000  10598000 *10638000* 10644000  10649000  10656010  10658000  10795000
        10796000  10804000 *10869000* 10911000  10926500  10926510  10929000  10930000  10965000  10967000 *10987000*
        12015000  12020000  12024000  12036000  12036200  12044000  12060600  12061000  12061725  12062600  12062800
        12064400 *13074000* 13086000 *13088105* 13200000  13310350 *13324000* 13348200  13381000  13372000 *13476000*
        13480000  13483000 *13499000**13504000**13511000**13513000* 13517000 *13521000* 13547000 *13567000**13569000*
        13572000  13575000 *13802000**13815000* 13816000  13819400  13911100  13911400  14062000  14063000 *14067000*
        14129000 *14171000**14173000**14189000**14255000**14264000**14269140* 14269446  14269448  14269520  14269540
        14314000 *14332000* 14349000  14352000  14353000 *14360000* 14399000 *14408000* 14480000  14507050 *14507070*
        14507080  14507090  14507100  14507105 *14507110**14507160* 14516000  14519000  15077000  15254500  16121000
        16124000  16159000  16200000  16204000  16209000  16211100  16211300  16212200  16212500  16212600  16212700
        16214000  16225000 *16252000* 16254000  16283000  16321000  16329000  16342000  16344000  16370000  16376000
        16379500  16408000  16412000  16435000 *16449000**16450000* 16479000 *16489000**17004700* 17004710
I  --  INTEGER  --  DECLARED IN SEGMENT 29 AT 02927130
I  --  INTEGER  --  DECLARED IN SEGMENT 41 AT 04237000
       *04239000* 04241000 *04242000* 04243000
I  --  REAL  --  DECLARED IN SEGMENT 68 AT 06501000
       *06506000**06512000*
I  --  REAL  --  DECLARED IN SEGMENT 93 AT 08493030
       *08493070**08493140**08493160* 08493170
I  --  DEFINE  --  DECLARED IN SEGMENT 120 AT 10258200
        10277000  10282000
I  --  INTEGER  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 167 AT 17002020
        17002015  17002020  17002040
I  --  INTEGER  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 167 AT 17002055
        17002050  17002055  17002075
I  --  INTEGER  --  DECLARED IN SEGMENT 173 AT 17054400
```

```
        *17054600* 17054800  17055010
I   --  DEFINE  --  DECLARED IN SEGMENT 175 AT 17073100
        17079000  17080000
I   --  REAL  --  DECLARED IN SEGMENT 177 AT 17091120
        *17094900* 17095100  17095200  17095300  17095310
ID  --  REAL  --  DECLARED IN SEGMENT 20 AT 02310000
        *02312000* 02313000
ID  --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 54 AT 05325040
        05325320
IDARRAY  --  ALPHA ARRAY  --  DECLARED IN SEGMENT 3 AT 01006000
        08948000  09026000  09395000  09395500 *09396000**09397000**09399000**09399100**09400000**09402000**09403000*
        *09405000**09405200* 09407000  13101100
IDBLDR  --  STREAM LABEL  --  DECLARED IN SEGMENT 10 AT 02091500  --  OCCURS AT 02110500
        02095000  02103000
IDENT  --  LABEL  --  DECLARED IN SEGMENT 27 AT 02639000  --  OCCURS AT 02865000
        02643000
IDLOC  --  INTEGER  --  DECLARED IN SEGMENT 3 AT 01341000
        08948000  08949000 *09026000* 09027000  09394000  13101100  13102000
IDLOC  --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 103 AT 08932000
        08934000  08938000
IDLOC  --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 131 AT 13020000
        13023000  13035000
IDLOCTEMP  --  INTEGER  --  DECLARED IN SEGMENT 3 AT 01341000
        *09027000* 09394000
IDMAX  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01479000
        07099000  07215000  07661000  07689500  13088010  13088060  13337000  13911300  17047100  17095100  17095200
IDNOF  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01007410
        02001660  02001795  17025400  17042350  17091900  17093850  17094050  17094100
IDTYPE  --  REAL ARRAY  --  DECLARED IN SEGMENT 172 AT 17047100
        *17084010* 17094900
IDV  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01651000
IFCLAUSE  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 06409000  --  FORWARD AT 03018000
        06296000  07563000
IFEXP  --  INTEGER PROCEDURE  --  DECLARED IN SEGMENT 3 AT 06294000  --  FORWARD AT 03013000
        06005000  06143000 *06299000* 06311000
IFS  --  PROCEDURE  --  DECLARED IN SEGMENT 157 AT 16191000
        16239000  16486000
IFSB  --  LABEL  --  DECLARED IN SEGMENT 162 AT 16193000  --  OCCURS AT 16216000
        16194000
IFSC  --  LABEL  --  DECLARED IN SEGMENT 162 AT 16193000  --  OCCURS AT 16207000
        16194000
IFSTMT  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 07561000  --  FORWARD AT 03022000
        07587000  07751000
IFSW  --  SWITCH LABEL  --  DECLARED IN SEGMENT 162 AT 16194000
        16197000
IFTOG  --  LABEL  --  DECLARED IN SEGMENT 162 AT 16193000  --  OCCURS AT 16217000
        16194000  16215000
IFV  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01231000
        06004000  06141000  16211100  16212200  16222000
II  --  INTEGER  --  DECLARED IN SEGMENT 152 AT 14507030
       *14507050* 14507160
IMPFUN  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 06057000  --  FORWARD AT 03039000
        06074000  06098000
IMPOP  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01272000
        06196000
INC  --  DEFINE  --  DECLARED IN SEGMENT 124 AT 10516000
        10584000  10586000  10587000  10658000  10660000  10661000
INC  --  DEFINE  --  DECLARED IN SEGMENT 156 AT 15192000
```

```
        15244000  15245000  15246000  15250000
INCLUDECARD  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 02238100
        02571000
INCR  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01153000
        02658000  02705200  02776100  05124000  05189000  05222000  05382000  07152000  07369000  10585000  10659000
        13206000  13240000  13307000  13559000  13560000  14098000  14116000
INCRF  --  REAL  --  DECLARED IN SEGMENT 3 AT 01607000
       *13206000* 14214000  14314000
INDEC  --  LABEL  --  DECLARED IN SEGMENT 143 AT 14014000  --  OCCURS AT 14158000
        14019000
INDEX  --  REAL  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 02001650
        02001640  02001645  02001660  02001695
INDEX  --  REAL  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 02001750
        02001740  02001745  02001795  02001805  02001810  02001815  02001820  02001822
INDEX  --  INTEGER  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 05005000
        05006000
INDEX  --  REAL  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 05008000
        05009000
INDEX  --  INTEGER  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 07048000
        07045000  07046000  07086000  07152000  07241000  07369000
INDEX  --  DEFINE  --  DECLARED IN SEGMENT 79 AT 07595000
        07601000  07610000
INDEX  --  DEFINE  --  DECLARED IN SEGMENT 153 AT 15028000
        15034000  15035000
INDEX  --  INTEGER  --  DECLARED IN SEGMENT 164 AT 16314000
       *16320000**16328000**16334000* 16346000  16349000
INDEXS  --  PROCEDURE  --  DECLARED IN SEGMENT 157 AT 16311000
        16355000  16490000
INF  --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 130 AT 12154000
        12157000
INFO  --  REAL ARRAY  --  DECLARED IN SEGMENT 3 AT 01007000
        02001805  02013517  02013520  02013525  02214250  02214260  02226000  02234750  02234900  02238582  02265600
        02265700  02265850  02265950  02368000  02411000  02486000  02761500  02867000  02869000  02877010  02878000
        02879000  02881000  02923000  04172000  04187000  04209000 *04210000**04211000* 04221000 *04223000* 05006000
       *05009000* 05039500  05045000  05058000 *05233000* 05325470  07727055 *08017800**08035000* 09034500 *09085000*
       *09214105* 09292000  09296000  09298000  09300200  09305000  09415000  09416000 *10967000* 10974000  10982000
       *10984000**10989000* 10997000  12162000  12163000  13102000  13281000  13310000 *13593000* 14440000  14507040
        14507090  14507100  14507105  14507170  15034000 *16288000**17004710**17025100**17025300* 17080000
INFO  --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 3 AT 02005000
        02006000
INFO  --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 121 AT 10236000
        10238000
INFOINDEX  --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 3 AT 05317000
        05324000
INPRO  --  BOOLEAN  --  DECLARED IN SEGMENT 103 AT 08929000
       *08975000* 08979000
INPROCHECK  --  BOOLEAN PROCEDURE  --  DECLARED IN SEGMENT 103 AT 08940000
       *08941100**08944000* 08975000
INPUT1  --  BOOLEAN PROCEDURE  --  DECLARED IN SEGMENT 170 AT 17006000
       *17012000* 17045000
INPUT2  --  BOOLEAN PROCEDURE  --  DECLARED IN SEGMENT 172 AT 17070000
       *17082000* 17119000
INSERTCOP  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01561520
        02193800  02238290  02238514  02238570  09274100
INSERTDEPTH  --  INTEGER  --  DECLARED IN SEGMENT 3 AT 01561570
        02193800  02224030  02224032  02224040 *02224090* 02224110 *02238280* 02238290  02238300  02238385  02238440
        02238514  02238550  02238552  02238570  02238574  02238602  02238610  02238620  02238630  02238640  02238650
```

```
        02238654  02238656  02238660  02238690  02238704  02238710 *02238780* 02238810  09274100
INSERTFID  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01561540
        02224110  02238290  02238552  02238610  02238620  02238630  02238650  02238656  02238810
INSERTFILE  --  DEFINE  --  DECLARED IN SEGMENT 124 AT 10535000
        10583000  10657000
INSERTINFO  --  REAL ARRAY  --  DECLARED IN SEGMENT 3 AT 01561580
        02193800 *02224030* 02224032  02224040  02224110 *02238290**02238300* 02238385  02238440 *02238514**02238550*
       *02238552* 02238570  02238574  02238602 *02238610* 02238620 *02238630* 02238650 *02238654**02238656* 02238660
        02238690 *02238704* 02238810 *09274100*
INSERTINX  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01561550
        02224030  02224032  02238290  02238385  02238660  02238690  02238704
INSERTMAX  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01561510
        01561580  02238280
INSERTMID  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01561530
        02224110  02238290  02238550  02238602  02238630  02238650  02238654  02238810
INSERTSEQ  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01561560
        02224040  02238300  02238440
INT  --  OWN BOOLEAN  --  DECLARED IN SEGMENT 89 AT 08012000
       *08060000* 08070000
INTARRAYID  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01205000
        04522000  05381000  06062290  06073450  07087000  07100000  07151000  07292000  07464250  07684500  07814100
        07834000  07939000  08057000  08401000  08457000  08597100  08665000  08728100  10311000  10579000  10805000
        12034100  12036400  12062255  13330550  13346000  14094000  15316000  15333000
INTBIT  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01000960
        02577000
INTEGERDEC  --  LABEL  --  DECLARED IN SEGMENT 143 AT 14013000  --  OCCURS AT 14141000
        14018000
INTERPTI  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01579000
        08287200
INTERPTO  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01577000
        08698900
INTID  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01201000
        04509000  06062290  06062340  06286800  08038000  08061000  10307000  10574000  10599000  10872000  10926510
        12034100  12035000  12042000  12062250  14141000  15086000  15102000
INTOG  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01001250
INTPROCID  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01197000
        06062295  07282000  07330000  07463000  10654000  12034110  12036100  12062250  13278000  14295000  15078000
        15101000
INTRNSICPROCID  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01188000
        10599000
INTSTRPROCID  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01193000
        14284000  14294000
INTV  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01285000
        06120100
INV  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01290000
INX  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01652000
        04531000  05179000  08957000  08980000
IOBUFFSIZE  --  INTEGER  --  DECLARED IN SEGMENT 3 AT 01340070
        09361160 *13184000*
IODEC  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 13001000
        14157000  14158000  14159000
IOT  --  INTEGER  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 13003000
        13001000  13002000  13153000  13157040  13159000
IOTEMP  --  INTEGER  --  DECLARED IN SEGMENT 131 AT 13039550
       *13113000**13148200**13148300* 13157030 *13157040* 13184000
IOTEMPO  --  INTEGER  --  DECLARED IN SEGMENT 131 AT 13039550
       *13157020* 13157030  13157040
IPART  --  LABEL  --  DECLARED IN SEGMENT 27 AT 02639000  --  OCCURS AT 02758000
```

```
        02643000
IS  --  LABEL  --  DECLARED IN SEGMENT 43 AT 04501000  --  OCCURS AT 04548000
        04527000  04540000
ISD  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01653000
        06062435  06399000  07519000  12038000  12063300  15103000  15334000
ISKIP  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 121 AT 10236000
        10237000  10238000  06072000  06120200  08413040  08414022  08415030  08416022  08418300  08621010  08625000
        08627000  08630000  13501000  13526000  15103000  15279000  15317000  15334000
J  --  REAL  --  DECLARED IN SEGMENT 3 AT 01618000
       *13042000**13045000**13061000**13067000* 13072000 *13077000**13080000* 13193000 *1319649O**13196550* 13196600
       *13210000**13324000**13349000* 13351000 *13454000**13716000**13720000**13722000**13724000* 13778000 *13904000*
       *13907000* 13919000 *14069000**14129000* 14130000  14132000  14134000 *14135000**14136000**14136100**14137000*
       *14157000* 14161000  14167000 *14278000**14281000**14295100**14421000* 14425000  14438000  14440000  14445100
        14446000  14448000 *14449000*
J  --  INTEGER  --  DECLARED IN SEGMENT 29 AT 02927130
       *02927490* 02927510  02927530  02927540  02927570  02927580  02927610  02927612  02927620  02927660  02927690
        02927700
J  --  INTEGER  --  DECLARED IN SEGMENT 39 AT 04166000
       *04170000* 04172000  04179000  04187000
J  --  DEFINE  --  DECLARED IN SEGMENT 83 AT 07656000
        07696500  07697000
J  --  REAL  --  DECLARED IN SEGMENT 86 AT 07802000
       *07904000**07912000*
J  --  INTEGER  --  DECLARED IN SEGMENT 102 AT 08902000
       *08919000**08921000* 08922000  08923000
J  --  INTEGER  --  DECLARED IN SEGMENT 103 AT 08930000
       *08947000* 08949000  08950000
J  --  REAL  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 10228000
        10259600 *10259700* 10264600  10264650
J  --  REAL  --  DECLARED IN SEGMENT 129 AT 12102000
       *12116000*
J  --  INTEGER  --  DECLARED IN SEGMENT 130 AT 12153000
       *12163000* 12165000
J  --  REAL  --  DECLARED IN SEGMENT 134 AT 13221000
       *13232000* 13239000 *13257000**13258000* 13261000  13264000 *13265000* 13270000 *13271000*
J  --  REAL  --  DECLARED IN SEGMENT 147 AT 14254100
       *14259010**14259015**14259075* 14259080  14259085  14266000
J  --  REAL  --  DECLARED IN SEGMENT 154 AT 15075000
       *15089010* 15101000 *15234000* 15255000  15266000  15273100 *15280000* 15281000  15290000  15291100  15296000
        15303000  15321000  15344000  15350000
J  --  REAL  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 156 AT 15217000
        15215000  15216000  15218000
J  --  INTEGER  --  DECLARED IN SEGMENT 164 AT 16314000
       *16348000**16349000*
J  --  INTEGER  --  DECLARED IN SEGMENT 165 AT 16364000
       *16387000* 16389000 *16392000* 16393000
J  --  INTEGER  --  DECLARED IN SEGMENT 168 AT 17002030
       *17002035* 17002040
J  --  INTEGER  --  DECLARED IN SEGMENT 169 AT 17002065
       *17002070* 17002075
J  --  INTEGER  --  DECLARED IN SEGMENT 173 AT 17054400
       *17054900* 17055010  17055020
JEDEN  --  INTEGER  --  DECLARED IN SEGMENT 17 AT 02264100
       *02265000* 02265100 *02265350* 02265450  02265600  02265700  02265850  02265950
JFC  --  DEFINE  --  DECLARED IN SEGMENT 157 AT 16016000
         16219000
JFW  --  DEFINE  --  DECLARED IN SEGMENT 157 AT 16007000
```

```
        16043000  16045000  16079000  16232000
JNS  --  DEFINE  --  DECLARED IN SEGMENT 157 AT 16023000
        16453000
JOINFO  --  REAL  --  DECLARED IN SEGMENT 3 AT 01482000
        16127000 *16128000* 16134000  16138000  16139000 *16143000* 16440000 *16442000* 16445000  16447000  16449000
JOINT  --  REAL  --  DECLARED IN SEGMENT 161 AT 16118000
       *16127000* 16143000
JRV  --  DEFINE  --  DECLARED IN SEGMENT 157 AT 16009000
        16079000
JUMPCHAIN  --  PROCEDURE  --  DECLARED IN SEGMENT 157 AT 16085000
        16097000  16139000  16173000
JUMPCHKNX  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 13616000  --  FORWARD AT 03058000
        10615000  10800000  14222000  14276500  14387000  14486000
JUMPCHKX  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 13596000  --  FORWARD AT 03059000
        10616500  10907000  13041000  13196340  13904000  14156000  14173000  14227000  14506000
JUMPCTR  --  INTEGER  --  DECLARED IN SEGMENT 3 AT 01459000
        06359000 *06360000* 13739000  14497000 *14499000* 14537000 *14603000*
JUMPLACE  --  INTEGER  --  DECLARED IN SEGMENT 123 AT 10346000
       *10347000* 10366000
JUMPLEVEL  --  REAL  --  DECLARED IN SEGMENT 3 AT 01485000
        16260000  16265000 *16269000**16433000**16435000* 16452000
JUMPS  --  PROCEDURE  --  DECLARED IN SEGMENT 157 AT 16431000
        16455000  16493000
JUMPV  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01242000
JUNK  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01430000
        06072000  06072070  06072090  06120200  06285800  06288800  06399000  07111310  07134000  07418000  07419000
        07519000  07946000  08413040  08414022  08415030  08416022  08418300  08621010  08625000  08627000  08630000
        10328000  10837000  10851000  10854000  10959000  10960000  13525000  15007000  15222000  15230000  15279000
        15320000  15330000
K  --  REAL  --  DECLARED IN SEGMENT 3 AT 01693000
       *14085000*
K  --  STREAM LABEL  --  DECLARED IN SEGMENT 10 AT 02091500  --  OCCURS AT 02099000
        02099000
K  --  INTEGER  --  DECLARED IN SEGMENT 29 AT 02927130
       *02927510* 02927530  02927540 *02927600* 02927610 *02927660* 02927690  02927700
K  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 55 AT 05341000
        05342000
K  --  DEFINE  --  DECLARED IN SEGMENT 89 AT 08016000
        08017800  08017900  08033000  08034000  08035000  08094000  08180000  08223000  08232000
K  --  INTEGER  --  DECLARED IN SEGMENT 102 AT 08902000
       *08921000*
K  --  REAL  --  DECLARED IN SEGMENT 120 AT 10258100
       *10263200* 10263800 *10264200**10264400*
K  --  REAL  --  DECLARED IN SEGMENT 126 AT 10957000
       *10972000* 10974000 *10979000* 10982000  10984000 *10996000* 10997000
K  --  REAL  --  DECLARED IN SEGMENT 129 AT 12102000
       *12123500* 12125000  12126000
K  --  REAL  --  DECLARED IN SEGMENT 131 AT 13039000
       *13077000* 13080000
K  --  REAL  --  DECLARED IN SEGMENT 138 AT 13449000
       *13559000* 13563000
K  --  REAL  --  DECLARED IN SEGMENT 140 AT 13737000
       *13750000**13764000**13765000*
K  --  REAL  --  DECLARED IN SEGMENT 147 AT 14254100
       *14259010* 14259040
K  --  INTEGER  --  DECLARED IN SEGMENT 165 AT 16364000
       *16389000*
K  --  INTEGER  --  DECLARED IN SEGMENT 173 AT 17054400
```

```
      *17055010**17055020* 17055025
KEY   --  REAL  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 04231000
       04239000  04241000
KLASSF  --  REAL  --  DECLARED IN SEGMENT 3 AT 01603000
      *13201000**13226000* 13246000  13253000  13278000 *13349000* 13372000 *13913000* 14308000 *14507130*
KLUDGF  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 07930000
       07954000  08450200  08658200  13767400
KNOT  --  BOOLEAN  --  DECLARED IN SEGMENT 34 AT 02973500
      *02974500* 02978500
KOUNT  --  INTEGER  --  DECLARED IN SEGMENT 3 AT 01583000
       13101100  13103000 *13306000*
L  --  INTEGER  --  DECLARED IN SEGMENT 3 AT 01428000
       02195000  02195750  02214200  02227500  02228750  02236250  04050000 *04052000* 04053000 *04056000* 04057000
      *04074000* 04086000  04102000 *04105000**04109000* 04119000 *04120000**04124000* 04145000  04147000  04149000
      *04151000**04172000* 04175000  04178000  04182000 *04185000**04186000* 04210000 *04216000* 04218000  04223000
       04279200  04280000  04284200  04285000  04293000  04296000 *04298000**05167000**06154000**06207000**06289600*
      *06297000**06300000* 06301000  06307000 *06308000* 07025020  07108500 *07111230**07111320* 07111325  07111330
       07130000  07138000  07194000  07270000  07272000  07305000  07312000 *07411100**07459000* 07485000 *07487000*
      *07493000**07495000* 07496000 *07522000* 07523000  07524000 *07540000* 07545000 *07577000* 07579000 *07583000*
      *07585000* 07586000  07608000 *07609500* 07609700  07610000  07646460  07646510 *07646533**07646543* 07646545
       07646570  07646705  07646730 *07646780* 07646800  07727065 *07727070* 07727080 *07859040**07936000**07950000*
       08046000 *08047000**08050000* 08077000 *08078000**08089000* 08093000  08097000 *08106000* 08122000  08124000
       08127000 *08146000* 08147000 *08148000**08150000* 08152000  08160000 *08168000* 08169000  08170000  08171000
       08179000 *08190000* 08191000  08215000 *08221000* 08227000  08228000 *08231000* 08401045  08597450 *08728300*
      *08737000**08747000**08800000**08817000**08857000**08858000**08914000**08918000**08923000**09306100* 09308000
      *10301000**10330000* 10348000  10350000  10366000  10616000 *10801000**10802000* 10906000 *10922000**10926000*
      *10929000* 10958000 *10960000**10971000**10978000* 10981000  10989000  10994000 *10997000* 10999000 *11000000*
      *12000000**13055000**13072000**13088090* 13113000  13148200  13157020 *13196420**13196600**13232000* 13235000
      *13238000**13239000* 13258000 *13263000**13264000* 13267000 *13270000* 13273000 *13277000* 13479000  13509000
      *13529000* 13606000  13612000 *13628000* 13752000 *13768000* 13769000  14054000 *14055200* 14222000  14387000
      *14451100* 14451200  14451300 *14451500**14451700* 14487000  14563000 *14570000* 14571000  14577000 *14595000*
       14601000  15031000 *15033000* 15037000  16032000 *16034000**16038000**16040000**16047000* 16049000  16070000
       16088000 *16089000* 16090000  16092000 *16094000**16096000* 16121000  16138000  16167000  16172000  16218000
       16232000  16257000  16287000
L  --  REAL  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 01719000
L  --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 3 AT 01742100
       01742110
L  --  STREAM LABEL  --  DECLARED IN SEGMENT 3 AT 01758000  --  OCCURS AT 01764000
       01762000
L  --  STREAM LABEL  --  DECLARED IN SEGMENT 10 AT 02091500  --  OCCURS AT 02101000
       02101000
L  --  LABEL  --  DECLARED IN SEGMENT 10 AT 02127500  --  OCCURS AT 02128000
       02135500
L  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 02183500
       02183750  02186500
L  --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 12 AT 02196530
       02196550
L  --  REAL  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 03020000
L  --  INTEGER  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 03051001
       03051000
L  --  LABEL  --  DECLARED IN SEGMENT 41 AT 04232000  --  OCCURS AT 04243000
       04241000
L  --  REAL  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 05123000
       05124000
L  --  REAL  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 05128000
       05131000  05132000
L  --  REAL  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 05307000
```

```
          05312000
L  --  INTEGER  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 05333000
          05331000  05376100
L  --  LABEL  --  DECLARED IN SEGMENT 137 AT 13360000  --  OCCURS AT 13364000
          13366000
L  --  LABEL  --  DECLARED IN SEGMENT 141 AT 13776000  --  OCCURS AT 13780000
          13815000
L  --  LABEL  --  DECLARED IN SEGMENT 164 AT 16313000  --  OCCURS AT 16347000
          16319000  16340000
L  --  INTEGER  --  DECLARED IN SEGMENT 165 AT 16364000
          *16391000*  16393000
L  --  LABEL  --  DECLARED IN SEGMENT 157 AT 16475000  --  OCCURS AT 16483000
          16476000
L  --  LABEL  --  DECLARED IN SEGMENT 171 AT 17009000  --  OCCURS AT 17014000
          17011000
L  --  LABEL  --  DECLARED IN SEGMENT 175 AT 17073000  --  OCCURS AT 17085000
          17081000
LAB  --  LABEL  --  DECLARED IN SEGMENT 85 AT 07716000  --  OCCURS AT 07739000
          07723000
LABELBAT  --  REAL  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 03025000
          03023000  03024000
LABELBAT  --  REAL  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 07537000
          07535000  07536000  07539000
LABELCTR  --  REAL ARRAY  --  DECLARED IN SEGMENT 125 AT 10738000
          *10880000*  10884000  *10913000*  10919000
LABELCTRINX  --  DEFINE  --  DECLARED IN SEGMENT 125 AT 10747000
          10802000  10880000  10885000  10894000  10913000  10919000  10921000
LABELDEC  --  LABEL  --  DECLARED IN SEGMENT 143 AT 14014000  --  OCCURS AT 14187000
          14019000
LABELID  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01206000
          06062390  06393000  07099000  07146000  07215000  07506000  07553000  07645000  07661000  07689500  10642000
          10876000  10908000  10965000  12063000  13088010  13088060  13246000  13337000  13911300  14194000  17047100
          17095100  17095200
LABELR  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 07593000
          07646000  07739000
LABELS  --  PROCEDURE  --  DECLARED IN SEGMENT 157 AT 16156000
          16177000  16485000
LABELV  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01286000
          14399000
LABLMONFILE  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01535000
          07614000
LAMPER  --  LABEL  --  DECLARED IN SEGMENT 58 AT 06090000  --  OCCURS AT 06128000
          06092020  06107000  06110000  06113000
LAST  --  LABEL  --  DECLARED IN SEGMENT 156 AT 15184000  --  OCCURS AT 15303000
          15342500
LASTADDRESS  --  INTEGER  --  DECLARED IN SEGMENT 2 AT 00504200
          *02236250*  02236750  *14049100**14593000**17002530*
LASTADDRESS  --  REAL  --  DECLARED IN SEGMENT 172 AT 17069600
          *17079000*  17080000  *17084000*  17091800  17095750  *17096850*
LASTCRDPATCH  --  BOOLEAN  --  DECLARED IN SEGMENT 3 AT 01001500
          *02208000*  02221250  *02221750*
LASTELCLASS  --  INTEGER  --  DECLARED IN SEGMENT 3 AT 01329100
          *02934000*  08860500  10124200  10134100
LASTENTRY  --  REAL  --  DECLARED IN SEGMENT 3 AT 01376000
          04170000  *04189000*  04208000  04210000  04211000  *04214000*  07609100  07646750  07727050  14276500
LASTGT  --  DEFINE  --  DECLARED IN SEGMENT 151 AT 14420000
          14438000
LASTGT  --  DEFINE  --  DECLARED IN SEGMENT 157 AT 16003000
```

```
        16089000  16266000
LASTINFO  --  REAL  --  DECLARED IN SEGMENT 3 AT 01619000
        02761500  02761501  04178000 *04179000**09033000**09214530**09214980* 10656014  10978500  10978600  13102000
        13103000  13110000  13222000 *13290000* 13309000 *13311000* 13788000  13807000  14079000  14083000  14175000
       *14213000* 14232000  14239000  14240000  14245000 *14249000* 14259017  14259094  14259098  14259100  14259102
        14259155 *14265930* 14269260  14320000 *14330000**14373000* 14421000 *14463000* 16053000 *16055000* 16445000
       *16447000*
LASTINFOT  --  REAL  --  DECLARED IN SEGMENT 143 AT 14029000
       *14330000* 14373000  14463000
LASTRESULT  --  INTEGER  --  DECLARED IN SEGMENT 120 AT 10258000
       *10262000* 10272000 *10276000*
LASTSEQROW  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01576000
        02013517  02013520  02013525  02214250  02214260  02226000  02234750  02234900  02238582  02368000  02411000
        02486000  05039500  05045000  05058000  09033000  09034500  09214980  09415000  09416000
LASTSEQUENCE  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01575000
        02013517  02013520  02013525  02214250  02214260  02226000  02234750  02234900  02238582  02368000  02411000
        02486000  05039500  05045000  05058000  09033000  09034500  09214980  09415000  09416000
LASTUSED  --  INTEGER  --  DECLARED IN SEGMENT 3 AT 01391000
        02013480  02013526  02013533 *02013537**02013540**02013541**02013565* 02013566  02013617  02013621  02013628
        02013714  02194500 *02204000**02209250* 02212000  02212250 *02212750**02224100* 02224500  02229000  02230750
        02232500 *02238720**02332000**02430000* 02437500 *02438000**02458000**02525009**02721000* 02900000 *02901000*
        02908000 *07030000**09029000**09037000*
LBC  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01645000
LBEG  --  LABEL  --  DECLARED IN SEGMENT 85 AT 07718000  --  OCCURS AT 07754000
        07727000
LBJ  --  REAL  --  DECLARED IN SEGMENT 138 AT 13449000
       *13473000**13547000* 13552000  13556000
LBLREF  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01007486
        07598100  16159100  17092300
LBLTOG  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 172 AT 17047200
        17047400  17050900
LBLTOG  --  OWN BOOLEAN  --  DECLARED IN SEGMENT 177 AT 17091110
       *17092500* 17094800 *17095400*
LBP  --  DEFINE  --  DECLARED IN SEGMENT 143 AT 14023000
        14108000
LBRK  --  LABEL  --  DECLARED IN SEGMENT 85 AT 07718000
LBUFF  --  REAL ARRAY  --  DECLARED IN SEGMENT 3 AT 01561610
        02224030  02224040  02224050  02238580  02238582  02238590  02238652  02238680  02238690
LCARD  --  LABEL  --  DECLARED IN SEGMENT 11 AT 02191000  --  OCCURS AT 02194750
        02191250
LCLOSE  --  LABEL  --  DECLARED IN SEGMENT 85 AT 07717000  --  OCCURS AT 07743000
        07725000
LCR  --  INTEGER  --  DECLARED IN SEGMENT 3 AT 01330000
        02013313  02013445  02013516  02013517 *02013621**02013625**02013715* 02013716  02013717  02129500  02193250
       *02204000**02209000* 02213000 *02214000* 02214250  02214500 *02215750**02224050* 02225250  02226000  02232000
        02234750  02235750  02238512  02368000  02369000  02410000  02485000  02716000  02717000 *02719000* 02902000
       *02903000* 02904000  02905000 *07025010**07025030*
LCR  --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 3 AT 01756000
        01759100  01760000  01763000  01767000  01774000  01779000
LCR  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 02005000
        02006000
LCR  --  REAL  --  NAME PARAMETER  --  DECLARED IN SEGMENT 13 AT 02198750
        02198500 *02199750**02200500* 02201000 *02202000*
LCRLINK  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01561110
        02013534  02013535  02013625
LDBL  --  LABEL  --  DECLARED IN SEGMENT 85 AT 07717000  --  OCCURS AT 07746000
        07725000
```

```
LDEC -- LABEL -- DECLARED IN SEGMENT 85 AT 07716000 -- OCCURS AT 07731000
        07724000
LDES -- DEFINE -- DECLARED IN SEGMENT 3 AT 01299000
        07646660  08860300  13784000  14575000  14601000
LDICT -- REAL ARRAY -- DECLARED IN SEGMENT 2 AT 00504600
        09399150 *13652000*
LDO -- LABEL -- DECLARED IN SEGMENT 85 AT 07717000 -- OCCURS AT 07749000
        07725000
LDOT -- LABEL -- DECLARED IN SEGMENT 58 AT 06090000 -- OCCURS AT 06127000
        06098000  06101000  06104000  06120200
LE -- LABEL -- DECLARED IN SEGMENT 61 AT 06224000 -- OCCURS AT 06228000
LEFTPAREN -- DEFINE -- DECLARED IN SEGMENT 3 AT 01212000
        02975000  06062120  06072030  06072200  06073200  06118000  06415000  07112000  07409000  07452000  07462000
        07662500  07829000  07844000  07849000  07859020  07901000  08386000  08395000  08418100  08503000  08593000
        08613100  08722000  08785000  08842000  08905000  08966000  10264100  10570000  10797000  12007000  12060500
        12114000  13104000  13791000  13806000  14174000  14259020  14326000  16125000  16225000  16282000  16480000
LENGTH1 -- LABEL -- DECLARED IN SEGMENT 21 AT 02360000 -- OCCURS AT 02374000
        02364000
LENGTH2 -- LABEL -- DECLARED IN SEGMENT 21 AT 02360000 -- OCCURS AT 02380000
LENGTH3 -- LABEL -- DECLARED IN SEGMENT 21 AT 02360000 -- OCCURS AT 02381000
        02364000
LENGTH4 -- LABEL -- DECLARED IN SEGMENT 21 AT 02360000 -- OCCURS AT 02399000
        02364000
LENGTH5 -- LABEL -- DECLARED IN SEGMENT 21 AT 02361000 -- OCCURS AT 02476000
        02364000
LENGTH6 -- LABEL -- DECLARED IN SEGMENT 21 AT 02361000 -- OCCURS AT 02518000
        02365000
LENGTH7 -- LABEL -- DECLARED IN SEGMENT 21 AT 02361000 -- OCCURS AT 02569000
        02365000
LENGTH8 -- LABEL -- DECLARED IN SEGMENT 21 AT 02361000 -- OCCURS AT 02574000
        02365000
LENGTH9 -- LABEL -- DECLARED IN SEGMENT 21 AT 02361000 -- OCCURS AT 02575000
        02365000
LEQ -- DEFINE -- DECLARED IN SEGMENT 3 AT 01655000
        08050000
LEQ -- DEFINE -- DECLARED IN SEGMENT 167 AT 17002000
LERR -- LABEL -- DECLARED IN SEGMENT 85 AT 07716000 -- OCCURS AT 07730000
        07720000  07723000  07724000
LETTER -- STREAM VARIABLE -- VALUE PARAMETER -- DECLARED IN SEGMENT 119 AT 10100000
        10101000  10105000
LEVEL -- INTEGER -- DECLARED IN SEGMENT 3 AT 01402000
        04611000  05208000  06397000  07926100  10644000  10656010  10965000  13339500  13371000  13739000  13767100
        13767200  13819400  13905000  13912000 *14053000* 14207000  14307000 *14323000* 14477000  14478030  14497000
        14499000 *14500000* 14537000  14557500  14560000  14567000  14569100  14575000  14597000  14603000 *14604000*
        14611000
LEVEL -- DEFINE -- DECLARED IN SEGMENT 157 AT 16005000
        16171000  16265000
LEVELF -- REAL -- DECLARED IN SEGMENT 3 AT 01605000
        06397000  06397300 *13204000* 13339500  13371000 *13911400* 13912000  14207000  14307000
LEXIT -- DEFINE -- DECLARED IN SEGMENT 125 AT 10723000
        10906000  10928000
LF -- DEFINE -- DECLARED IN SEGMENT 3 AT 01561600
        02224030  02224032  02224080  02224110  02238640  02238650  02238652  02238654  02238656  02238680  02238800
        02238810
LFC -- DEFINE -- DECLARED IN SEGMENT 3 AT 01648000
LFILL -- LABEL -- DECLARED IN SEGMENT 85 AT 07717000 -- OCCURS AT 07750000
        07726000
LFOR -- LABEL -- DECLARED IN SEGMENT 85 AT 07717000 -- OCCURS AT 07747000
```

```
        07725000
LFTBRKET  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01263000
        05273800  06314000  06421000  07947000  08240000  08409000  08614000  08653100  08653130  08653200  08698000
        10264100  10295000  10588000  10809000  12114000  13088085  13473000  14259020  14269320  15233000
LFTPAREN  --  DEFINE  --  DECLARED IN SEGMENT 157 AT 16003000
        16125000  16225000  16282000  16480000
LGO  --  LABEL  --  DECLARED IN SEGMENT 85 AT 07718000  --  OCCURS AT 07752000
        07726000
LGTFLD  --  DEFINE  --  DECLARED IN SEGMENT 157 AT 16004000
        16172000
LGTH  --  STREAM VARIABLE  --  DECLARED IN SEGMENT 3 AT 02001070
        02001120  02001160  02001360  02001568
LIB  --  BOOLEAN  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 13 AT 02198500
        02199000
LIB  --  BOOLEAN  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 13 AT 02202500
        02204000  02204250  02207750  02209250
LIBARRAY  --  REAL ARRAY  --  DECLARED IN SEGMENT 3 AT 01561065
       *02013480**02013490**02013495**02013497* 02013505  02013520 *02013530**02013534**02013535**02013537* 02013538
        02013565  02013567  02013580  02013600  02013615  02013620  02013625 *09275350* 09275500
LIBCLAST  --  LABEL  --  DECLARED IN SEGMENT 13 AT 02210500  --  OCCURS AT 02220000
        02211000
LIBEND  --  LABEL  --  DECLARED IN SEGMENT 13 AT 02210500
LIBINDEX  --  INTEGER  --  DECLARED IN SEGMENT 3 AT 01561120
        02013480  02013490  02013495  02013497  02013505  02013520  02013530  02013534  02013535  02013536  02013537
        02013538 *02013542**02013560* 02013565  02013567  02013575  02013580  02013600  02013610  02013615  02013620
        02013625  02013635  02131500  02732000
LIBRARY  --  SWITCH FILE  --  DECLARED IN SEGMENT 3 AT 01561050
        02013245  02013250  02013270  02013275  02013528  02013680  02013685  02013697  02013705  02013710  02013715
        02200250  02200500
LIBRARYFIL  --  FILE  --  DECLARED IN SEGMENT 3 AT 01561590
        02224030  02224032  02224080  02224110  02238640  02238650  02238652  02238654  02238656  02238680  02238800
        02238810
LIBTLAST  --  LABEL  --  DECLARED IN SEGMENT 13 AT 02210500  --  OCCURS AT 02223000
        02211000
LIF  --  LABEL  --  DECLARED IN SEGMENT 85 AT 07717000  --  OCCURS AT 07751000
        07726000
LIN  --  REAL ARRAY  --  DECLARED IN SEGMENT 3 AT 01559010
        02194000  02195000  02195750  02196580  02196590  02214200  02226750  02227000  02227250  02227500  02228750
        04149000  04150000  05038000  05039500  05039600  05044000  05045900  05046000  05264000  05264500  05265000
        05325450  05325460  05325470  05325480  05325490  07025020  09300150  09301000  09302000  10074000  10075000
LIN  --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 3 AT 05237000
        05238500
LIN  --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 54 AT 05325040
        05325090
LINE  --  FILE  --  DECLARED IN SEGMENT 3 AT 01559000
        01829000  01835700  02195000  02195750  02196590  02214200  02227250  02227500  02228750  02264950  02265150
        02265300  02265450  02265650  02265750  02265900  02266000  02464000  02927440  02927460  02927710  02927722
        04150000  04279200  04280000  04284200  04285000  05039600  05046000  05265000  05325490  07025020  09302000
        09407020  09416002  09417000  10075000  13633100  13633200  13649000  13650000  14461500  14493500  17002520
        17054500  17054800  17055020  17091510  17094300  17094400  17094550  17095500  17096300
LINF  --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 3 AT 02183500
        02184250  02186750
LINE  --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 45 AT 05016000
        05019000  05024000
LINECOUNT  --  DEFINE  --  DECLARED IN SEGMENT 177 AT 17091140
        17091520  17094410  17095510  17096350
LINK  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01154000
```

```
          02722500  02726000  02882200  05124000  05132000  05189000  05233000  05234000  06062380  10585000  10656014
          10656015  10659000  12036200  12062800  12123000  13207000  13232000  13286000  13308000  13361000  13362000
          13364000  14259102  14448000  14470000  15034000  15080000
LINK   -- INTEGER  -- DECLARED IN SEGMENT 39 AT 04166000
          *04182000* 04185000
LINK   -- DEFINE  -- DECLARED IN SEGMENT 79 AT 07595000
          07601000  07605000  07607000  07608000  07609000
LINK   -- REAL  -- DECLARED IN SEGMENT 81 AT 07646390
          *07646420**07646533**07646543* 07646710  07646720  07646730 *07646740*
LINK   -- INTEGER  -- DECLARED IN SEGMENT 115 AT 09318000
          *09351000* 09353000
LINK   -- REAL  -- DECLARED IN SEGMENT 160 AT 16087000
          *16092000* 16094000
LINKA  -- REAL  -- DECLARED IN SEGMENT 147 AT 14254100
          *14259094* 14259100
LINKB  -- REAL  -- DECLARED IN SEGMENT 147 AT 14254100
          *14259094* 14259098
LINKC  -- DEFINE  -- DECLARED IN SEGMENT 3 AT 01156000
          02001660  02001695  02001795  02001805  02001815  02001822  02212250  02724000  02761501  02867000  02876000
          05006000  05009000  05325470  09292000  09296000  09298000  09300200  09305000  10251000  10263000  12117000
          12162000  12163000  13103000  13281000  13310000  13310200  13310350  13310750  13310950  13593000  15034000
          16288000  17004710  17025100  17025300  17080000
LINKF  -- REAL  -- DECLARED IN SEGMENT 3 AT 01606000
          *13207000* 13339100  14210000  14211000  14213000  14214000  14310000  14314000
LINKP  -- REAL  -- DECLARED IN SEGMENT 137 AT 13359000
          *13361000* 13362000  13364000
LINKR  -- DEFINE  -- DECLARED IN SEGMENT 3 AT 01155000
          02001660  02001695  02001795  02001805  02001815  02001822  02212250  02724000  02761501  02867000  02875000
          05006000  05009000  05325470  09292000  09296000  09298000  09300200  09305000  10251000  10254000  12117000
          12162000  12163000  13103000  13281000  13302000  13303000  13310000  13310200  13310350  13310750  13310950
          13593000  15034000  16288000
LINKT  -- REAL  -- DECLARED IN SEGMENT 137 AT 13359000
          *13361000* 13362000  13364000  13365000 *13366000*
LINKTOG  -- BOOLEAN  -- DECLARED IN SEGMENT 3 AT 01399000
          04048000  04100000 *04213000**04222000**04295000* 06021000 *06022000*
LISTABIT  -- DEFINE  -- DECLARED IN SEGMENT 3 AT 01000970
          02194250  02511000  02602500
LISTADDRESS  -- INTEGER  -- DECLARED IN SEGMENT 50 AT 05187540
          *05187570* 05187600  05187620
LISTADDRESS  -- INTEGER  -- DECLARED IN SEGMENT 92 AT 08385500
          *08488000* 08489500  08489510
LISTADDRESS  -- INTEGER  -- DECLARED IN SEGMENT 97 AT 08591500
          *08692000* 08694000  08694500
LISTATOG  -- DEFINE  -- DECLARED IN SEGMENT 3 AT 01001260
          02194250  02602500
LISTBIT  -- DEFINE  -- DECLARED IN SEGMENT 3 AT 01000980
          02013179  02013181  02402000  02602500  09028000
LISTDFC  -- LABEL  -- DECLARED IN SEGMENT 143 AT 14014000  --  OCCURS AT 14163000
          14019000  14185000
LISTELEMENT  -- PROCEDURE  -- DECLARED IN SEGMENT 3 AT 10289000  --  FORWARD AT 03076000
          08163000  08201000  10297000  10357000
LISTER  -- BOOLEAN  -- DECLARED IN SEGMENT 3 AT 01001570
          02195000  02195750  02214200  02227500  02464000 *02602500* 02910600  05036000  07025020  07646640  07693500
          *09028000* 09028005  09408000  13646000  13780002  14060000  14461500  14493500
LISTGEN  -- REAL PROCEDURE  -- DECLARED IN SEGMENT 3 AT 10344000  --  FORWARD AT 03077000
          08481000  08688000 *10348000* 10369000  14177000
LISTID  -- DEFINE  -- DECLARED IN SEGMENT 3 AT 01181000
```

```
        07122540  08479000  08687000  13196520  14170000  14174000
LISTMODE  --  BOOLEAN  --  DECLARED IN SEGMENT 3 AT 01720000
        08163000  08201000 *10352000**10365000*
LISTOG  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01001270
        02013179  02013181  02602500  09028000
LISTPBIT  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01000990
        02195000  02228750  02480000
LISTPLACE  --  INTEGER  --  DECLARED IN SEGMENT 123 AT 10346000
       *10348000* 10367000
LISTPTOG  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01001280
        02195000  02228750
LISTV  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01288000
        14167000
LITERAL  --  INTEGER  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 04003000
        04004000
LITNO  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01210000
        02850000  05275000  05277000  06314650  06321000  06330500  07185000  07662000  07669000  07925000  10600000
        10926500  13083000  13480000  13511000  13572000  14269460  14269500  16119000  16204000  16223000  16326100
        16341000  16374000  16384100  16384700  16410000  16435000  16479000
LITV  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01250000
        16372000  16381000
LLITOG  --  BOOLEAN  --  DECLARED IN SEGMENT 138 AT 13450000
       *13482000**13498000* 13508000  13527000  13540000
LLOCK  --  LABEL  --  DECLARED IN SEGMENT 85 AT 07717000  --  OCCURS AT 07744000
        07725000
LND  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01656000
        04545000  06204000  06289400
LNG  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01657000
        04112000  05171000  06205000  08418400  08613400  08957000  08980000
LO  --  REAL  --  DECLARED IN SEGMENT 143 AT 14029000
       *14304000* 14496000
LOC  --  DEFINE  --  DECLARED IN SEGMENT 151 AT 14420000
        14438000
LOC  --  DEFINE  --  DECLARED IN SEGMENT 157 AT 16003000
        16070000  16163000  16255000
LOCAL  --  BOOLEAN PROCEDURE  --  DECLARED IN SEGMENT 3 AT 05206000
       *05210000* 07508000  07554000  07599000  10912100
LOCALS  --  REAL ARRAY  --  DECLARED IN SEGMENT 143 AT 14024000
        14050000
LOCALV  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01249000
        14396000
LOCBEGIN  --  DEFINE  --  DECLARED IN SEGMENT 143 AT 14022000
        14050000
LOCFLD  --  DEFINE  --  DECLARED IN SEGMENT 157 AT 16004000
        16138000
LOCKSTMT  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 08701000  --  FORWARD AT 03065000
        07744000  08755000
LOCKTOG  --  BOOLEAN  --  DECLARED IN SEGMENT 92 AT 08385600
       *08390000* 08391000  08418600
LOCKTOG  --  BOOLEAN  --  DECLARED IN SEGMENT 97 AT 08591600
       *08592000* 08613200
LOCKV  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01220000
        08390000  08592000
LOCLID  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01179000
        07081000  07230000  14391000  14409000  14425000  16198000  16212000  16225000  16282000  16317000  16326100
        16330000  16335000  16340000  16368000  16406000  16479000  16482000  17095310
LOCV  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01247000
```

```
        16336000  16339000
LOD  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01658000
        04076000  04077000  04523000  05143000  05177000  05186000  05187600  05201000  05227000  05316000  07116000
        07122530  07217510  07225000  07311000  07358000  07420000  07421000  07463500  07469750  07470500  07515000
        07529200  07618000  07859060  08079000  08401060  08402000  08489520  08489550  08493860  08515000  08597600
        08605000  08695000  08696520  08728500  08860750  08910000  08915000  08917000  08970000  08976000  08980000
        08989000  08992000  10836000  10868000  10889000  10890000  15009000  15010000  15267000  15291000  15322000
        15324000
LODPOINT  --  LABEL  --  DECLARED IN SEGMENT 70 AT 07066000  --  OCCURS AT 07222000
        07257000  07264000  07275000  07279000  07313000  07336000  07348000
LOGI  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01569000
        04076000
LOLD  --  INTEGER  --  DECLARED IN SEGMENT 143 AT 14030000
        *14054000* 14595000
LOOK  --  BOOLEAN STREAM PROCEDURE  --  DECLARED IN SEGMENT 3 AT 02001020
        02001220  02001460 *02001540* 02001600  02013295
LOOP  --  STREAM LABEL  --  DECLARED IN SEGMENT 3 AT 02001080  --  OCCURS AT 02001120
        02001563  02001580
LOOP  --  LABEL  --  DECLARED IN SEGMENT 81 AT 07646410  --  OCCURS AT 07646470
        07646495  07646535  07646555
LOR  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01659000
        08413060  08416022  08630000  10960000
LOWBD  --  LABEL  --  DECLARED IN SEGMENT 70 AT 07065000  --  OCCURS AT 07165000
        07201000
LOWER  --  REAL  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 05299000
        05297000  05298000  05304000
LP  --  LABEL  --  DECLARED IN SEGMENT 70 AT 07066000  --  OCCURS AT 07276000
        07340000
LPROC  --  LABEL  --  DECLARED IN SEGMENT 85 AT 07716000  --  OCCURS AT 07736000
        07720000  07721000
LPRT  --  REAL  --  DECLARED IN SEGMENT 3 AT 01483000
        14450000 *16049000**16075000*
LREAD  --  LABEL  --  DECLARED IN SEGMENT 85 AT 07716000  --  OCCURS AT 07740000
        07724000
LRELSE  --  LABEL  --  DECLARED IN SEGMENT 85 AT 07718000  --  OCCURS AT 07753000
        07727000
LRTS  --  LABEL  --  DECLARED IN SEGMENT 70 AT 07065000  --  OCCURS AT 07138000
        07209000
LRTS  --  LABEL  --  DECLARED IN SEGMENT 122 AT 10292000  --  OCCURS AT 10329000
        10322000
LRWND  --  LABEL  --  DECLARED IN SEGMENT 85 AT 07717000  --  OCCURS AT 07745000
        07725000
LS  --  REAL  --  DECLARED IN SEGMENT 140 AT 13737000
       *13738000**13752000* 13772000
LSPACE  --  LABEL  --  DECLARED IN SEGMENT 85 AT 07717000  --  OCCURS AT 07742000
        07724000
LSPROC  --  LABEL  --  DECLARED IN SEGMENT 85 AT 07716000  --  OCCURS AT 07737000
        07720000
LSS  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01661000
        06072080
LSS  --  DEFINE  --  DECLARED IN SEGMENT 167 AT 17002000
        17117400
LSTR  --  INTEGER  --  DECLARED IN SEGMENT 3 AT 01723000
        10330000 *10350000*
LSTRTN  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01430000
        10302000  10350000  10360000
LSTSEQ  --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 45 AT 05016000
        05020000  05023000  05024200
```

```
LSTUSD  -- DEFINE  -- DECLARED IN SEGMENT 3 AT 01561090
        02013480  02013565
LTAPE  -- LABEL  -- DECLARED IN SEGMENT 11 AT 02191000  --  OCCURS AT 02195500
        02191250
LTEMP  -- DEFINE  --  DECLARED IN SEGMENT 49 AT 05163000
        05167000
LTEMP  -- DEFINE  --  DECLARED IN SEGMENT 98 AT 08717000
        08737000
LTEMP  -- DEFINE  --  DECLARED IN SEGMENT 99 AT 08776000
        08800000
LTEMP  -- DEFINE  --  DECLARED IN SEGMENT 100 AT 08837000
        08857000
LTEMP  -- DEFINE  --  DECLARED IN SEGMENT 153 AT 15030000
        15033000  15037000
LTLCR  --  INTEGER  --  DECLARED IN SEGMENT 3 AT 01561120
       *02013705* 02013713  02208250  02223750  02224750
LVAR  --  LABEL  --  DECLARED IN SEGMENT 85 AT 07716000  --  OCCURS AT 07738000
        07721000  07722000  07723000
LVL  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01151000
        05116000  05208000  06359000  07926100  10644000  10656010  10965000  13204000  13819400  13911400  16171000
        16265000
LWHILE  --  LABEL  --  DECLARED IN SEGMENT 85 AT 07717000  --  OCCURS AT 07748000
        07725000
LWRBND  --  DEFINE  --  DECLARED IN SEGMENT 125 AT 10755000
        10814000  10818000  10819000  10820000  10821000  10839000  10843000  10844000  10845000  10846000
LWRBND  --  DEFINE  --  DECLARED IN SEGMENT 156 AT 15204000
        15246000  15248000  15249000
LWRITE  --  LABEL  --  DECLARED IN SEGMENT 85 AT 07716000  --  OCCURS AT 07741000
        07724000
LX  --  LABEL  --  DECLARED IN SEGMENT 126 AT 10956000  --  OCCURS AT 10965000
        10970000
L1  --  LABEL  --  DECLARED IN SEGMENT 55 AT 05334000  --  OCCURS AT 05355000
        05364000  05366000
L1  --  LABEL  --  DECLARED IN SEGMENT 57 AT 06060110  --  OCCURS AT 06062210
        06062155
L1  --  LABEL  --  DECLARED IN SEGMENT 64 AT 06313500  --  OCCURS AT 06318500
        06314700
L1  --  LABEL  --  DECLARED IN SEGMENT 104 AT 09025000  --  OCCURS AT 09292000
        09305100
L1  --  LABEL  --  DECLARED IN SEGMENT 119 AT 10085000  --  OCCURS AT 10158000
        10155000  10155500
L1  --  LABEL  --  DECLARED IN SEGMENT 127 AT 12005000  --  OCCURS AT 12009000
        12057000
L1  --  LABEL  --  DECLARED IN SEGMENT 128 AT 12060300  --  OCCURS AT 12060550
        12061300  12065000
L1  --  LABEL  --  DECLARED IN SEGMENT 155 AT 15088000  --  OCCURS AT 15092000
        15116100
L1  --  LABEL  --  DECLARED IN SEGMENT 164 AT 16313000  --  OCCURS AT 16349000
        16353000
L1  --  LABEL  --  DECLARED IN SEGMENT 165 AT 16365000  --  OCCURS AT 16397000
        16372000
L1  --  LABEL  --  DECLARED IN SEGMENT 157 AT 16475000  --  OCCURS AT 16484000
        16480000
L10  --  LABEL  --  DECLARED IN SEGMENT 61 AT 06216000  --  OCCURS AT 06228000
        06220000
L10  --  LABEL  --  DECLARED IN SEGMENT 70 AT 07069000  --  OCCURS AT 07273000
        07073000
```

```
L10  --  LABEL  --  DECLARED IN SEGMENT 157 AT 16475000  --  OCCURS AT 16493000
         16476000
L11  --  LABEL  --  DECLARED IN SEGMENT 58 AT 06082000  --  OCCURS AT 06096000
         06086000
L11  --  LABEL  --  DECLARED IN SEGMENT 61 AT 06216000  --  OCCURS AT 06235000
         06220000  06226000
L11  --  LABEL  --  DECLARED IN SEGMENT 70 AT 07069000  --  OCCURS AT 07314000
         07073000
L12  --  LABEL  --  DECLARED IN SEGMENT 58 AT 06082000  --  OCCURS AT 06093000
         06086000
L12  --  LABEL  --  DECLARED IN SEGMENT 61 AT 06216000  --  OCCURS AT 06228000
         06220000
L12  --  LABEL  --  DECLARED IN SEGMENT 70 AT 07069000  --  OCCURS AT 07219000
         07073000
L13  --  LABEL  --  DECLARED IN SEGMENT 58 AT 06082000  --  OCCURS AT 06093000
         06086000
L13  --  LABEL  --  DECLARED IN SEGMENT 61 AT 06216000  --  OCCURS AT 06242000
         06220000
L13  --  LABEL  --  DECLARED IN SEGMENT 70 AT 07069000  --  OCCURS AT 07219000
         07073000
L14  --  LABEL  --  DECLARED IN SEGMENT 58 AT 06082000  --  OCCURS AT 06099000
         06086000
L14  --  LABEL  --  DECLARED IN SEGMENT 61 AT 06216000  --  OCCURS AT 06235000
         06220000
L14  --  LABEL  --  DECLARED IN SEGMENT 70 AT 07069000  --  OCCURS AT 07219000
         07073000
L15  --  LABEL  --  DECLARED IN SEGMENT 58 AT 06082000  --  OCCURS AT 06099000
         06086000
L15  --  LABEL  --  DECLARED IN SEGMENT 61 AT 06216000  --  OCCURS AT 06235000
         06220000
L15  --  LABEL  --  DECLARED IN SEGMENT 70 AT 07069000  --  OCCURS AT 07219000
         07073000
L16  --  LABEL  --  DECLARED IN SEGMENT 58 AT 06082000  --  OCCURS AT 06099000
         06086000
L16  --  LABEL  --  DECLARED IN SEGMENT 61 AT 06216000  --  OCCURS AT 06235000
         06220000
L16  --  LABEL  --  DECLARED IN SEGMENT 70 AT 07069000  --  OCCURS AT 07219000
         07073000
L17  --  LABEL  --  DECLARED IN SEGMENT 58 AT 06082000  --  OCCURS AT 06093000
         06086000
L17  --  LABEL  --  DECLARED IN SEGMENT 61 AT 06216000  --  OCCURS AT 06245000
         06220000
L17  --  LABEL  --  DECLARED IN SEGMENT 70 AT 07069000  --  OCCURS AT 07325000
         07073000
L18  --  LABEL  --  DECLARED IN SEGMENT 58 AT 06082000  --  OCCURS AT 06102000
         06086000
L18  --  LABEL  --  DECLARED IN SEGMENT 61 AT 06216000  --  OCCURS AT 06235000
         06220000
L18  --  LABEL  --  DECLARED IN SEGMENT 70 AT 07070000  --  OCCURS AT 07325000
         07074000
L19  --  LABEL  --  DECLARED IN SEGMENT 58 AT 06082000  --  OCCURS AT 06102000
         06086000
L19  --  LABEL  --  DECLARED IN SEGMENT 61 AT 06216000  --  OCCURS AT 06235000
         06220000
L19  --  LABEL  --  DECLARED IN SEGMENT 70 AT 07070000  --  OCCURS AT 07322000
         07074000
L2   --  LABEL  --  DECLARED IN SEGMENT 55 AT 05334000  --  OCCURS AT 05373000
         05362000
```

```
L2  -- LABEL --  DECLARED IN SEGMENT 57 AT 06060110  --  OCCURS AT 06062165
     06062185
L2  -- LABEL --  DECLARED IN SEGMENT 64 AT 06313500  --  OCCURS AT 06328000
     06321000
L2  -- LABEL --  DECLARED IN SEGMENT 119 AT 10085000  --  OCCURS AT 10172000
     10132000  10134600  10144000
L2  -- LABEL --  DECLARED IN SEGMENT 127 AT 12005000  --  OCCURS AT 12055000
     12018000  12048000
L2  -- LABEL --  DECLARED IN SEGMENT 128 AT 12060300  --  OCCURS AT 12064900
     12064500
L2  -- LABEL --  DECLARED IN SEGMENT 157 AT 16475000  --  OCCURS AT 16485000
     16481000
L20  -- LABEL --  DECLARED IN SEGMENT 58 AT 06083000  --  OCCURS AT 06102000
     06087000
L20  -- LABEL --  DECLARED IN SEGMENT 61 AT 06217000  --  OCCURS AT 06235000
     06221000
L20  -- LABEL --  DECLARED IN SEGMENT 70 AT 07070000  --  OCCURS AT 07322000
     07074000
L21  -- LABEL --  DECLARED IN SEGMENT 58 AT 06083000  --  OCCURS AT 06093000
     06087000
L21  -- LABEL --  DECLARED IN SEGMENT 61 AT 06217000  --  OCCURS AT 06248000
     06221000
L21  -- LABEL --  DECLARED IN SEGMENT 70 AT 07070000  --  OCCURS AT 07344000
     07074000
L22  -- LABEL --  DECLARED IN SEGMENT 58 AT 06083000  --  OCCURS AT 06105000
     06087000  06094950
L22  -- LABEL --  DECLARED IN SEGMENT 61 AT 06217000  --  OCCURS AT 06235000
     06221000
L22  -- LABEL --  DECLARED IN SEGMENT 70 AT 07070000  --  OCCURS AT 07344000
     07074000
L23  -- LABEL --  DECLARED IN SEGMENT 58 AT 06083000  --  OCCURS AT 06105000
     06087000
L23  -- LABEL --  DECLARED IN SEGMENT 61 AT 06217000  --  OCCURS AT 06235000
     06221000
L23  -- LABEL --  DECLARED IN SEGMENT 70 AT 07070000  --  OCCURS AT 07341000
     07074000
L24  -- LABEL --  DECLARED IN SEGMENT 58 AT 06083000  --  OCCURS AT 06105000
     06087000
L24  -- LABEL --  DECLARED IN SEGMENT 61 AT 06217000  --  OCCURS AT 06235000
     06221000
L24  -- LABEL --  DECLARED IN SEGMENT 70 AT 07070000  --  OCCURS AT 07341000
     07074000
L25  -- LABEL --  DECLARED IN SEGMENT 58 AT 06083000  --  OCCURS AT 06093000
     06087000
L25  -- LABEL --  DECLARED IN SEGMENT 61 AT 06217000  --  OCCURS AT 06248000
     06221000
L25  -- LABEL --  DECLARED IN SEGMENT 70 AT 07070000  --  OCCURS AT 07356000
     07074000
L26  -- LABEL --  DECLARED IN SEGMENT 58 AT 06083000  --  OCCURS AT 06105000
     06087000
L26  -- LABEL --  DECLARED IN SEGMENT 61 AT 06217000  --  OCCURS AT 06235000
     06221000
L26  -- LABEL --  DECLARED IN SEGMENT 70 AT 07070000  --  OCCURS AT 07356000
     07074000
L27  -- LABEL --  DECLARED IN SEGMENT 58 AT 06083000  --  OCCURS AT 06105000
     06087000
L27  -- LABEL --  DECLARED IN SEGMENT 61 AT 06217000  --  OCCURS AT 06235000
```

06221000
L27 -- LABEL -- DECLARED IN SEGMENT 70 AT 07070000 -- OCCURS AT 07353000
07074000
L28 -- LABEL -- DECLARED IN SEGMENT 58 AT 06083000 -- OCCURS AT 06105000
06087000
L28 -- LABEL -- DECLARED IN SEGMENT 61 AT 06217000 -- OCCURS AT 06235000
06221000
L28 -- LABEL -- DECLARED IN SEGMENT 70 AT 07070000 -- OCCURS AT 07353000
07074000
L29 -- LABEL -- DECLARED IN SEGMENT 58 AT 06083000 -- OCCURS AT 06093000
06087000
L29 -- LABEL -- DECLARED IN SEGMENT 61 AT 06217000 -- OCCURS AT 06251000
06221000
L29 -- LABEL -- DECLARED IN SEGMENT 70 AT 07070000 -- OCCURS AT 07372000
07074000
L3 -- LABEL -- DECLARED IN SEGMENT 57 AT 06060110 -- OCCURS AT 06062135
06062500
L3 -- LABEL -- DECLARED IN SEGMENT 64 AT 06313500 -- OCCURS AT 06331500
06330500
L3 -- LABEL -- DECLARED IN SEGMENT 119 AT 10085000 -- OCCURS AT 10170000
10167000
L3 -- LABEL -- DECLARED IN SEGMENT 127 AT 12005000 -- OCCURS AT 12056000
12026000 12039000
L3 -- LABEL -- DECLARED IN SEGMENT 128 AT 12060300 -- OCCURS AT 12065000
12063600
L3 -- LABEL -- DECLARED IN SEGMENT 157 AT 16475000 -- OCCURS AT 16486000
16476000
L30 -- LABEL -- DECLARED IN SEGMENT 58 AT 06084000 -- OCCURS AT 06093000
06088000
L30 -- LABEL -- DECLARED IN SEGMENT 61 AT 06218000 -- OCCURS AT 06254000
06222000
L30 -- LABEL -- DECLARED IN SEGMENT 70 AT 07070000 -- OCCURS AT 07375000
07074000
L31 -- LABEL -- DECLARED IN SEGMENT 58 AT 06084000 -- OCCURS AT 06111000
06088000
L31 -- LABEL -- DECLARED IN SEGMENT 61 AT 06218000 -- OCCURS AT 06236000
06222000
L31 -- LABEL -- DECLARED IN SEGMENT 70 AT 07071000 -- OCCURS AT 07383000
07075000
L32 -- LABEL -- DECLARED IN SEGMENT 58 AT 06084000 -- OCCURS AT 06108000
06088000
L32 -- LABEL -- DECLARED IN SEGMENT 61 AT 06218000 -- OCCURS AT 06236000
06222000
L32 -- LABEL -- DECLARED IN SEGMENT 70 AT 07071000 -- OCCURS AT 07378000
07075000
L33 -- LABEL -- DECLARED IN SEGMENT 58 AT 06084000 -- OCCURS AT 06111000
06088000
L33 -- LABEL -- DECLARED IN SEGMENT 61 AT 06218000 -- OCCURS AT 06236000
06222000
L33 -- LABEL -- DECLARED IN SEGMENT 70 AT 07071000 -- OCCURS AT 07383000
07075000
L34 -- LABEL -- DECLARED IN SEGMENT 58 AT 06084000 -- OCCURS AT 06122000
06088000
L34 -- LABEL -- DECLARED IN SEGMENT 61 AT 06218000 -- OCCURS AT 06257000
06222000
L35 -- LABEL -- DECLARED IN SEGMENT 58 AT 06084000 -- OCCURS AT 06114000
06088000
L35 -- LABEL -- DECLARED IN SEGMENT 61 AT 06218000 -- OCCURS AT 06232000

```
          06222000
L4   --  LABEL  --  DECLARED IN SEGMENT 70 AT 07069000  --  OCCURS AT 07223000
          07073000
L4   --  LABEL  --  DECLARED IN SEGMENT 127 AT 12005000  --  OCCURS AT 12025000
          12041140
L4   --  LABEL  --  DECLARED IN SEGMENT 128 AT 12060300  --  OCCURS AT 12061300
          12061800
L4   --  LABEL  --  DECLARED IN SEGMENT 157 AT 16475000  --  OCCURS AT 16487000
          16476000
L5   --  LABEL  --  DECLARED IN SEGMENT 70 AT 07069000  --  OCCURS AT 07245000
          07073000
L5   --  LABEL  --  DECLARED IN SEGMENT 128 AT 12060300  --  OCCURS AT 12061800
          12064000
L5   --  LABEL  --  DECLARED IN SEGMENT 157 AT 16475000  --  OCCURS AT 16488000
          16476000
L6   --  LABEL  --  DECLARED IN SEGMENT 70 AT 07069000  --  OCCURS AT 07252000
          07073000
L6   --  LABEL  --  DECLARED IN SEGMENT 157 AT 16475000  --  OCCURS AT 16489000
          16476000
L7   --  LABEL  --  DECLARED IN SEGMENT 70 AT 07069000  --  OCCURS AT 07258000
          07073000
L7   --  LABEL  --  DECLARED IN SEGMENT 157 AT 16475000  --  OCCURS AT 16490000
          16476000  16482000
L8   --  LABEL  --  DECLARED IN SEGMENT 70 AT 07069000  --  OCCURS AT 07223000
          07073000
L8   --  LABEL  --  DECLARED IN SEGMENT 157 AT 16475000  --  OCCURS AT 16491000
          16476000
L9   --  LABEL  --  DECLARED IN SEGMENT 61 AT 06215000  --  OCCURS AT 06251000
          06219000
L9   --  LABEL  --  DECLARED IN SEGMENT 70 AT 07069000  --  OCCURS AT 07262000
          07073000
L9   --  LABEL  --  DECLARED IN SEGMENT 157 AT 16475000  --  OCCURS AT 16492000
          16476000
M    --  BOOLEAN  --  DECLARED IN SEGMENT 55 AT 05343000
          *05353000* 05355000  05357000 *05359000**05365000*
M    --  REAL  --  DECLARED IN SEGMENT 131 AT 13039000
          *13074000**13076000* 13111000
MACRO  --  REAL ARRAY  --  DECLARED IN SEGMENT 3 AT 01487000
          *09215000* 16346000  16349000
MACRO  --  BOOLEAN  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 10228000
          10263600  10276500
MACROID  --  BOOLEAN  --  DECLARED IN SEGMENT 3 AT 01717800
          02882100  02913000 *12107000**12127000* 13305200 *14265900**14266100*
MAKCAST  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 02181500
          02229000  02230750  02908000
MAKEALABEL  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 06500000
          07686000
MAKEPARAM  --  STREAM PROCEDURE  --  DECLARED IN SEGMENT 130 AT 12154000
          12161000  12162000
MAKEUPACCUM  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 13368000  --  FORWARD AT 01627000
          13349000  13917000  14507140  06285600  06289400 *06289800*
MARK  --  REAL  --  DECLARED IN SEGMENT 3 AT 01615000
          14116300  14117000 *14314000**14319000* 14333100  14334000  14374000  14430000  14451000  14463000
MARKMONITORED  --  LABEL  --  DECLARED IN SEGMENT 124 AT 10553000  --  OCCURS AT 10573000
          10667000
MASK  --  BOOLEAN STREAM PROCEDURE  --  DECLARED IN SEGMENT 55 AT 05341000
          05342000  05348000  05353000
```

```
MATCH  --  DEFINE  --  DECLARED IN SEGMENT 177 AT 17091115
       17091800  06061100
MAXLCR  --  REAL  --  NAME PARAMETER  --  DECLARED IN SEGMENT 13 AT 02198750  --  OCCURS AT 06072010
       02198500  02199750 *02200500**02202000*
MAXLTLCR  --  INTEGER  --  DECLARED IN SEGMENT 3 AT 01561120
       02013527 *02013710* 02208250  02223750
MAXROW  --  INTEGER  --  DECLARED IN SEGMENT 3 AT 01684000
      *13537000*
MAXSAVE  --  INTEGER  --  DECLARED IN SEGMENT 3 AT 01598000
       13533000 *13534000*
MAXSTACK  --  REAL  --  DECLARED IN SEGMENT 3 AT 01683000
      *09032000* 13230000 *13231000**13748000* 13750000  14277000 *14478040**14502000* 14535000 *14536000* 14567000
MAXSTACKO  --  REAL  --  DECLARED IN SEGMENT 143 AT 14029000
      *14277000* 14502000
MAXTLCR  --  INTEGER  --  DECLARED IN SEGMENT 3 AT 01331000
       02208500  02219500  02439000 *02456000*
MAXV  --  DEFINE  --  DECLARED IN SEGMENT 57 AT 06061200
       06072080
MAYI  --  BOOLEAN  --  DECLARED IN SEGMENT 92 AT 08385700
      *08401045* 08424100  08450100
MCPBIT  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01001000
       02397000
MCPTOG  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01001290
MDESC  --  STREAM PROCEDURE  --  DECLARED IN SEGMENT 104 AT 09005000
       09352000
MDS  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01661100
       13767300  14569200
MEDIUM  --  ALPHA  --  DECLARED IN SEGMENT 3 AT 01001580
      *02013566* 02195000  02195750 *02203250**02204250**02205500**02214100* 02214200 *02224100* 02227500  02228750
      *02238720* 05038000  07025020
MERGEBIT  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01001010
       01835600  02425000  02425500  02426000  02427000  02428000  02435000  02436000
MERGEI  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01573100
       08906000  08965000
MERGEOPTION  --  LABEL  --  DECLARED IN SEGMENT 21 AT 02363000  --  OCCURS AT 02437000
       02428000
MERGESTMT  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 08901000  --  FORWARD AT 03082000
       07827000  08927000
MERGETOG  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01001300
       01835600  02425500  02427000  02428000  02436000
MERRIMAC  --  BOOLEAN PROCEDURE  --  DECLARED IN SEGMENT 3 AT 10370000
      *10565200* 10680000  14148000
MIN  --  INTEGER  --  DECLARED IN SEGMENT 5 AT 01822000
      *01828000* 01834000
MKABS  --  REAL STREAM PROCEDURE  --  DECLARED IN SEGMENT 3 AT 01733000
      *01734000* 02013245  02013528  02013617  02013621  02013705  02013710  02013715  02200500  02202000  02202050
       02214000  02215750  02216500  02217750  02221000  02224050  02226000  02238582  02417000  02456000  02492000
       02903000  07025010  07025030  08948000  09026000  13101100
MKS  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01662000
       04075000  05167000  06062110  06064000  06318500  06328000  06331700  06358000  06503000  07111230  07111320
       07408000  07418000  07441000  07514000  07529100  07617000  08400000  08493570  08513000  08597000  08698100
       08906000  08953000  08956000  08985000  09272000  10803000  12041130  12061700  12064000  13053000  13082500
       13159000  13166000  13196400  13470000  13565000  13904000  15007000  15241000
MODE  --  INTEGER  --  DECLARED IN SEGMENT 3 AT 01402000
       05115000  05351000  06365000  07732000  08093000  10618000  13906000  13914000  14178000  14229000  14241000
      *14303000* 14478030 *14501000*
MON  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01147000
       02896000  16254000  16288000
```

```
MONITORDEC -- LABEL -- DECLARED IN SEGMENT 143 AT 14014000 -- OCCURS AT 14142000
        14019000
MONITORV -- DEFINE -- DECLARED IN SEGMENT 3 AT 01291000
        14129000
MOVE -- STREAM PROCEDURE -- DECLARED IN SEGMENT 3 AT 01735000
        01736300  02013250  02013275  02013520  02013525  02013615  02013697  02212250  02366100  02602600  02916000
        02927530  02927540  02927570  02927610  02927620  04147000  05039500  05058000  07029000  07666500  07673000
        07675000  07697000  09314000  09395000  09407000  10071000  13281000  13310000  14050000  14259000  14259080
        14440000  14507040  14507105  14507170
MOVEANDCLOCK -- INTEGER PROCEDURE -- DECLARED IN SEGMENT 3 AT 02927100
       *02927410* 02927770  09396000  09399000  09399100  09402000  09407010  13640000  13654000
MOVECHARACTERS -- STREAM PROCEDURE -- DECLARED IN SEGMENT 2 AT 00515000
        00522000  02705000  02776300  02777050  02781000  07673500  07677000  09415000  09416000  16349000  16384800
        16392000
MOVECODE -- STREAM PROCEDURE -- DECLARED IN SEGMENT 3 AT 01688010
        07646630  07646690  07694000  07698500  08860150  08860450  13777600  13818500  14049000  14594000  14595000
MOVEIT -- LABEL -- DECLARED IN SEGMENT 27 AT 02638000 -- OCCURS AT 02704000
MOVEXREFINFO -- STREAM PROCEDURE -- DECLARED IN SEGMENT 7 AT 02001760
        02001785  02001805
MRCLEAN -- BOOLEAN -- DECLARED IN SEGMENT 3 AT 01379000
        04164000 *09040000* 04064000  04067000  08049000  12061725
MULFID -- STREAM VARIABLE -- VALUE PARAMETER -- DECLARED IN SEGMENT 131 AT 13020000
        13021000  13027000
MULFID -- REAL -- DECLARED IN SEGMENT 131 AT 13039000
       *13081000**13097000* 13102000
MULOP -- DEFINE -- DECLARED IN SEGMENT 3 AT 01277000
        08435000  08437000  08653120  08653125  08653180  08653195  12022000  12061500
MVE -- STREAM PROCEDURE -- DECLARED IN SEGMENT 15 AT 02238180
        02238385  02238440
MVEWD -- STREAM PROCEDURE -- DECLARED IN SEGMENT 15 AT 02238212
        02238580
MYCLASS -- INTEGER -- DECLARED IN SEGMENT 3 AT 01001590
       *02968000* 02971000 *02974500* 02975000  02976500  02977500 *02978500* 02982000  02983000
M0 -- STREAM LABEL -- DECLARED IN SEGMENT 54 AT 05325080 -- OCCURS AT 05325320
        05325240
M1 -- STREAM LABEL -- DECLARED IN SEGMENT 54 AT 05325080 -- OCCURS AT 05325350
        05325250
M2 -- STREAM LABEL -- DECLARED IN SEGMENT 54 AT 05325080 -- OCCURS AT 05325360
        05325260
M3 -- STREAM LABEL -- DECLARED IN SEGMENT 54 AT 05325080 -- OCCURS AT 05325380
        05325270
M4 -- STREAM LABEL -- DECLARED IN SEGMENT 54 AT 05325080 -- OCCURS AT 05325390
        05325280
M4 -- PROCEDURE -- DECLARED IN SEGMENT 156 AT 15215000
        15290000  15303000
M5 -- STREAM LABEL -- DECLARED IN SEGMENT 54 AT 05325080 -- OCCURS AT 05325400
        05325290
M6 -- STREAM LABEL -- DECLARED IN SEGMENT 54 AT 05325080 -- OCCURS AT 05325410
        05325300
M7 -- STREAM LABEL -- DECLARED IN SEGMENT 54 AT 05325080 -- OCCURS AT 05325420
        05325310
N -- STREAM VARIABLE -- VALUE PARAMETER -- DECLARED IN SEGMENT 2 AT 00515000
        00516000  00521000
N -- REAL -- DECLARED IN SEGMENT 3 AT 01693000
       *14099000* 14100000  14101000  14104000  14108000  14109000 *14111000*
N -- INTEGER -- VALUE PARAMETER -- DECLARED IN SEGMENT 3 AT 01717950
N -- STREAM VARIABLE -- VALUE PARAMETER -- DECLARED IN SEGMENT 3 AT 01737350
```

```
          01737450
N  --  STREAM VARIABLE  --  DECLARED IN SEGMENT 3 AT 01759000
          01760000  01779000 *01781000*
N  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 7 AT 02001760
          02001765  02001780
N  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 02001831
          02001833
N  --  INTEGER  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 8 AT 02013176
          02013180
N  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 12 AT 02196530
          02196560
N  --  INTEGER  --  DECLARED IN SEGMENT 16 AT 02249000
         *02251000**02252000* 02255000  02259000
N  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 17 AT 02264400
          02264550
N  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 17 AT 02264700
          02264850
N  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 29 AT 02927240
          02927270
N  --  REAL  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 03061010
N  --  REAL  --  DECLARED IN SEGMENT 40 AT 04204000
         *04208000* 04209000  04221000  04223000
N  --  INTEGER  --  DECLARED IN SEGMENT 41 AT 04237000
         *04238000* 04239000 *04240000*
N  --  INTEGER  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 05325010
          05325460  05325470  05325480
N  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 54 AT 05325040
          05325050  05325230
N  --  BOOLEAN  --  DECLARED IN SEGMENT 59 AT 06151000
         *06152000* 06154000
N  --  REAL  --  DECLARED IN SEGMENT 81 AT 07646390
         *07646420* 07646490 *07646495**07646510* 07646545 *07646548**07646556* 07646670
N  --  REAL  --  DECLARED IN SEGMENT 86 AT 07802000
         *07820000**07821200**07823000**07825000* 07904000  07912000
N  --  REAL  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 08493470
          08493520  08493705  08493770  08493840
N  --  REAL  --  DECLARED IN SEGMENT 120 AT 10258100
         *10264650* 10264760  10264810
N  --  REAL  --  DECLARED IN SEGMENT 126 AT 10957000
          10967000 *10968000**10971000* 10972000 *10978000* 10979000 *10987000* 10989000 *10990000**10995000* 10996000
          11001000
N  --  INTEGER  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 12151000
          12150000  12151000  12163000
N  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 130 AT 12154000
          12158000
N  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 131 AT 13020000
          13021000  13031000 *13034000* 13036000
N  --  REAL  --  DECLARED IN SEGMENT 134 AT 13221000
         *13258000* 13265000  13271000
N  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 172 AT 17051400
          17051500  17051900
NAME  --  REAL  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 02927110
          02927100  02927110  02927430  02927440  02927460
NB  --  REAL  --  DECLARED IN SEGMENT 148 AT 14269060
         *14269448**14269540* 14269560  14269600
NBITF  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01154300
          05273400  05274200  06313700  06314200  14269448  14269600
NCII  --  INTEGER  --  DECLARED IN SEGMENT 3 AT 01624000
```

```
        *04177000**13228000* 13750000  13764000  14056000 *14057000* 14560000  14567000 *14611000*
NCIIO  -- INTEGER -- DECLARED IN SEGMENT 143 AT 14031000
        *14056000* 14611000
NCR  -- STREAM VARIABLE -- VALUE PARAMETER -- DECLARED IN SEGMENT 2 AT 00511000
        00512000
NCR  -- INTEGER -- DECLARED IN SEGMENT 3 AT 01330000
        02013313  02013317  02013445  02013527  02013528 *02013617**02013620**02013715* 02128500  02129000  02129500
       *02213000**02214000**02225250**02229000**02230750* 02410000  02485000  02665000  02680000  02696000  02697000
       *02720000* 02730000  02731000  02745000  02750000  02753000  02761000  02769000  02786000  02791000  02807000
        02813000  02888000  02902000 *02903000* 02936000  02940000  02946000  10167400  10167600
NCR  -- STREAM VARIABLE -- VALUE PARAMETER -- DECLARED IN SEGMENT 3 AT 01756000
        01761000 *01768000**01773000* 01780000
NCR  -- STREAM VARIABLE -- NAME PARAMETER -- DECLARED IN SEGMENT 10 AT 02089500
        02126000
NCR  -- STREAM VARIABLE -- VALUE PARAMETER -- DECLARED IN SEGMENT 3 AT 02183500
        02183750  02184500
NCRLINK -- DEFINE -- DECLARED IN SEGMENT 3 AT 01561110
        02013530  02013620
NCRV  -- STREAM VARIABLE -- VALUE PARAMETER -- DECLARED IN SEGMENT 10 AT 02090000
        02090500  02092500 *02102000* 02124500
NDIV32  -- STREAM VARIABLE -- VALUE PARAMETER -- DECLARED IN SEGMENT 3 AT 01737000
        01737050  01737150
NED  -- STREAM LABEL -- DECLARED IN SEGMENT 3 AT 01742600 -- OCCURS AT 01743100
        01742900
NEQ  -- DEFINE -- DECLARED IN SEGMENT 3 AT 01664000
        06069000
NEQ  -- DEFINE -- DECLARED IN SEGMENT 167 AT 17002000
NESTBIT  -- DEFINE -- DECLARED IN SEGMENT 3 AT 01001020
        02472000
NESTLEVEL  -- REAL -- DECLARED IN SEGMENT 3 AT 01484000
       *16126000**16142000* 16166000  16171000  16260000  16265000
NESTOG  -- DEFINE -- DECLARED IN SEGMENT 3 AT 01001310
NESTS  -- PROCEDURE -- DECLARED IN SEGMENT 157 AT 16115000
        16144000  16484000
NEW  -- STREAM VARIABLE -- NAME PARAMETER -- DECLARED IN SEGMENT 3 AT 02016000
        02018000
NEWBASE  -- BOOLEAN -- DECLARED IN SEGMENT 3 AT 01001490
        02192000 *02192500**02583000**09028920*
NEWBIT  -- DEFINE -- DECLARED IN SEGMENT 3 AT 01001030
        02193750  02238570  02388000  09275300
NEWID  -- DEFINE -- DECLARED IN SEGMENT 167 AT 17002009
        17080000
NEWINCL  -- DEFINE -- DECLARED IN SEGMENT 3 AT 01001330
NEWINCLBIT  -- DEFINE -- DECLARED IN SEGMENT 3 AT 01001040
NEWINX  -- INTEGER -- DECLARED IN SEGMENT 3 AT 01000860
       *09361005* 09361080  09361100
NEWTAPE  -- FILE -- DECLARED IN SEGMENT 3 AT 01560000
        02194000  02238590  09275500  09407400
NEWTOG  -- DEFINE -- DECLARED IN SEGMENT 3 AT 01001320
        02193750  02238570  09275300
NEXT  -- INTEGER PROCEDURE -- DECLARED IN SEGMENT 3 AT 02956000
       *02968000* 02968500  02974500  02978500
NEXT  -- LABEL -- DECLARED IN SEGMENT 156 AT 15184000 -- OCCURS AT 15253000
        15292000
NEXTENT  -- PROCEDURE -- DECLARED IN SEGMENT 3 AT 02932000
        02954000  10112000  10113200  10118000  10123000  10134520  10134540  10150120  10150150  10150165  10150195
        10158000  10167700  10172000  10178000
```

```
NEXTENTRY  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01561100
       02013497  02013537  02013538  02013567
NEXTINFO  --  REAL  --  DECLARED IN SEGMENT 3 AT 01620000
       02265350  04178000  04179000 *09033000**09214510* 09214520  09214530 *09214980* 10582000 *13291000* 13302000
       13303000  13304000 *13305000* 13308000  13309000  13310000  13310200  13310580  13310750  13310950  13311000
      *13312000* 13470000 *13471000* 13566000 *13567000* 13593000 *13594000* 13789000 *13795000* 14055300 *14214000*
      *14248000* 14259094  14319000  14329000 *14330000**14373000* 14375000 *14463000* 16053000  16055000  16442000
NEXTLINK  --  DEFINE  --  DECLARED IN SEGMENT 79 AT 07596000
       07607000  07609000
NEXTSAVE  --  BOOLEAN  --  DECLARED IN SEGMENT 143 AT 14034100
      *14276000**14294100**14295100* 14311100  14470000  14482000  14487010
NEXTTEXT  --  INTEGER  --  DECLARED IN SEGMENT 3 AT 01481400
      *02723500* 02724000 *09251300* 10250000  10254000  10263000 *10285000* 12117000 *12118000* 12121000  14055500
       14265950 *14574500*
NHI  --  REAL  --  DECLARED IN SEGMENT 3 AT 01689000
      *02681000**02688000**02785000* 02800000  02801000  02833000  02835000  02838000  02840000  12015000
NINFOO  --  INTEGER  --  DECLARED IN SEGMENT 143 AT 14030000
      *14055300* 14555000  14569000  14573000
NLAB  --  REAL  --  DECLARED IN SEGMENT 151 AT 14418000
      *14422000**14437000* 14450000
NLO  --  REAL  --  DECLARED IN SEGMENT 3 AT 01689000
      *02681000**02688000**02785000* 02800000  02833000  02835000  02838000  02840000  12014000
NLOC  --  REAL  --  DECLARED IN SEGMENT 151 AT 14418000
      *14422000**14433000* 14450000
NLOCS  --  DEFINE  --  DECLARED IN SEGMENT 143 AT 14022000
       14024000  14050000
NMOD32  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 01737000
       01737050  01737200
NO  --  REAL  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 03044000
       03042000  03043000
NO  --  REAL  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 13636000
       13634000  13635000  13639000  13640000  13649000  13650000  13652000
NODIM  --  DEFINE  --  DECLARED IN SEGMENT 124 AT 10513000
       10583000  10584000  10590000  10656030  10657000  10658000
NODIM  --  DEFINE  --  DECLARED IN SEGMENT 125 AT 10719000
       10808000  10827000  10868000
NODIM  --  DEFINE  --  DECLARED IN SEGMENT 156 AT 15185000
       15244000
NODIMPART  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01532000
       07241000  07361000  07412000  10584000  10658000  10808000  15245000
NOHEADING  --  BOOLEAN  --  DECLARED IN SEGMENT 3 AT 01001480
      *01836000* 02195000  02195750  02196590  02214200  02227250  02227500  02228750  02264950  05039600  05046000
       05265000  05325490  07025020 *09028050* 09302000  09408000  13633000  13648000
NONBLANK  --  BOOLEAN STREAM PROCEDURE  --  DECLARED IN SEGMENT 3 AT 01742200
      *01743200* 01743300
NONLITNO  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01209000
       02851000  02924000  04178000  06286600  07669000  08017700  08032000  08994040  10593000  10926500  13226000
       16384100  16384700
NOO  --  REAL  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 03044000
       03042000  03043000
NOO  --  REAL  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 13636000
       13634000  13635000  13649000  13650000
NOOFARRAYS  --  INTEGER  --  DECLARED IN SEGMENT 3 AT 01340050
       09361040  09361050 *13551500*
NOP  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01665000
       07108500  07108505  07525000  08094000  10802000
NOP  --  DEFINE  --  DECLARED IN SEGMENT 157 AT 16012000
```

```
        16038000  16122000  16227000
NOPAR  --  DEFINE  --  DECLARED IN SEGMENT 124 AT 10546000
        10656030  10657000  10658000
NOPARPART  --  DEFINE  --  DECLARED IN SEGMENT 124 AT 10541000
        10658000
NOPARTIAL  --  LABEL  --  DECLARED IN SEGMENT 8 AT 02013175  --  OCCURS AT 02013475
        02013320
NORELEASE  --  BOOLEAN  --  DECLARED IN SEGMENT 3 AT 01561215
       *02013627**02134000* 02215500  02216250 *02229100**02234650**02735000*
NORMAL  --  LABEL  --  DECLARED IN SEGMENT 42 AT 04311010  --  OCCURS AT 04311120
        04311060  04311080
NORMAL  --  LABEL  --  DECLARED IN SEGMENT 70 AT 07065000  --  OCCURS AT 07124000
        07284000
NORMALREF  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01007495
        02882200  13310580
NOROWS  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01561220
        01561240
NOTOP  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01268000
        02964000  02974500  06152000
NSBS  --  LABEL  --  DECLARED IN SEGMENT 70 AT 07066000  --  OCCURS AT 07226000
        07272000  07374000
NSEGS  --  INTEGER  --  DECLARED IN SEGMENT 29 AT 02927130
       *02927380* 02927390
NTEXTO  --  INTEGER  --  DECLARED IN SEGMENT 143 AT 14032500
       *14055500* 14574500
NULL  --  REAL  --  DECLARED IN SEGMENT 81 AT 07646390
       *07646420* 07646490 *07646495* 07646560 *07646565* 07646570 *07646575*
NUMBER  --  INTEGER  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 03033000
        03032000
NUMBER  --  INTEGER  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 10185000
        10184000  10185000  10187000 *10188000* 10189000 *10190000* 10191000  10192000
NUMBERS  --  STREAM LABEL  --  DECLARED IN SEGMENT 10 AT 02091500  --  OCCURS AT 02117000
        02096000  02102500
NUMLE  --  BOOLEAN  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 89 AT 08080000
        08095000  08162000  08165000
NUMSEQUENCEERRORS  --  INTEGER  --  DECLARED IN SEGMENT 3 AT 01000700
       *02226300**02227600* 09416002  09416004  09416006
NXTELBT  --  INTEGER  --  DECLARED IN SEGMENT 3 AT 01319000
        02644000  02910000 *02912000**02917000**08110000**08174000**09038000* 10263400 *10263500* 10277000  10282000
        12112000 *12113000* 12125000
NXTINFOTEMP  --  DEFINE  --  DECLARED IN SEGMENT 124 AT 10531000
        10582000  10590000  10625000  10627000  10634000
N1  --  ALPHA  --  DECLARED IN SEGMENT 5 AT 01822000
       *01828500* 01835500 *01835800*
N2  --  ALPHA  --  DECLARED IN SEGMENT 5 AT 01822000
       *01828500* 01835500 *01835800*
OCR  --  REAL  --  DECLARED IN SEGMENT 134 AT 13221000
       *13249000* 13250000  13252000 *13257000* 13262000  13263000
OCTALWORDS  --  STREAM PROCEDURE  --  DECLARED IN SEGMENT 17 AT 02264400
        02264650  02265100  02265250  02265700  02265950
OCTALWORDS  --  STREAM PROCEDURE  --  DECLARED IN SEGMENT 29 AT 02927240
        02927360  02927690
OCTIZF  --  BOOLEAN STREAM PROCEDURE  --  DECLARED IN SEGMENT 3 AT 02001838
       *02001852* 02001854  02772000  07677500
OLDENTLPTR  --  INTEGER  --  DECLARED IN SEGMENT 143 AT 14024300
       *14049200* 14593000
OLDLASTADDRESS  --  INTEGER  --  DECLARED IN SEGMENT 143 AT 14024200
```

```
      *14049100* 14593000
OLDNINFOO  --  REAL  --  DECLARED IN SEGMENT 3 AT 01620400
      *14055250* 14606100
OLDSEQ  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 01741200
       01741400
OMIT  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 02183500
       02183750  02186750
OMITBIT  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01001050
       02194250  02195000  02195750  02214200  02227500  02228750  02235250  02421000  07025020
OMITTING  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01001340
       02194250  02195000  02195750  02214200  02227500  02228750  02235250  07025020
ONEPARENSH  --  LABEL  --  DECLARED IN SEGMENT 97 AT 08576000  --  OCCURS AT 08611000
       08638000
OP  --  INTEGER  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 04600000
       04603000  04604000  04605000  04606000  04607000
OPARSIZE  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01000902
       01000904  02316000  02331000  02525006
OPBIT  --  DEFINE  --  DECLARED IN SEGMENT 156 AT 15199000
       15247000
OPCLASS  --  REAL  --  DECLARED IN SEGMENT 35 AT 02981000
      *02982000* 02983000  02983500  02984000
OPCLASS  --  INTEGER  --  DECLARED IN SEGMENT 56 AT 06042000
      *06048000* 06050000  06052000  06055000
OPCLASS  --  INTEGER  --  DECLARED IN SEGMENT 60 AT 06182000
      *06188000* 06192000  06196000  06209000
OPCODE  --  DEFINE  --  DECLARED IN SEGMENT 165 AT 16366000
       16379500
OPDC  --  DEFINE  --  DECLARED IN SEGMENT 89 AT 08016000
       08108000  08130000  08217000
OPERATOR  --  INTEGER  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 04006000
       04007000
OPERATOR  --  INTEGER  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 04011000
       04010000  04014000
OPERATOR  --  INTEGER  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 04030000
       04028000  04029000  04034000
OPERATOR  --  INTEGER  --  DECLARED IN SEGMENT 56 AT 06042000
      *06044000* 06054000
OPERATOR  --  INTEGER  --  DECLARED IN SEGMENT 60 AT 06182000
      *06184000* 06208000
OPERATOR  --  INTEGER  --  DECLARED IN SEGMENT 62 AT 06282200
      *06284400* 06289200
OPINX  --  INTEGER  --  DECLARED IN SEGMENT 3 AT 01000800
      *02311000**02312000* 02314000  02316000  02317000  02334000  02338000  02342000  02356000
OPLIT  --  DEFINE  --  DECLARED IN SEGMENT 124 AT 10522000
       10598500  10603000  10617000  10626000
OPTIONLENGTH  --  SWITCH LABEL  --  DECLARED IN SEGMENT 21 AT 02364000
       02373000
OPTIONS  --  REAL ARRAY  --  DECLARED IN SEGMENT 3 AT 01000904
       02312000 *02314000* 02317000 *02331000**02334000**02338000**02352000**02356000**02426000**02525007**09027002*
      *09028005*
OPTIONWORD  --  BOOLEAN  --  DECLARED IN SEGMENT 3 AT 01000910
       01835600 *02013179**02013181* 02191500  02193750  02194250  02195000  02195750  02196590  02214200  02225750
       02226250  02227250  02227500  02228750  02231500  02232500 *02233750*  02234800  02235250  02238570 *02330000*
      *02335000**02339000**02353000**02357000**02425500* 02427000  02428000  02436000 *02525005*  02533000  02602500
       02882200  02910600  02927420  02927640  04148000  04150000  04279100  04284100  04297000  05039600  05046000
       05048000  05263000  05265000  05325490  05376100  07025020  07598100  07646640  07687600  07693500  07835500
       08178100 *09028000* 09275300  09300000  09302000  09407020  09416001  10072000  10075000  13310050  13310525
       13310580  13348100  13633100  13646000  13649000  13780002  14060000  14310500  14461500  14469100  14493500
```

```
        15091100  15116000  15301100  15342300  16159100  16318500  17001000  17094250  17094500
OPTOG  --  BOOLEAN  --  DECLARED IN SEGMENT 102 AT 08903000
       *08908000* 08911000  08913000
OROP  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01273000
        02965000  02984000
OTHERS  --  LABEL  --  DECLARED IN SEGMENT 57 AT 06060100  --  OCCURS AT 06064000
        06061000
OUTDEC  --  LABEL  --  DECLARED IN SEGMENT 143 AT 14014000  --  OCCURS AT 14159000
        14019000
OUTPRO  --  BOOLEAN  --  DECLARED IN SEGMENT 103 AT 08929000
       *08969000* 08977000  08979000
OUTPROCHECK  --  BOOLEAN PROCEDURE  --  DECLARED IN SEGMENT 3 AT 05395000
      *05396100**05400000* 08909000  08969000
OUTPUTSOURCE  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 02190500
        02196500  02230500  02235000
OUTPUT1  --  PROCEDURE  --  DECLARED IN SEGMENT 170 AT 17016000
        17045000
OUTPUT2  --  PROCEDURE  --  DECLARED IN SEGMENT 172 AT 17087000
        17096900  17119000
OUTV  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01289000
OVER  --  LABEL  --  DECLARED IN SEGMENT 132 AT 13196320  --  OCCURS AT 13196610
        13196370  13196530
OWNERR  --  LABEL  --  DECLARED IN SEGMENT 143 AT 14013000  --  OCCURS AT 14135000
        14018000
OWNTOG  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 172 AT 17047300
        17047400  17048100
OWNV  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01280000
        13456000  13720000
P  --  REAL  --  DECLARED IN SEGMENT 3 AT 01489000
       *14335000* 14387000  14430000  14450000  14463000
P  --  INTEGER  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 02635000
        02644000 *02917000* 02920000  02923000  02924000
P  --  REAL  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 03074000
P  --  REAL  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 03075000
P  --  REAL  --  DECLARED IN SEGMENT 129 AT 12102000
       *12108000* 12117000  12121000  12122000  12123000 *12123500* 12125000  12126000
PACK  --  STREAM PROCEDURE  --  DECLARED IN SEGMENT 3 AT 04270000
        04276000  04296000
PACKALPHA  --  STREAM PROCEDURE  --  DECLARED IN SEGMENT 119 AT 10100000
        10105000  10146000
PACKIN  --  LABEL  --  DECLARED IN SEGMENT 120 AT 10258300  --  OCCURS AT 10264910
        10264000  10264200  10264400  10264420  10264820
PACKINFO  --  STREAM PROCEDURE  --  DECLARED IN SEGMENT 121 AT 10236000
        10240000  10251000  10254000
PAN  --  FORMAT  --  DECLARED IN SEGMENT 117 AT 09409000
        09417000
PANA  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 06413000  --  FORWARD AT 03021000
        06065000  06067000  06068000  06071000  06073000  06120200  06233000  06417000  07116000
PARAMTOG  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 172 AT 17047300
        17047400  17049410
PARENCHECK  --  LABEL  --  DECLARED IN SEGMENT 73 AT 07461750  --  OCCURS AT 07471500
        07463750  07465500  07467000
PARENCOUNTER  --  DEFINE  --  DECLARED IN SEGMENT 83 AT 07655000
        07659500  07683500  07694000  07695500
PARSE  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 06312000  --  FORWARD AT 03014000
        06035000  06129000  06174000  06273000  06336700
PASFILE  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 05157000
```

```
        05181000  08732000  08795000  08852000
PASSALPHA  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 15020000  --  FORWARD AT 03075000
        07420000  07514000  07617000  10889000  15009000  15038000  15323000
PASSFILE  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 05137000
        05147000  06073300  06503000  07111000  07261000  07468250  07814400  07832000  07848000  08408000  08493570
        08515000  08605000  08911000  08920000  08968000  08974000  13066000
PASSFORMAT  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 05214000
        05227000  07248000  08441000  08646000
PASSLIST  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 05187520
        05187630  13196540
PASSLIST  --  LABEL  --  DECLARED IN SEGMENT 92 AT 08383000  --  OCCURS AT 08474000
        08433000  08437075  08443000
PASSLIST  --  LABEL  --  DECLARED IN SEGMENT 97 AT 08586000  --  OCCURS AT 08683000
        08652000  08653160  08653230  08657000
PASSMONFILE  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 05148000
        05156000  07421000  07515000  07619000  10890000  15010010  15010020  15324000
PASSN  --  LABEL  --  DECLARED IN SEGMENT 125 AT 10734000  --  OCCURS AT 10827000
        10852000
PASSPARLABL  --  LABEL  --  DECLARED IN SEGMENT 91 AT 08236000
PASSTYPE  --  INTEGER PROCEDURE  --  DECLARED IN SEGMENT 3 AT 15014000  --  FORWARD AT 03074000
        07419000  10888000  15008000 *15019000* 15321000
PAY  --  REAL ARRAY  --  DECLARED IN SEGMENT 172 AT 17069500
        17083000  17091510  17094300  17094400  17094650  17094700  17095500  17095550  17096000  17096300  17096450
PCTR  --  INTEGER  --  DECLARED IN SEGMENT 70 AT 07050000
       *07078000* 07086000  07152000 *07236000* 07241000  07369000
PD  --  REAL  --  NAME PARAMETER  --  DECLARED IN SEGMENT 3 AT 10954100
        10954000 *10978500*
PDES  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01299000
        13772000
PDINX  --  INTEGER  --  DECLARED IN SEGMENT 3 AT 01368000
        05261000 *05266000**09030000* 09297000 *09299000* 09348000  13638000 *13642000*
PDPRT  --  REAL ARRAY  --  DECLARED IN SEGMENT 3 AT 01342000
       *05261000**09297000* 09349000 *13638000*
PERCENT  --  LABEL  --  DECLARED IN SEGMENT 27 AT 02637000  --  OCCURS AT 02731000
        02641000  02906000
PERIOD  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01264000
        06127000  06266000  06277000  07021000  07026000  07111200  08493580  15111000  15339000
PERMANENT  --  BOOLEAN  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 05333000
        05331000  05345000
PINFOO  --  INTEGER  --  DECLARED IN SEGMENT 143 AT 14026000
       *14375000* 14471000  14483000  14489000
PJ  --  REAL  --  DECLARED IN SEGMENT 3 AT 01617000
       *13343000* 14073000  14085000  14116300  14117000 *14120000**14323200* 14333100  14334000  14335000  14349000
        14364000  14374000 *14387000**14391000* 14451400  14455000  14507010  14507180 *16046000* 16052000 *16074000*
PLACE  --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 119 AT 10100000
        10102000  10104000
PLUG  --  PROCEDURE  --  DECLARED IN SEGMENT 62 AT 06283000
        06284000  06285800  06288000
PLUG  --  PROCEDURE  --  DECLARED IN SEGMENT 89 AT 08020000
        08116000  08188000  08213000  08218000
PNCH  --  FILE  --  DECLARED IN SEGMENT 3 AT 01561005
        05057000  05058000  05059000
POINTER  --  INTEGER  --  DECLARED IN SEGMENT 134 AT 13219000
       *13222000* 13223000  13226000  13229000 *13240000* 13249000  13255000  13257000  13263000  13279000  13281000
        13283500  13284000  13286000 *13287000* 13290000
POP  --  REAL ARRAY  --  DECLARED IN SEGMENT 3 AT 01371000
       *02560000* 04605000
PORS  --  INTEGER  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 05325010
```

```
        05325460  05325470  05325480
PORS  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 54 AT 05325040
        05325050  05325090
POSITION  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 04270000
        04271000  04273000
POWERALL  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01579350
        12041130  12064000
POWERSOFTEN  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01574000
        07420000  08401060  08402000  08493860  08515000  08597600  08605000  10889000  15009000  15322000
PRIMARY  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 06079000  --  FORWARD AT 03005000
        06016000  06018000  06020000  06023000  06035000  06049000  06129000  06131000  10927000
PRINTCARD  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 02182500
        02195000  02195750  02214200  02227500  02228750  07025020
PRINTDOLLARBIT  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01001060
        02228750  02377000
PRINTDOLLARTOG  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01001350
        02228750
PRINTER  --  DEFINE  --  DECLARED IN SEGMENT 177 AT 17091000
PRINTI  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01573000
        07423000  07517000  07620000  10891200  15012000  15271100  15328000  15361000  15362000  15374100
PRINTSEGNO  --  FORMAT  --  DECLARED IN SEGMENT 4 AT 01800000
        13633100  13633200
PRINTSIZE  --  FORMAT  --  DECLARED IN SEGMENT 4 AT 01801000
        13649000  13650000
PRINTXREFSTATISTICS  --  PROCEDURE  --  DECLARED IN SEGMENT 172 AT 17053300
        17055200  17119200
PRL  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01666000
        07471250
PROAD  --  INTEGER  --  DECLARED IN SEGMENT 143 AT 14032000
       *14313000**14320000* 14387000  14476000  14489000
PROADO  --  REAL  --  DECLARED IN SEGMENT 3 AT 01613000
        14036000 *14476000*
PROCEDUREDEC  --  LABEL  --  DECLARED IN SEGMENT 143 AT 14015000  --  OCCURS AT 14270000
        14020000  14137000
PROCID  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01187000
        05390000  05396000  06091000  07399000  07404000  07463000  13278000  14284000  14292000  14294100
PROCSTMT  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 07391000  --  FORWARD AT 03029000
        06104000  06247000  07425000  07736000
PROCV  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01293000
PROGDFSCBLDR  --  REAL PROCEDURE  --  DECLARED IN SEGMENT 3 AT 05245000  --  FORWARD AT 03047000
       *05266000* 07111330  07130000  07203000  07270000  07305000  07646660  08860300  09273000  09296000  09313000
        10348000  10615000  10801000  10978500  13252000  13261000  13772000  13784000  14387000  14575000  14601000
        16049000  16075000
PROGRAM  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 09003000
        09420000  17000000
PROINFO  --  REAL  --  DECLARED IN SEGMENT 3 AT 01608000
        06062380  12036200  12062800  14304000 *14314000**14320000* 14391000  14392000  14454000  14457000  14469100
        14470000 *14496000* 15080000
PRT  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 05237000
        05237500  05238500
PRT  --  REAL  --  DECLARED IN SEGMENT 81 AT 07646390
       *07646450* 07646660
PRT  --  INTEGER  --  DECLARED IN SEGMENT 101 AT 08860100
       *08860250**08860300* 08860750
PRT  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 104 AT 09010000
        09011000  09013000
PRT  --  REAL ARRAY  --  DECLARED IN SEGMENT 115 AT 09317000
```

```
        09354100  09402000  09407000  09407010
PRT  --  INTEGER  --  DECLARED IN SEGMENT 141 AT 13776000
       *13783000* 13784000  13788000  13807000
PRTAD  --  REAL  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 13734000
        13731000  13732000  13772000
PRTADR  --  INTEGER  --  DECLARED IN SEGMENT 115 AT 09318000
       *09350000* 09354100  09355000
PRTBIT  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01001070
        02395000  05376100  09300000
PRTE  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01667000
        04018000  04023000  04033000  05316000  07122520  07529200  08068000  08489510  08694500
PRTI  --  INTEGER  --  DECLARED IN SEGMENT 3 AT 01703000
        05352000  05353000 *05367000* 05368000 *09039000* 14050000  14054000 *14607000*
PRTIMAX  --  INTEGER  --  DECLARED IN SEGMENT 3 AT 01703000
        05347000  05348000 *05350000* 05361000 *05368000**09039000* 09361120  09402000  09403000  09418000
PRTIO  --  INTEGER  --  DECLARED IN SEGMENT 143 AT 14030000
       *14054000* 14607000
PRTOG  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01001360
        05376100  09300000
PTOG  --  BOOLEAN  --  DECLARED IN SEGMENT 3 AT 01592000
        13310580  13339100  13342000  13370000 *14332000**14373000**14388000*
PUNCH  --  STREAM PROCEDURE  --  DECLARED IN SEGMENT 46 AT 05050000
        05056000  05057000
PUNCHBIT  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01001080
        02506000  05048000
PUNCHTOG  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01001370
        05048000
PURGE  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 13215000  --  FORWARD AT 03068000
        13767000  13771000  14471000  14483000  14569000  14573000
PURGEBIT  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01001090
        02508000
PURGETOG  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01001380
PURPT  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01628000
        13285000  13309000  14088000  14118000  14446000
PUT  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 05008000  --  FORWARD AT 01718200
        02723500  05132000  07522000  07545000  07610000  09214520  10625000  10649000  10651000  10656014  10656015
        10880000  10885000  10911000  10913000  10929000  10978500  12121000  13304000  13365000  13556000  13561000
        13788000  13807000  14109000  14116000  14117000  14178000  14211000  14239000  14245000  14259100  14259150
        14259155  14265950  14269600  14310000  14334000  14353000  14391000  14430000  14450000  14457000  14470000
        16074000  16138000  16166000  16167000  16172000  16257000  16260000
PUTNBUMP  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 13589000  --  FORWARD AT 03057000
        04178000  04179000  09034000  09214990  10576000  10583000  10586000  10656030  10660000  13547000  13789000
        13791000  13795000  14195000  14222000  14240000  14319000  14331000  14403000  16051000  16053000  16054000
        16443000  16445000  16446000
PUTOGETHER  --  PROCEDURE  --  DECLARED IN SEGMENT 120 AT 10234000
        10257000  10263900  10268000  10269000  10270000  10273000  10274000
PUTSEQNO  --  STREAM PROCEDURE  --  DECLARED IN SEGMENT 3 AT 02005000
        02006000  02013517  02013716  02202060  02214250  02218000  02234750  02238582  02368000  02716000  02904000
P1  --  INTEGER  --  NAME PARAMETER  --  DECLARED IN SEGMENT 3 AT 03033000
        03032000
P1  --  REAL  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 03057000
P1  --  BOOLEAN  --  DECLARED IN SEGMENT 64 AT 06313000
       *06319500* 06328000 *06329000* 06331700 *06332200* 06332500
P1  --  INTEGER  --  NAME PARAMETER  --  DECLARED IN SEGMENT 3 AT 10185000
        10184000 *10189000* 10190000
P1  --  REAL  --  NAME PARAMETER  --  DECLARED IN SEGMENT 3 AT 15070000
        15102000  15104000  15114000  15119000  15236000  15262500  15263000  15263050  15272900  15305000  15319000
        15329000 *15331000* 15334000  15335000 *15336000* 15342100  15349000  15364000 *15364500* 15367000 *15369100*
```

```
        15369300 *15369600*
P2  --  BOOLEAN  --  DECLARED IN SEGMENT 3 AT 01588000
       *13080000* 13108000  13341000  13342000  13345000  13347000 *13457000* 13500000  13506000  13541000  13553000
        13719000 *13720000**14133000**14318000**14321000*
P2  --  INTEGER  --  NAME PARAMETER  --  DECLARED IN SEGMENT 3 AT 03033000
        03032000
P2  --  BOOLEAN  --  DECLARED IN SEGMENT 64 AT 06313000
       *06329000* 06331500 *06332200* 06333500
P2  --  INTEGER  --  NAME PARAMETER  --  DECLARED IN SEGMENT 3 AT 10185000
        10184000 *10191000* 10192000
P3  --  BOOLEAN  --  DECLARED IN SEGMENT 3 AT 01589000
       *13080000* 13108000 *13461000* 13532000  13552500  13553000  13721000 *13722000**14133000**14138000**14139000*
       *14140000**14141000*
P3  --  BOOLEAN  --  DECLARED IN SEGMENT 64 AT 06313000
       *06329700**06332200* 06334000
P4  --  BOOLEAN  --  DECLARED IN SEGMENT 3 AT 01589500
       *13080000**13467000* 13552500  13553000  13723000 *13724000**14133000*
Q   --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 2 AT 00523000
        00524000  00526000
Q   --  ALPHA  --  DECLARED IN SEGMENT 3 AT 01316000
       *02013180**02238832* 02313000  02314000 *02327000* 02343000 *02371000* 02375000  02376000  02378000  02382000
        02384000  02386000  02389000  02392000  02394000  02396000  02400000  02406000  02419100  02419200  02420000
        02422000 *02424000* 02433000  02462000  02467000  02469000  02471000  02473000  02477000  02479000  02481000
        02495000  02497000 *02499000* 02505000  02507000  02509000  02515000  02519000  02521000  02523000  02529000
        02566000  02570000  02574100  02576000  02588000 *02590000**02650000**02697600**02698000* 02700000 *02754000*
       *02770000**02822000**02846000**02865000* 02867000  02879000 *02942000**02952000**02958500* 02960500  02961000
        02964000  02964500  02965000  02965500  05043000  05044000  07690000  07803000  07810100  07811000  07821100
        07822000  07824000  07825500  07826000  07827000  07828000  07843000  07859010 *08450010* 08493160  10565100
        10565200 *13282000* 13909000  13910000 *14441000*
Q   --  INTEGER  --  DECLARED IN SEGMENT 5 AT 01822000
        01833000  01834000
Q   --  REAL ARRAY  --  NAME PARAMETER  --  DECLARED IN SEGMENT 3 AT 04231000
        04239000  04241000  04243000
Q   --  REAL  --  DECLARED IN SEGMENT 42 AT 04311020
       *04311050**04311070* 04311090
Q   --  INTEGER  --  DECLARED IN SEGMENT 44 AT 04601000
       *04604000* 04605000  04606000
Q   --  BOOLEAN  --  DECLARED IN SEGMENT 55 AT 05343000
       *05352000* 05355000  05357000 *05363000**05371000**05374000* 05376100
Q   --  DEFINE  --  DECLARED IN SEGMENT 89 AT 08015000
        08093000  08094000  08185000  08200000  08210000  08218000
Q   --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 130 AT 12154000
        12159000
QUIT --  LABEL  --  DECLARED IN SEGMENT 102 AT 08904000  --  OCCURS AT 08926000
        08905000  08909000  08912000  08914000  08915000  08916000  08917000  08922000  08924000
QUIT --  LABEL  --  DECLARED IN SEGMENT 103 AT 08931000  --  OCCURS AT 08997000
        08966000  08969000  08972000  08975000  08983000  08987000  08988000  08990000  08991000  08993000
QUOTE --  LABEL  --  DECLARED IN SEGMENT 27 AT 02637000  --  OCCURS AT 02690000
        02641000
R   --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 02183500
        02183750  02185750  02186400
RA  --  DEFINE  --  DECLARED IN SEGMENT 119 AT 10110000
        10167000
RANGE  --  BOOLEAN PROCEDURE  --  DECLARED IN SEGMENT 3 AT 05297000
       *05304000* 07463000  07464250  08403000  08439000  08457000  08508000  08598000  08644000  08665000  08727000
        08728100  08790000  08847000  10574000  10579000  10599000  10654000  10805000  10872000
RCA --  DEFINE  --  DECLARED IN SEGMENT 157 AT 16008000
```

```
        16046000  16077000
RD  --  DEFINE  --  DECLARED IN SEGMENT 119 AT 10110000
        10131000
RDS  --  DEFINE  --  DECLARED IN SEGMENT 103 AT 08931000
        08955000
RDV  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01668000
RE  --  DEFINE  --  DECLARED IN SEGMENT 119 AT 10111000
        10157000
READACARD  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 02196750  --  FORWARD AT 02065000
        02130500  02133500  02238000  02731000  02735000  02948000  09035000
READSTMT  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 08290000  --  FORWARD AT 03061000
        07740000  08493000
READTAPE  --  PROCEDURE  --  DECLARED IN SEGMENT 13 AT 02198500
        02202250  02208250  02208500  02219500  02223750
READXFORM  --  LABEL  --  DECLARED IN SEGMENT 92 AT 08385100  --  OCCURS AT 08442000
        08450020
REALARRAYID  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01203000
        07155000  07355000  12036400  13453000
REALDEC  --  LABEL  --  DECLARED IN SEGMENT 143 AT 14013000  --  OCCURS AT 14138000
        14018000  14135000  14136000  14136100
REALID  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01199000
        06062340  06286800  07111250  07145000  07193000  07343000  07381000  08038000  10926510  12035000  13469000
        14138000  14507130
REALPROCID  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01195000
        07318000  07324000  12036100
REALSTRPROCID  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01191000
        14294000
REALV  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01283000
        06117000  06234000
RECOUNT  --  REAL  --  DECLARED IN SEGMENT 3 AT 01561200
        02013497  02013510 *02013515**02013563**02013568* 02013570  02013670  02013705  02132000 *02199500* 02207750
       *02222000* 02224000  02733000
RECOV  --  LABEL  --  DECLARED IN SEGMENT 134 AT 13220000  --  OCCURS AT 13281000
        13251000  13256000
RED  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 02011000
REFCOUNT  --  DEFINE  --  DECLARED IN SEGMENT 172 AT 17069300
        17080100
REFIDNOF  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01007425
        02001700  17080000  17091800  17091900  17093850  17094050  17094100  17117000  17117300
REFTYPE  --  REAL  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 02001650
        02001640  02001645  02001695
REGO  --  OWN REAL  --  DECLARED IN SEGMENT 89 AT 08010000
       *08093000* 08094000 *08111000* 08120000 *08160000**08170000* 08185000  08200000 *08210000* 08218000  08231000
REL  --  BOOLEAN  --  DECLARED IN SEGMENT 2 AT 00504801
        06094950 *07464750* 15263050
RELAD  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 05237000
        05237500  05242000
RELAD  --  REAL  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 05259000
        05245000  05259000  05261000  05264500
RELAD  --  REAL  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 13734000
        13731000  13732000  13738000  13769000
RELAD  --  REAL  --  DECLARED IN SEGMENT 143 AT 14029000
       *14487000* 14489000 *14534000**14563000**14574000* 14575000
RELATION  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 06281000  --  FORWARD AT 03012000
        06240000  06290200  07111285  07211000  10325000
RELEASES  --  PROCEDURE  --  DECLARED IN SEGMENT 157 AT 16281000
        16289000  16488000
```

```
RELEASEV  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01233000
        08732200  08738000  08795200  08801000
RELOP  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01275000
        06144000  06239000  06287200  06288800  06290000  07111275  07211000  08493530  10325000  14260000  14269280
        16208000
RELSESTMT  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 07461250
        07472250  07753000
REMCOUNT  --  OWN INTEGER  --  DECLARED IN SEGMENT 120 AT 10230000
        10245000  10247000 *10248000**10249000**10263000*
REMEMBERSEQNO  --  REAL  --  DECLARED IN SEGMENT 154 AT 15075550
       *15085100* 15116000  15301100  15342300
REMOTE  --  FILE  --  DECLARED IN SEGMENT 3 AT 01561055
        05045900
REMOTOG  --  BOOLEAN  --  DECLARED IN SEGMENT 3 AT 01561060
        05036000  05044000  05045900 *09028100*
REPEAT  --  INTEGER  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 04011000
        04010000  04013000  04014000
REPEAT  --  REAL  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 119 AT 10088000
        10086000  10087000  10091000
REPEAT  --  INTEGER  --  DECLARED IN SEGMENT 119 AT 10106000
       *10112000**10113000**10113100* 10117000  10125000 *10127000* 10127100  10127200 *10134545**10134580* 10134590
        10136500 *10150140**10150185* 10150225 *10150280**10159100**10167800* 10171000
RESULT  --  INTEGER  --  DECLARED IN SEGMENT 3 AT 01386000
       *02013195**02013200**02013290**02013318**02013330**02013335* 02013340 *02013365* 02013370 *02013390* 02013395
       *02013420**02013446* 02128500  02129000 *02238340* 02238370 *02238390**02238420* 02238430 *02238500**02327000*
        02366200 *02371000**02499000* 02500000  02579000  02586000 *02590000* 02591000  02598000 *02602700**02647000*
        02648000 *02663000**02666000**02667000**02694000**02698000**02700000**02744000**02747000**02749000**02751000*
       *02752000**02754000**02757000**02759000**02763000**02766000**02770000**02788000**02793000**02806000**02809000*
       *02812000**02816000**02820000* 02821000 *02846000**02889000**02890000**02938000**02942000**02951000**02958500*
        02959000  02959500 *10140000**10167500* 10264600  10265000  10272000  10276000
RESULT  --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 10 AT 02089500
        02092500
RESULT  --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 131 AT 13005000
        13010000
RESULTSWITCH  --  SWITCH LABEL  --  DECLARED IN SEGMENT 27 AT 02643000
        02648000
RESULTV  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 10 AT 02089500
        02090500  02094000
RETURNSTORE  --  OWN REAL  --  DECLARED IN SEGMENT 89 AT 08010000
        08087000  08090000 *08166000**08168000**08190000* 08193000 *08207000* 08215000
REVERSETOG  --  DEFINE  --  DECLARED IN SEGMENT 92 AT 08374000
        08387000  08389000  08391000  08420000
REWINDV  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01221000
RF  --  DEFINE  --  DECLARED IN SEGMENT 119 AT 10111000
        10155000  10161000
RI  --  DEFINE  --  DECLARED IN SEGMENT 119 AT 10110000
        10153000
RINX  --  REAL  --  DECLARED IN SEGMENT 135 AT 13301000
       *13302000* 13304000  13305000
RL  --  DEFINE  --  DECLARED IN SEGMENT 119 AT 10111000
        10169000
RLEFT  --  DEFINE  --  DECLARED IN SEGMENT 119 AT 10108000
        10117000
RMT  --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 45 AT 05016000
        05024000 *05024500*
RO  --  DEFINE  --  DECLARED IN SEGMENT 119 AT 10111000
        10129000
```

```
ROSE  -- LABEL  -- DECLARED IN SEGMENT 27 AT 02639000  --  OCCURS AT 02875000
      02879000  02881000
ROUND  -- LABEL  -- DECLARED IN SEGMENT 79 AT 07594000  --  OCCURS AT 07645000
      07600000  07604000
ROW  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 02001020
      02001030  02001090
ROW  --  INTEGER  --  DECLARED IN SEGMENT 55 AT 05344000
     *05352000**05360000* 05363000  05367000
RP  --  LABEL  --  DECLARED IN SEGMENT 58 AT 06090000  --  OCCURS AT 06125000
      06120000
RR  --  DEFINE  --  DECLARED IN SEGMENT 119 AT 10109500
      10155500  10161000
RRB1  --  BOOLEAN  --  DECLARED IN SEGMENT 3 AT 01522000
     *05169000* 05178000 *08387000**08389000* 08391000  08420000 *08606000* 08609000 *08738000* 08740000
RRB2  --  BOOLEAN  --  DECLARED IN SEGMENT 3 AT 01525000
     *05169000* 05172000  05176000
RRIGHT  --  DEFINE  --  DECLARED IN SEGMENT 119 AT 10107000
      10127200
RR1  --  REAL  --  DECLARED IN SEGMENT 2 AT 00507000
     *00529000**00530000* 01557000 *05168000* 05170000  05175000 *07614000* 07619000 *07622000* 07625000  07626000
     *07627000* 07633000 *07659500**07683500**07694000* 07695500 *08412010* 08412070  08413030  08413050  08413100
     *08481000**08619080* 08619090  08619095  08620000  08621000  08621010  08621020 *08688000**08737000* 08747000
     *08800000* 08817000 *08857000* 08858000 *10598000**10615000* 10621000  10622000  10626000 *10796000* 10891000
     *14329000* 14373000 *15244000*
RR10  --  REAL  --  DECLARED IN SEGMENT 2 AT 00507000
     *00539000* 01561005 *15034000* 15035000
RR11  --  REAL  --  DECLARED IN SEGMENT 2 AT 00507000
     *00529000* 00531000  00533000  00536000  00539000 *15033000* 15037000
RR2  --  REAL  --  DECLARED IN SEGMENT 2 AT 00507000
     *00530000* 01557000 *05167000* 07660000  07663500 *07664500* 07665500 *07666500* 07672000  07673500 *07674000*
      07675000  07677500 *07679500**07694000* 07696500  07697500  08737000  08800000  08857000 *10583000**10584000*
      10590000 *10656010* 10656014  10656015 *10656030**10657000* 10658000 *10808000* 10827000  10868000 *10881000*
      10883000 *10914000**10915000* 10916000  10927000 *15244000* 15250000
RR3  --  REAL  --  DECLARED IN SEGMENT 2 AT 00507000
    *00531000**00532000* 01559000 *07696500* 07697000 *10584000**10586000* 10587000 *10592000**10658000**10660000*
     10661000 *10808000* 10823000  10848000 *10906000* 10928000 *13795000* 13798000 *15244000* 15245000 *15246000*
     15250000
RR4  --  REAL  --  DECLARED IN SEGMENT 2 AT 00507000
    *00532000* 01559000 *10585000**10598500**10603000**10617000* 10626000 *10644000* 10651000  10652000 *10658000*
     10659000 *10802000**10880000* 10885000  10894000 *10913000* 10919000 *10921000**13789000* 13795000 *13797000*
     13798000 *15246000* 15248000  15249000
RR5  --  REAL  --  DECLARED IN SEGMENT 2 AT 00507000
    *00534000**00535000* 01560000 *10584000* 10587000 *10590000* 10634000 *10657000* 10661000 *10814000* 10818000
     10819000  10820000  10821000 *10833000**10839000* 10843000  10844000  10845000  10846000 *10869000**10871000*
     10891100 *10926000* 10929000
RR6  --  REAL  --  DECLARED IN SEGMENT 2 AT 00507000
    *00534000**00535000* 01560000 *10582000* 10590000 *10625000* 10627000  10634000 *10801000* 10926000  10928000
RR7  --  REAL  --  DECLARED IN SEGMENT 2 AT 00507000
    *00534000**00535000* 01560000 *10569000* 10576000  10583000  10649000  10657000 *10800000* 10929000
RR8  --  REAL  --  DECLARED IN SEGMENT 2 AT 00507000
    *00537000**00538000* 01561000 *10803000* 10814000  10826000  10835000  10839000  10868000  10874000  10883000
     10885000  10886000  10888000  10889000 *10911000* 10912000  10912100  10916000
RR9  --  REAL  --  DECLARED IN SEGMENT 2 AT 00507000
    *00537000**00538000* 01561000 *07599000* 07601000  07611000  07617000 *10813000* 10814000  10823000 *10838000*
     10839000  10848000 *15035000* 15037000
RSA  --  DEFINE  --  DECLARED IN SEGMENT 151 AT 14419100
      14451400
RSA  --  DEFINE  --  DECLARED IN SEGMENT 157 AT 16018000
```

```
RSCALE  --  DEFINE  --  DECLARED IN SEGMENT 119 AT 10109500
        10134590
RSTROKE  --  DEFINE  --  DECLARED IN SEGMENT 119 AT 10109000
        10122000  10126000
RT  --  DEFINE  --  DECLARED IN SEGMENT 119 AT 10110010
        10165505
RTBRKET  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01265000
        05274300  05278000  06314250  06325000  06329400  06330700  06332300  06422000  08277000  08417010  08493650
        08633000  08698500  10264300  10298000  10306000  10318000  10321000  10592000  10630000  10828000  12126000
        13088105  13514000  13549000  13569000  14259120  14269680  15260000  15294000
RTN  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01669000
        13744000
RTPARFN  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01266000
        02976500  06062220  06062430  06072110  06072260  06073650  06073860  06125000  06261000  06416000  07096000
        07111255  07186000  07205000  07238000  07471750  07683500  07838000  07852000  07854000  07859080  07909000
        07943000  08418700  08422000  08445000  08465000  08489580  08520000  08606000  08613600  08637000  08647000
        08673000  08696550  08729000  08732100  08749000  08795100  08818000  08853000  08860650  08924000  08994080
        08995500  08996000  10264300  10305000  10320000  10368000  10668000  10863000  10902000  12039000  12063600
        12126000  13182000  14259120  14336000  15369400  16132000  16282000  16478000
RTPARFN  --  LABEL  --  DECLARED IN SEGMENT 27 AT 02639000  --  OCCURS AT 02744000
        02642000
RTS  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01670000
        07111315  07138000  07272000  10329000  10361000  10616000  10906000  10985000  10992000  10995000  13264000
        13268000
RU  --  DEFINE  --  DECLARED IN SEGMENT 119 AT 10109500
        10150225
RV  --  DEFINE  --  DECLARED IN SEGMENT 119 AT 10109500
        10167200
RWNDSTMT  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 08825000  --  FORWARD AT 03066000
        07745000  08860000
RX  --  DEFINE  --  DECLARED IN SEGMENT 119 AT 10110000
        10164000
S  --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 7 AT 02001760
        02001780
S  --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 3 AT 02001838
        02001842
S  --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 3 AT 02001858
        02001868
S  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 02183500
        02183750  02186300
S  --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 17 AT 02264400
        02264500
S  --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 17 AT 02264700
        02264800
S  --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 29 AT 02927240
        02927300
S  --  BOOLEAN  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 03051001
        03051000
S  --  BOOLEAN  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 03067000
S  --  INTEGER  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 03100000
S  --  REAL  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 04278000
        04277000  04278000  04279200  04280000  04281000  04284200  04284300  04284400  04285000  04286000  04287000
S  --  REAL  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 04291000
        04296000  04297000
S  --  INTEGER  --  DECLARED IN SEGMENT 43 AT 04502000
        *04548000* 04551000
S  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 53 AT 05309000
```

```
          05311000
S  --  SWITCH LABEL  --  DECLARED IN SEGMENT 57 AT 06061000
          06062000
S  --  SWITCH LABEL  --  DECLARED IN SEGMENT 58 AT 06085000
          06091000
S  --  SWITCH LABEL  --  DECLARED IN SEGMENT 61 AT 06219000
          06225000
S  --  BOOLEAN  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 62 AT 06283000
          06283800
S  --  BOOLEAN  --  DECLARED IN SEGMENT 67 AT 06391000
          *06393000* 06397200  06398000
S  --  SWITCH LABEL  --  DECLARED IN SEGMENT 70 AT 07072000
          07216000
S  --  SWITCH LABEL  --  DECLARED IN SEGMENT 85 AT 07719000
          07728000
S  --  BOOLEAN  --  NAME PARAMETER  --  DECLARED IN SEGMENT 89 AT 08027000
          *08029000*
S  --  BOOLEAN  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 89 AT 08063000
          *08065000* 08069000
S  --  BOOLEAN  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 89 AT 08072000
          08075000  08079000
S  --  BOOLEAN  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 119 AT 10089000
          10086000  10087000  10098000
S  --  BOOLEAN  --  DECLARED IN SEGMENT 119 AT 10106000
          *10127200* 10171000
S  --  REAL  --  DECLARED IN SEGMENT 127 AT 12004000
          *12006000**12025000**12030000* 12039100 *12055000*
S  --  REAL  --  DECLARED IN SEGMENT 128 AT 12060200
          *12061300**12062200* 12063610 *12064900*
S  --  REAL  --  DECLARED IN SEGMENT 129 AT 12102000
          *12103000**12122000**12127000* 12128100 *12144900*
S  --  REAL  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 157 AT 16029000
          16033000  16034000 *16038000*
S  --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 172 AT 17047200
          17047700 *17049700* 17050000  17050100
SAVADDRSF  --  REAL  --  DECLARED IN SEGMENT 131 AT 13039450
          *13081500* 13160000  13166000
SAVECARD  --  INTEGER  --  DECLARED IN SEGMENT 3 AT 01561570
          02224100 *02238720*
SAVEDIM  --  REAL  --  DECLARED IN SEGMENT 138 AT 13449000
          *13473000**13532000* 13533000  13534000
SAVEINFO  --  REAL  --  DECLARED IN SEGMENT 138 AT 13449000
          *13470000* 13558000  13560000  13561000 *13563000**13566000*
SAVEINFO  --  REAL  --  DECLARED IN SEGMENT 145 AT 14165000
          *14175000* 14178000  14179000  14180000
SAVEINFO  --  REAL  --  DECLARED IN SEGMENT 148 AT 14269060
          *14269260* 14269600  14269620
SAVEINF02  --  REAL  --  DECLARED IN SEGMENT 138 AT 13449000
          *13471000* 13556000  13560000  13563000 *13567000*
SAVEINX  --  INTEGER  --  DECLARED IN SEGMENT 22 AT 02324000
          *02331000**02342000* 02352000
SAVEINX  --  INTEGER  --  DECLARED IN SEGMENT 21 AT 02365200
          *02525006* 02525007
SAVEIT  --  LABEL  --  DECLARED IN SEGMENT 148 AT 14269100  --  OCCURS AT 14269590
          14269450
SAVEL  --  REAL  --  DECLARED IN SEGMENT 3 AT 01623000
          13606000 *13628000* 14055000 *14609000*
SAVELO  --  INTEGER  --  DECLARED IN SEGMENT 143 AT 14030000
          *14055000* 14609000
```

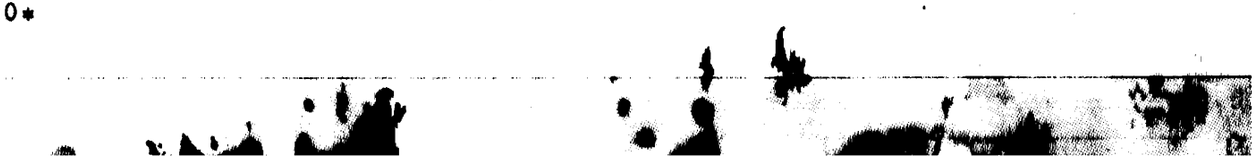
```
SAVEPRTOG  --  BOOLEAN  --  DECLARED IN SEGMENT 3 AT 01411020
       13641000  13645000 *14059100**14610100*
SAVEPRTOGO  --  BOOLEAN  --  DECLARED IN SEGMENT 143 AT 14034100
      *14059100* 14610100
SAVERR  --  LABEL  --  DECLARED IN SEGMENT 143 AT 14013000  --  OCCURS AT 14136000
       14018000
SAVETIME  --  INTEGER  --  DECLARED IN SEGMENT 2 AT 00503000
       01556200  09407050
SAVETIMES  --  PROCEDURE  --  DECLARED IN SEGMENT 167 AT 17002015
       17002525  17091180  17117910
SAVEV  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01281000
       08739000  08809000  13459000  13722000  14295100
SAVL  --  REAL  --  DECLARED IN SEGMENT 158 AT 16031000
      *16032000* 16040000  16047000
SAVL  --  REAL  --  DECLARED IN SEGMENT 160 AT 16087000
      *16088000* 16096000
SB  --  REAL  --  DECLARED IN SEGMENT 148 AT 14269060
      *14269446**14269520* 14269560  14269600
SBIT  --  BOOLEAN  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 07047000
       07045000  07046000  07117000  07127000  07171000  07226000  07249000  07347000  07359000  07382000
SBITF  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01154200
       05273300  05274100  06313650  06314150  14269446  14269600
SBV  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01244000
       16197000  16415000
SCAN  --  STREAM PROCEDURE  --  DECLARED IN SEGMENT 10 AT 02089500
       02127000  02128500
SCANAGAIN  --  LABEL  --  DECLARED IN SEGMENT 27 AT 02639000  --  OCCURS AT 02646000
       02650000  02697600  02728000  02737000  02768000  02890000  02908000
SCANNER  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 02066000  --  FORWARD AT 01730000
       02013195  02013200  02013290  02013318  02013330  02013335  02013365  02013390  02013420  02013446  02137000
       02238340  02238390  02238420  02238500  02327000  02371000  02499000  02590000  02647000  02663000  02666000
       02694000  02698000  02744000  02747000  02749000  02751000  02752000  02754000  02759000  02763000  02766000
       02770000  02788000  02793000  02806000  02809000  02812000  02816000  02820000  02889000  02890000  02938000
       02942000  02951000  02958500  10140000  10167500
SCATTERFLBAT  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 13197000  --  FORWARD AT 03035000
       06396000  07079000  07395000  13208000  13327000  13908000  14206000  14305000  14507110
SCLASS  --  INTEGER  --  DECLARED IN SEGMENT 70 AT 07050000
      *07081000* 07084000 *07087000* 07088000  07089000 *07090000* 07117000  07150000  07151000  07229000  07230000
      *07282000* 07330000
SCNN  --  REAL STREAM PROCEDURE  --  DECLARED IN SEGMENT 15 AT 02238130
       02238150  02238340  02238390  02238420  02238500
SCOUNT  --  INTEGER  --  DECLARED IN SEGMENT 21 AT 02365100
      *02366200* 02602700
SCOUNT  --  INTEGER  --  DECLARED IN SEGMENT 152 AT 14507030
      *14507050* 14507160
SCRAM  --  INTEGER  --  DECLARED IN SEGMENT 3 AT 01301000
      *02865000* 02874000 *12164000* 13308000  13361000  13362000  14259102  14507050 *14507060* 14507070 *14507160*
SCV  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01246000
       16206000  16337000
SEARCH  --  ALPHA PROCEDURE  --  DECLARED IN SEGMENT 3 AT 04231000
      *04243000* 04244000  04603000
SEARCHLIB  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 02013165
       02013720  02133000  02229000  02230750  02735000  02908000
SEC  --  DEFINE  --  DECLARED IN SEGMENT 157 AT 16022000
       16342000
SECOND  --  INTEGER  --  NAME PARAMETER  --  DECLARED IN SEGMENT 3 AT 05271000
       05270000 *05273400**05274200**05281000* 05282000
```

```
SECOND  --  INTEGER  --  DECLARED IN SEGMENT 64 AT 06312500
      *06313700**06314200**06321500* 06323500 *06326500* 06331800  06333500  06336200  06336300
SECOND  --  INTEGER  --  DECLARED IN SEGMENT 65 AT 06339000
       06341000  06343000
SECRET  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01628000
       13331000  14333000  14345000
SED  --  DEFINE  --  DECLARED IN SEGMENT 151 AT 14419000
       14455000
SED  --  DEFINE  --  DECLARED IN SEGMENT 157 AT 16017000
       16286000
SEEKTOG  --  BOOLEAN  --  DECLARED IN SEGMENT 92 AT 08385600
      *08387000**08390500* 08391000  08415020  08418650
SEG  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 172 AT 17047200
       17047400  17049600
SEGDICT  --  REAL ARRAY  --  DECLARED IN SEGMENT 115 AT 09317000
       09351000 *09355000**09358000* 09359000  09399000
SEGMENT  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 13634000  --  FORWARD AT 03042000
       07646670  07697500  08860400  09308000  09315000  13656000  13814000  14577000
SEGMENTSTART  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 13632000  --  FORWARD AT 03041000
       07646640  07693500  13683300  13780002  14060000
SEGMNT  --  INTEGER  --  DECLARED IN SEGMENT 115 AT 09318000
      *09350000* 09351000  09352000  09353000  09355000 *09357000* 09358000  09359000
SEGNOF  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01007405
       13310450  17094800
SEGSBIT  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01001100
       02470000  07646640  07693500  13646000  13780002  14060000
SEGSIZEMAX  --  INTEGER  --  DECLARED IN SEGMENT 3 AT 01687000
      *13643000*
SEGSTOG  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01001390
       07646640  07693500  13646000  13780002  14060000
SEMICOLON  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01230000
       05109000  07012000  07032000  07646480  07646525  10264410  10672000  10926520  10935000  13816000  14062000
       14063000  14269840  14325000  14338000  14357000  14519000  16131000
SEQ  --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 3 AT 01756000
       01759100
SEQ  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 04130000
       04132000
SEQBIT  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01001110
       02191500  02234800  02393000
SEQCOMPARE  --  PROCEDURE  --  DECLARED IN SEGMENT 13 AT 02202500
       02209750  02224500  02224750
SEQERRBIT  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01001120
       02226250  02520000
SEQERRTOG  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01001420
       02226250
SEQLOC  --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 3 AT 02013060
       02013080  02013090
SEQNO  --  REAL  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 02001650
       02001640  02001645  02001695
SEQNO  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 172 AT 17047200
       17047400  17050400
SEQNO  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 172 AT 17051400
       17051500  17052100
SEQNOF  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01007430
       17092600  17093400  17094800  17095750  17095900  17117702  17117708
SEQSUM  --  INTEGER  --  DECLARED IN SEGMENT 3 AT 01561120
       02013525  02013615 *02013645* 02013713  02201000
SEQTOG  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01001400
```

```
        02191500  02234800
SEQUENCEWARNING  --  STREAM PROCEDURE  --  DECLARED IN SEGMENT 3 AT 01742100
        02227000
SEQXEQTOG  --  BOOLEAN  --  DECLARED IN SEGMENT 3 AT 01001560
       *02331060**02525008**02525012* 02525013 *09028100*
SERR  --  FORMAT  --  DECLARED IN SEGMENT 118 AT 09414101
        09416002
SES  --  DEFINE  --  DECLARED IN SEGMENT 151 AT 14419000
        14454000
SETTING  --  BOOLEAN  --  DECLARED IN SEGMENT 3 AT 01000802
       *02327000* 02338000  02352000  02356000  02427000  07084000
SETUPHEADING  --  STREAM PROCEDURE  --  DECLARED IN SEGMENT 172 AT 17047200
        17051200  17094700
SFILENO  --  INTEGER  --  DECLARED IN SEGMENT 3 AT 01755000
        08945000 *08946000* 08956000  08985000
SFN  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 103 AT 08932000
        08933000 *08937000*
SGAVL  --  INTEGER  --  DECLARED IN SEGMENT 3 AT 01369000
        07646650 *07646680* 07693000  07697500 *07699000* 08860200 *08860400**09030000**09295000**09312000* 09361180
        09399000  09399150  09400000  09419000  13633100  13633200  13780000  13781000 *13814000**14061000*
SGNO  --  INTEGER  --  DECLARED IN SEGMENT 3 AT 01370000
        02195000  02195750  02214200  02227500  02228750  05262000  05264500  07025020  07646640 *07646650* 07646670
       *07646680* 07697500 *08860200* 08860400 *08860450**09030000**09295000* 09298100  09300200 *09312000* 09315000
        13310450  13779000 *13781000* 13814000 *13818000* 14054000 *14061000* 14577000  14596000
SGNO  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 05237000
        05237500  05243000
SGNO  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 104 AT 09010000
        09011000  09021000
SGNOO  --  INTEGER  --  DECLARED IN SEGMENT 143 AT 14030000
       *14054000* 14577000  14596000  06061000
SIGN  --  BOOLEAN  --  DECLARED IN SEGMENT 62 AT 06282600  --  OCCURS AT 06068000
       *06286400* 06288600
SIGNA  --  BOOLEAN  --  DECLARED IN SEGMENT 62 AT 06282600
        06285800 *06286000**06286200* 06288000
SIGNA  --  OWN BOOLEAN  --  DECLARED IN SEGMENT 89 AT 08012000
       *08056000* 08065000  08098000  08108000  08130000  08179000  08181000  08189000  08212000  08217000 *08227000*
SIGNB  --  OWN BOOLEAN  --  DECLARED IN SEGMENT 89 AT 08012000
        08050000  08112000 *08121000* 08126000  08183000  08196000 *08203000* 08214000
SIGNC  --  OWN BOOLEAN  --  DECLARED IN SEGMENT 89 AT 08012000
        08044000 *08112000* 08185000  08197000 *08202000* 08219000 *08220000*
SIMPARITH  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 06032000  --  FORWARD AT 03003000
        06006000  06277000  07111270  07210000  10324000  15369500
SIMPBOO  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 06168000  --  FORWARD AT 03009000
        06145000  07111260  07111290  07212000  10326000
SIMPGO  --  BOOLEAN PROCEDURE  --  DECLARED IN SEGMENT 3 AT 07548000  --  FORWARD AT 03026000
       *07555000* 07557000  07564000  07570000  07581000  07646515
SIMPI  --  BOOLEAN PROCEDURE  --  DECLARED IN SEGMENT 89 AT 08052000
       *08061000* 08176000  08211000  08224000
SIMPLF  --  BOOLEAN  --  DECLARED IN SEGMENT 62 AT 06282600
        06285800 *06287200* 06287400
SIMPLF  --  BOOLEAN PROCEDURE  --  DECLARED IN SEGMENT 89 AT 08027000
       *08033000**08038000* 08039000  08112000  08181000  08183000  08185000
SIMPLFB  --  DEFINE  --  DECLARED IN SEGMENT 89 AT 08014000
        08044000  08112000  08220000
SIMPLFV  --  DEFINE  --  DECLARED IN SEGMENT 89 AT 08015000
        08074000  08099000  08211000  08224000  08227000
SINGLBIT  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01001130
```

```
        02013179  02013181  02195000  02195750  02196590  02214200  02227250  02227500  02228750  02522000  04150000
        04279100  04284100  05039600  05046000  05265000  05325490  07025020  09302000  10075000  13633100  13649000
        17094250  17094500
SINGLTOG  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01001430
        02013179  02013181  02195000  02195750  02196590  02214200  02227250  02227500  02228750  04150000  04279100
        04284100  05039600  05046000  05265000  05325490  07025020  09302000  10075000  13633100  13649000  17094250
        17094500
SIV  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01236000
        16319000  16320000  16328000  16334000
SIZ  --  INTEGER  --  DECLARED IN SEGMENT 28 AT 02762000
       *02767000*
SIZE  --  REAL  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 02927110
        02927100  02927110  02927380  02927470  02927490  02927580  02927610  02927612  02927620  02927660  02927690
        02927700
SIZE  --  REAL  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 03044000
        03042000  03043000
SIZE  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 05317000
        05318000  05324000
SIZE  --  REAL  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 13636000
        13634000  13635000  13639000  13640000 *13642000* 13643000  13644000  13645000  13649000  13650000  13654000
SIZEALPHA  --  DEFINE  --  DECLARED IN SEGMENT 153 AT 15025000
        15035000  15037000
SK  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 2 AT 00515000
        00516000  00517000  00521000
SKAN  --  DEFINE  --  DECLARED IN SEGMENT 15 AT 02238220
        02238340  02238390  02238420  02238500
SKAN  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 02277000
        02327000  02371000  02499000  02590000  02958500
SKANAGAIN  --  LABEL  --  DECLARED IN SEGMENT 21 AT 02360000  --  OCCURS AT 02370000
        02378000  02383000  02385000  02389000  02465000  02468000  02478000  02503000  02527000  02584000  02595000
        02599000
SKIP  --  REAL  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 119 AT 10088000
        10086000  10087000  10092000
SKIP  --  INTEGER  --  DECLARED IN SEGMENT 119 AT 10106000
       *10116000* 10127000  10127200 *10139000* 10143000  10146000 *10147000**10150125**10150190* 10150225 *10153000*
       *10161500**10162000**10165000**10170000* 10171000
SKIP  --  LABEL  --  DECLARED IN SEGMENT 177 AT 17091100  --  OCCURS AT 17096850
        17094150
SKIPCOUNT  --  INTEGER  --  DECLARED IN SEGMENT 121 AT 10241000
       *10247000* 10253000  10254000  10256000
SKIPIT  --  DEFINE  --  DECLARED IN SEGMENT 34 AT 02974000
        02974500  02978500
SKIPS  --  PROCEDURE  --  DECLARED IN SEGMENT 157 AT 16403000
        16419000  16492000
SKIPV  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01241000
SKIP1  --  LABEL  --  DECLARED IN SEGMENT 64 AT 06313500  --  OCCURS AT 06332500
        06313750  06314500  06329800
SKIP2  --  LABEL  --  DECLARED IN SEGMENT 64 AT 06313500  --  OCCURS AT 06331100
        06327000
SKP  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 02001831
        02001833
SKP  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 02001838
        02001842
SKP  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 02001858
        02001890
SKSC  --  LABEL  --  DECLARED IN SEGMENT 120 AT 10259000  --  OCCURS AT 10263500
        10283000
SLB  --  LABEL  --  DECLARED IN SEGMENT 138 AT 13452000  --  OCCURS AT 13547000
```

```
        13580000  13584000
SND  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01671000
        06062190  06072070  06288800  07418000  07923000  08993500  10360000  10837000  10919000  13761000  15007000
        15104000  15318000  15320000  15335000
SOP  --  BOOLEAN  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 14003000
        14001000  14002000  14036000  14059100  14554000  14597000
SORCE  --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 2 AT 00515000
       *00518000* 00521000
SORTA  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01580000
        08911100  08955000
SORTI  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01579100
        08926000  08997000
SORTMERGETOG  --  BOOLEAN  --  DECLARED IN SEGMENT 3 AT 01786000
SORTSTMT  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 08928000  --  FORWARD AT 03081000
        07826000  08998000
SPAC  --  REAL  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 05259000
        05245000  05259000 *05260000* 05262000  05264500  05266000
SPACEITDOWN  --  DEFINE  --  DECLARED IN SEGMENT 143 AT 14023100
        14461500  14493500
SPACESTMT  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 08494000  --  FORWARD AT 03063000
        07742000  08527000
SPACEV  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01218000
SPCLMON  --  BOOLEAN  --  DECLARED IN SEGMENT 156 AT 15211000
       *15242000* 15326000 *15349000* 15351000  15354000  15360000
SPECIAL  --  REAL ARRAY  --  DECLARED IN SEGMENT 3 AT 01003000
        02655000  02666000  02755000  02961500  02963000  07031000 *09198000*
SPECIALCHAR  --  LABEL  --  DECLARED IN SEGMENT 27 AT 02637000  --  OCCURS AT 02651000
        02643000
SPECIALMATH  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01579355
        12061750
SPECIALSWITCH  --  SWITCH LABEL  --  DECLARED IN SEGMENT 27 AT 02641000
        02659000
SPECTOG  --  BOOLEAN  --  DECLARED IN SEGMENT 3 AT 01593000
       *05034000* 13048000  13050000  13077000  13082000  13193000  13196350  13196370  13305100  13310300  13310500
        13328000 *13330650* 13348100  13457000  13462000  13467000  13470000  13478000  13550000  13565000  13601000
        13621000  13717000  13819400  14070000 *14122000* 14142000  14149000  14160000  14161000  14169000  14187000
        14205000  14217000  14275000  14287000  14296000  14298000 *14364000**14368000* 14377000  14507000 *14507180*
SPMON  --  DEFINE  --  DECLARED IN SEGMENT 156 AT 15195000
        15242000
SPMONADR  --  DEFINE  --  DECLARED IN SEGMENT 156 AT 15207000
        15248000  15249000
SPRT  --  BOOLEAN ARRAY  --  DECLARED IN SEGMENT 3 AT 01698000
       *05348000* 05349000  05352000  05363000
SRESULT  --  INTEGER  --  DECLARED IN SEGMENT 21 AT 02365100
       *02366200* 02602700
SSCRAM  --  INTEGER  --  DECLARED IN SEGMENT 152 AT 14507030
       *14507050* 14507160
SSN  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01671100
        08418600  08613200  08653125  08653195  08924000  08993000
SSP  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01672000
        05226000  06067000  08415010  08418600  08613200  08627000  08653110  08653140  08653210
ST  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 46 AT 05050000
        05051000  05054000
STACKCT  --  REAL  --  DECLARED IN SEGMENT 3 AT 01568500
       *04078500* 04527000 *06053500**06098000**06124500**06128000**06195500**06260500**06267000**06289200**06296500*
       *06299500**06305500**06336600**06410000**07079500**07385000**07465250**07727100**14125500**15091000**15098000*
       *15101500**15217500**15234500**15299500**15376000*
```



```
STACKCTR -- REAL -- DECLARED IN SEGMENT 3 AT 01683000
      05370000 *05371000* 13748000  14058000  14477000 *14478040**14493000* 14535000  14536000 *14610000*
STACKCTR0 -- REAL -- DECLARED IN SEGMENT 143 AT 14029000
      *14058000**14477000* 14493000  14610000
STACKHEAD -- REAL ARRAY -- DECLARED IN SEGMENT 3 AT 01310000
      02874000  05233000 *05234000**09188000* 09214520 *09214530**13286000**13308000* 13361000 *13362000**14259102*
      *14448000* 14507070
STARS -- STREAM VARIABLE -- VALUE PARAMETER -- DECLARED IN SEGMENT 172 AT 17051400
      17051500  17052000  17052400
START -- REAL -- VALUE PARAMETER -- DECLARED IN SEGMENT 90 AT 08084000
      08082000  08083000  08086000
START -- LABEL -- DECLARED IN SEGMENT 124 AT 10550000 -- OCCURS AT 10565000
      10678000
START -- LABEL -- DECLARED IN SEGMENT 125 AT 10706000 -- OCCURS AT 10792000
      10933000
START -- LABEL -- DECLARED IN SEGMENT 131 AT 13040000 -- OCCURS AT 13195000
      13050000  13072000  13082000  13145000  13164000
START -- LABEL -- DECLARED IN SEGMENT 138 AT 13452000 -- OCCURS AT 13588000
      13564000
START -- LABEL -- DECLARED IN SEGMENT 143 AT 14017000 -- OCCURS AT 14063000
      14138000  14139000  14140000  14141000  14148000  14155000  14158000  14159000  14160000  14162000  14186000
      14199000  14253000  14269000  14269980  14379000  14504000  14507190  14521000
START -- LABEL -- DECLARED IN SEGMENT 145 AT 14166000 -- OCCURS AT 14184000
      14170000
START -- LABEL -- DECLARED IN SEGMENT 146 AT 14202000 -- OCCURS AT 14251000
      14218000
START -- LABEL -- DECLARED IN SEGMENT 147 AT 14254050 -- OCCURS AT 14269000
      14259040  14259075
START -- LABEL -- DECLARED IN SEGMENT 149 AT 14272000 -- OCCURS AT 14373000
START -- LABEL -- DECLARED IN SEGMENT 150 AT 14384000 -- OCCURS AT 14503000
START -- LABEL -- DECLARED IN SEGMENT 157 AT 16475000 -- OCCURS AT 16477000
      16485000
STARTCALL -- LABEL -- DECLARED IN SEGMENT 125 AT 10714000 -- OCCURS AT 10803000
      10900000
STARTINTRSC -- DEFINE -- DECLARED IN SEGMENT 104 AT 09024000
      09291000
START1 -- LABEL -- DECLARED IN SEGMENT 149 AT 14272000 -- OCCURS AT 14375000
      14325000  14364000
START2 -- LABEL -- DECLARED IN SEGMENT 149 AT 14273000 -- OCCURS AT 14376000
      14301000
STATS -- SWITCH FORMAT -- DECLARED IN SEGMENT 174 AT 17053500
      17054500  17054800  17055020
STATUS -- LABEL -- DECLARED IN SEGMENT 57 AT 06060000 -- OCCURS AT 06073200
      06061000
STD -- DEFINE -- DECLARED IN SEGMENT 3 AT 01673000
      04614000  06062435  07111310  07134000  07306000  07470750  07626000  07859070  07923000  07946000  08090000
      08728600  08954000  08955000  10302000  10328000  10622000  10922000  12038000  12061100  12061200  12063300
      13069000  13196570  13230000  13915000  14181000  14233000  14545000  14565000  15104000  15335000
STFPI -- INTEGER PROCEDURE -- DECLARED IN SEGMENT 3 AT 05003000
      05108000  05273100  05273900  05274300  05276000  05277000  05278000  05385000  06062120  06072030  06072200
      06073200  06073250  06118000  06286200  06287200  06314050  06314250  06321000  06330500  06415000  06421000
      06507000  07018000  07079000  07462000  07462500  07505000  07552000  07567000  07597000  07645000  07646440
      07646480  07646595  07661000  07680500  07684500  07754000  07814000  07829000  07831000  07834000  07844000
      07846000  07854000  07859020  07901000  07924000  07925000  07939000  08017600  08029000  08113000  08178000
      08240000  08243000  08386000  08401000  08412040  08412050  08413130  08414000  08416010  08418250  08431000
      08437000  08479000  08493530  08592000  08597100  08620010  08620020  08622010  08623000  08629000  08637000
      08647000  08653125  08653195  08687000  08698200  08738000  08801000  08908000  08967000  09252000  10593000
      10666000  10672000  10809000  10892000  10965000  10970000  12007000  12010000  12013000  12060500  13064000
```

```
          13088060  13092000  13188000  13196520  13351000  13474000  13564000  13791000  13911300  14126000  14259110
          14268000  14269320  14269480  14269500  14338000  14341000  14345000  14356000  14393000  14396000  14467000
          15090000  15113000  15233000  15253000  15254700  15260000  15300000  15342000  16125000  16160000  16197000
          16206000  16208000  16209000  16217000  16252000  16253000  16282000  16316000  16319000  16324000  16326000
          16326100  16340000  16367000  16368000  16372000  16384000  16406000  16434000  16435000
STEPIT  -- PROCEDURE --  DECLARED IN SEGMENT 3 AT 05002000  --  FORWARD AT 01741000
          05146000  05174000  05187620  05386000  06015000  06020000  06035000  06049000  06062120  06062150  06062165
          06062195  06062230  06062350  06062385  06062430  06072040  06072060  06072110  06072220  06072240  06072260
          06072270  06073830  06073860  06110000  06113000  06120000  06124000  06126000  06129000  06152000  06172000
          06189000  06256000  06259000  06262000  06270000  06286400  06288600  06303000  06316500  06323000  06326000
          06331000  06336500  06345000  06403000  06410000  06411000  06416000  06417000  06422000  06423000  06508000
          07010000  07215500  07244000  07283000  07407000  07441000  07463750  07470500  07471750  07485000  07487000
          07493000  07495000  07505000  07509000  07552000  07569000  07573000  07580000  07584000  07646430  07664000
          07668500  07691000  07808000  07813000  07841000  07851000  07857000  07859040  07859080  07903000  07907000
          07913000  07925000  07952000  08031000  08120000  08131000  08135000  08142000  08155000  08161000  08175000
          08192000  08231000  08245000  08281000  08391000  08412000  08412090  08413018  08414010  08414018  08416010
          08416018  08417010  08418200  08418350  08418800  08444000  08456000  08474000  08489570  08493060  08493625
          08493650  08493730  08502000  08507000  08519000  08592000  08611000  08613200  08613300  08618000  08621000
          08621004  08624000  08624004  08629000  08629004  08643000  08650000  08653125  08653195  08656000  08664000
          08678000  08696540  08697000  08698400  08698600  08721000  08726000  08729000  08748000  08753000  08784000
          08789000  08816000  08822000  08841000  08846000  08857000  08905000  08920000  08924000  08965000  08994020
          08995000  08996000  10297000  10298000  10309000  10356000  10368000  10573000  10591000  10606000  10678000
          10796000  10803000  10812000  10836000  10907000  10922000  10926000  10933000  10964000  10989000  12013000
          12017000  12025000  12030000  12034000  12036000  12036300  12040000  12045000  12054000  12060550  12061300
          12062200  12062400  12062600  12062900  12063610  12064400  12064800  13080000  13087000  13088085  13088090
          13088100  13088120  13096000  13111000  13148000  13327000  13805000  13907000  13911200  13917000  14174000
          14193000  14214000  14242000  14259000  14259070  14259140  14269220  14269380  14269640  14269705  14269840
          14314000  14359000  14403000  14472000  14601700  15098000  15254600  15274000  15306000  16130000  16176000
          16220000  16234000  16376000  16408000  16412000  16494000
STEPV  -- DEFINE --  DECLARED IN SEGMENT 3 AT 01255000
          08103000  08182000
STLABID  -- DEFINE --  DECLARED IN SEGMENT 3 AT 01178000
          14403000  14447000  16051000  16224000  16253000  16437000  16443000  16481000
STLB  -- REAL --  DECLARED IN SEGMENT 3 AT 01471000
          07165000  07201000 *07364000* 07365000  07366000 *15254400**15254700* 15255000  15262600  15273000
STMT  -- PROCEDURE --  DECLARED IN SEGMENT 3 AT 07714000  --  FORWARD AT 03027000
          05108000  07011000  07485000  07495000  07574000  07577000  07586000  07646520  07757000  08163000  08201000
          14488000  14515000
STMTSTART  -- OWN REAL --  DECLARED IN SEGMENT 89 AT 08010000
          08157000 *08169000**08191000* 08193000 *08206000* 08221000
STOP  -- LABEL --  DECLARED IN SEGMENT 150 AT 14384000  --  OCCURS AT 14491000
          14484000
STOPDEFINE  -- BOOLEAN --  DECLARED IN SEGMENT 3 AT 01597000
          02893000 *02911000**02926000**07017000**07646595**08493050**10263300**10281000**12111000**13079500**13088050*
         *13327000**13805000**14129000**14192000**14258000**14259060**14259104**14259130**14269200**14402000**14403000*
STOPENTRY  -- BOOLEAN --  DECLARED IN SEGMENT 3 AT 01594000
         *13048000**13061000**13077000**13189000**13196350**13196490* 13351000 *13780000**13817000**14171000**14183000*
         *14188000**14198000**14205000**14217000**14255000**14269000**14269120**14269940**14318000**14321000**14402000*
         *14404000*
STOPGSP  -- BOOLEAN --  DECLARED IN SEGMENT 3 AT 01694000
          13342000  13344000 *13783000**13817000**14171000**14183000**14188000**14198000**14205000**14217000**14255000*
         *14269000**14269120**14269940**14402000**14404000*
STOPOS  -- STREAM VARIABLE --  NAME PARAMETER --  DECLARED IN SEGMENT 3 AT 02001020
          02001375
STOPPER  -- REAL --  VALUE PARAMETER --  DECLARED IN SEGMENT 3 AT 03068000
STOPPER  -- REAL --  VALUE PARAMETER --  DECLARED IN SEGMENT 3 AT 13217000
          13215000  13216000  13223000  13291000
```

```
STOPPER  --  REAL  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 13734000
       13731000  13732000  13767000  13771000
STOPPOINT  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01561090
       02013495  02013600
STORE  --  LABEL  --  DECLARED IN SEGMENT 70 AT 07065000  --  OCCURS AT 07134000
       07208000  07212000
STORE  --  PROCEDURE  --  DECLARED IN SEGMENT 89 AT 08063000
       08070000  08130000  08135000  08138000  08216000
STORE  --  REAL  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 90 AT 08084000
       08082000  08083000  08089000
STORE  --  LABEL  --  DECLARED IN SEGMENT 122 AT 10292000  --  OCCURS AT 10328000
       10322000
STOREFIX  --  INTEGER  --  DECLARED IN SEGMENT 90 AT 08091000
      *08093000* 08157000  08171000
STORESUBS  --  LABEL  --  DECLARED IN SEGMENT 124 AT 10557000  --  OCCURS AT 10592000
       10629000
STR  --  REAL  --  DECLARED IN SEGMENT 87 AT 07922000
      *07923000* 07926000
STREAMERR  --  LABEL  --  DECLARED IN SEGMENT 143 AT 14017000  --  OCCURS AT 14137000
       14021000
STREAMSTMT  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 16001000  --  FORWARD AT 03040000
       07011000  14415000  16130000  16229000  16235000  16495000
STREAMTOG  --  BOOLEAN  --  DECLARED IN SEGMENT 3 AT 01416500
       02691000  02697500  02776500  04086000  04279000 *06502000* 07011000 *07689000**07691000**07698500* 10260050
      *10260100**10284000* 13310580  13339100  13342000  14077000  14082000 *14280000* 14324000  14361000  14377000
       14385000 *14461000*
STREAMV  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01297000
       14278000
STREAMWORDS  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 05230000
       05236000  14324000  14460000
STRING  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01278050
       02705100  02776600  02783000  07669500  07671000  16384000
STRMPROCSTMT  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 07427000  --  FORWARD AT 03030000
       06101000  06244000  07455000  07737000
STRNGCON  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01211000
       02702000  02776200  06286600  07669500  08017700  08032000  13089000  13092000  16210000  16384000
STRNGXT  --  LABEL  --  DECLARED IN SEGMENT 27 AT 02638000  --  OCCURS AT 02701000
STRPROCID  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01189000
       07441000  14281000
STRTPOS  --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 3 AT 02001020
       02001180
STUFF  --  STREAM PROCEDURE  --  DECLARED IN SEGMENT 12 AT 02196530
       02196580
STUFFBIT  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01001140
       02516000
STUFFF  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 05199000
       05203000  07111330  07138000  07272000  07633000  10367000  10621000  14228000
STUFFILE  --  STREAM PROCEDURE  --  DECLARED IN SEGMENT 103 AT 08932000
       08939000  08949000
STUFFTOG  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01001440
ST1  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 10 AT 02090000
       02090500 *02120000* 02124500 *02126000*
ST2  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 10 AT 02090000
       02090500 *02121000* 02121500
SUB  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01674000
       06071000  06286200  06329550  08030000  08044000  08214000  12015000  13544000  14544000  16329000
SUBC  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 13213000
       13492000  13502000  13579000  13584000
```

```
SUBLEVEL -- INTEGER -- DECLARED IN SEGMENT 3 AT 01402000
        04611000  05117000  06374000 *14477000**14492000*
SUBOP -- DEFINE -- DECLARED IN SEGMENT 3 AT 13211000
        13487000  13576000
SUBSCRIPT -- DEFINE -- DECLARED IN SEGMENT 124 AT 10507000
        10598000  10615000  10621000  10622000  10626000
SUBSCTR -- DEFINE -- DECLARED IN SEGMENT 125 AT 10789000
        10813000  10814000  10823000  10838000  10839000  10848000
SUBSLOOP -- LABEL -- DECLARED IN SEGMENT 125 AT 10736000 -- OCCURS AT 10836000
        10857000
SUM -- STREAM VARIABLE -- NAME PARAMETER -- DECLARED IN SEGMENT 3 AT 02013060
        02013074  02013076  02013120  02013130  02013136
SUPERFILEID -- DEFINE -- DECLARED IN SEGMENT 3 AT 01185000
        05141000  05169000  06073250  06092005  06225000  06227500  07107000  07467750  07685500  07729190  07814300
        07831000  07846000  08403000  08508000  08598000  08727000  08790000  08847000  08908000  08967000  08973000
        13049000
SUPERFRMTID -- DEFINE -- DECLARED IN SEGMENT 3 AT 01183000
        05221000  05224000  07105000  08439000  08644000  13787000  14091000
SUPERLISTID -- DEFINE -- DECLARED IN SEGMENT 3 AT 01278500
        05187580  07099000  07122500  07215000  07217500  08479000  08485000  08687000  08689000  13196360  13196520
SUPERMOVER -- DEFINE -- DECLARED IN SEGMENT 3 AT 01577500
        07953000
SUPERSTACK -- REAL ARRAY -- DECLARED IN SEGMENT 3 AT 01310000
        02865000 *09196100*
SVARMONFILE -- DEFINE -- DECLARED IN SEGMENT 3 AT 01529000
        15010020
SVTOG -- BOOLEAN -- DECLARED IN SEGMENT 136 AT 13323010
       *13330600* 13330650
SW -- SWITCH LABEL -- DECLARED IN SEGMENT 11 AT 02191250
        02194500
SW -- SWITCH LABEL -- DECLARED IN SEGMENT 22 AT 02326000
        02328000
SWITCHDEC -- LABEL -- DECLARED IN SEGMENT 143 AT 14015000 -- OCCURS AT 14200000
        14020000  14252000
SWITCHGEN -- BOOLEAN PROCEDURE -- DECLARED IN SEGMENT 3 AT 10954000
      *10976000**10998000* 12001000  14223000
SWITCHID -- DEFINE -- DECLARED IN SEGMENT 3 AT 01186000
        06393000  07507000  07728000  10642000  13253000  14217000
SWITCHIT -- PROCEDURE -- DECLARED IN SEGMENT 21 AT 02321000
        02359000  02377000  02388000  02393000  02395000  02397000  02402000  02412000  02415000  02419100  02419200
        02421000  02425000  02435000  02470000  02472000  02474000  02480000  02487000  02490000  02496000  02506000
        02508000  02511000  02516000  02520000  02522000  02531000  02567000  02574200  02577000  02600000
SWITCHV -- DEFINE -- DECLARED IN SEGMENT 3 AT 01292000
        13045000  13193000  13778000  14161000  14167000
SWITMONFILE -- DEFINE -- DECLARED IN SEGMENT 3 AT 01539000
        07516000
SYLLABLE -- REAL -- VALUE PARAMETER -- DECLARED IN SEGMENT 3 AT 03019000
SYLLABLE -- REAL -- VALUE PARAMETER -- DECLARED IN SEGMENT 3 AT 03028000
SYLLABLE -- STREAM VARIABLE -- VALUE PARAMETER -- DECLARED IN SEGMENT 3 AT 04270000
        04271000  04274000
SYMBOL -- STREAM VARIABLE -- VALUE PARAMETER -- DECLARED IN SEGMENT 3 AT 02183500
        02183750  02185250
T -- REAL -- DECLARED IN SEGMENT 3 AT 01564000
       *02655000* 02658000 *02666000**02691000* 02695000 *02701000**02702000**02705100**02705200**027210[illegible] 02721500
       *02755000**02776100**02776200**02776600**02798000**02813000* 02814000 *02823000* 02824000 *02826000* 02827000
        02842000 *02848000**02850000**02851000**02865000* 02867000 *02869000**02873000**02874000* 02875000 02876000
       *02877000* 02877010 *02877020**02878000**02882000* 02882200  02886000  02896000 *02897000* 02901000 02910000
```

```
          02910200  02910400 *02927390* 02927400 *05233000* 05234000  0[illegible]  *08057000* 0805[illegible]  08[illegible]  [illegible]061000
         *08069000* 08070000
T   --   REAL  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 01717900
T   --   STREAM VARIABLE  --  DECLARED IN SEGMENT 3 AT 01736000
          01736100  01736200
T   --   REAL  --  DECLARED IN SEGMENT 16 AT 02249000
         *02251000**02257000**02259000* 02260000
T   --   BOOLEAN  --  DECLARED IN SEGMENT 22 AT 02323000
T   --   STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 23 AT 02441000
          02443000  02444000
T   --   STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 28 AT 02446000
          02448000  02449000
T   --   INTEGER  --  DECLARED IN SEGMENT 32 AT 02957500
         *02960500**02961000* 02961500 *02964000* 02966500
T   --   BOOLEAN  --  DECLARED IN SEGMENT 35 AT 02981500
         *02982500* 02983000  02983500  02984000  02984500
T   --   INTEGER  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 03050000
T   --   INTEGER  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 04311000
          04311030  04312000  04318000
T   --   LABEL  --  DECLARED IN SEGMENT 61 AT 06224000  --  OCCURS AT 06267000
          06250000  06256000
T   --   DEFINE  --  DECLARED IN SEGMENT 83 AT 07655500
          07660000  07663500  07664500  07665500  07666500  07672000  07673500  07674000  07675000  07677500  07679500
          07681000  07694000  07696500  07697500
T   --   INTEGER  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 07930000
          07936000  07945000  07947000
T   --   BOOLEAN  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 08493015
          08493010  08493015  08493050
T   --   REAL  --  DECLARED IN SEGMENT 127 AT 12004000
         *12030000**12038000**12044000* 12047000
T   --   REAL  --  DECLARED IN SEGMENT 128 AT 12060200
         *12060500**12061000* 12061100  12061200 *12062200**12062400**12063200**12063500**12064400* 12064500 *12064800*
         *12065000* 12065200
T   --   REAL  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 12101000
          12108000
T   --   STREAM VARIABLE  --  DECLARED IN SEGMENT 131 AT 13007000
          13008000  13009000  13015000
T   --   REAL  --  DECLARED IN SEGMENT 133 AT 13199000
         *13200000* 13201000  13202000  13203000  13204000  13205000  13206000  13207000
T   --   REAL  --  DECLARED IN SEGMENT 138 AT 13449000
         *13513000* 13514000 *13530000* 13531000  13531100  13532000  13537000 *13547000* 13549000 *13558000* 13559000
         *13560000* 13561000 *13569000*
T   --   REAL  --  DECLARED IN SEGMENT 159 AT 16067000
         *16070000* 16075000
T   --   INTEGER  --  DECLARED IN SEGMENT 165 AT 16364000
          16393000  16394000
TABLE  --  INTEGER PROCEDURE  --  DECLARED IN SEGMENT 3 AT 02635000
         *02920000**02924000* 02927000  05002000  05003000  05221000  05224000  05294000  06062135  06315500  06322000
          06330700  07095000  07157000  07185000  07186000  07198000  07374000  07401000  07465000  07662500  07687500
          07689500  07732000  07735000  07812000  07816000  07837000  07859050  07941000  07947000  08401030  08442000
          08460000  08597300  08668000  08728400  10263400  10282000  10304000  10318000  10588000  10592000  10926500
          10926510  12020000  12060600  12112000  13480000  13511000  13513000  13547000  13569000  13572000  13911100
          14063000  14480000  14516000  15254500  16225000  16479000
TAKE  --  REAL PROCEDURE  --  DECLARED IN SEGMENT 3 AT 05005000  --  FORWARD AT 01718100
          02001810  02001820  02722500  02723500  04179000 *05006000* 05124000  05131000  05189000  05391000  05392000
          05397000  05398000  05399000  05404000  05405000  05406000  07086000  07152000  07241000  07276000  07290000
          07361000  07364000  07369000  07412000  07422000  07442000  07515000  07520000  07539000  07601000  08942000
          08943000  09214520  10586000  10625000  10651000  10656014  10656015  10660000  10814000  10839000  10885000
```

```
        10911000  10929000  10978500  12117000  12121000  13110000  13226000  13229000  13249000  13255000  13257000
        13279000  13284000  13286000  13362000  13364000  13365000  13558000  13788000  13797000  13807000  14084000
        14088000  14099000  14104000  14116300  14118000  14178000  14210000  14211000  14239000  14245000  14259100
        14259102  14259150  14259155  14265950  14269600  14310000  14320000  14333100  14352000  14391000  14425000
        14430000  14438000  14446000  14448000  14454000  14470000  15010010  15010020  15035000  15233100  15242000
        15244000  15246000  15273100  15281000  15324000  16070000  16072000  16074000  16089000  16138000  16139000
        16163000  16166000  16172000  16254000  16260000  16449000
TAKEFRST  --  REAL PROCEDURE  --  DECLARED IN SEGMENT 3 AT 05188000
       *05189000* 05222000  07256000  10583000  10649000  10656030  10808000  10880000  10881000  10913000  10914000
        14213000
TALL  --  REAL  --  DECLARED IN SEGMENT 154 AT 15072000
       *15077000* 15080000 *15083000* 15085000  15086000  15092020  15094000  15097000  15099000  15101000  15102000
        15105000  15107000  15116000  15124000  15233100 *15236000* 15237000  15242000  15244000  15267000  15269000
        15273100  15279000  15281000  15290000  15301100  15303000  15307000  15311000  15316000  15321000  15323000
        15324000  15333000  15342300  15344000  15352000  15353000  15357000  15372000
TALL  --  REAL  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 156 AT 15217000
        15215000  15216000  15220000  15224000  15228000
TALLYV  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01239000
        16319000  16341000
TAN  --  DEFINE  --  DECLARED IN SEGMENT 157 AT 16014000
        16214000
TAPE  --  FILE  --  DECLARED IN SEGMENT 3 AT 01561000
        01835800  02201750  02452000  02454000  02457000  09407300
TAPE  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 02188000
        02188500
TAPELAST  --  LABEL  --  DECLARED IN SEGMENT 13 AT 02210000  --  OCCURS AT 02219250
        02210750
TBUFF  --  REAL ARRAY  --  DECLARED IN SEGMENT 3 AT 01561056
        02201750  02202000  02456000  02457000  07025030
TB1  --  BOOLEAN  --  DECLARED IN SEGMENT 3 AT 01457000
       *07275000* 07279000 *07287000* 07295000 *07304000* 07308000 *07340000**07506000**07507000* 07509000 *08860350*
       *10243000**10260000* 10275000 *13792000**13808000**13810000* 13815000
TB1  --  BOOLEAN  --  DECLARED IN SEGMENT 146 AT 14204000
       *14206000**14212000* 14223000  14231000  14246000
TB2  --  BOOLEAN  --  DECLARED IN SEGMENT 141 AT 13777000
       *13778000* 13785000
TCLASS  --  INTEGER  --  DECLARED IN SEGMENT 164 AT 16314000
       *16315000* 16317000  16324000  16328000  16334000  16341000 *16348000* 16349000  16350000  16352000
TCOUNT  --  INTEGER  --  DECLARED IN SEGMENT 3 AT 01566000
        02251000  02252000 *02758000**02790000* 02796000  02798000  02803000 *02810000**02817000**02942000*
TD  --  LABEL  --  DECLARED IN SEGMENT 61 AT 06224000  --  OCCURS AT 06266000
        06233000  06244000  06247000
TEDOC  --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 3 AT 01688010
        01688030  01688070  01688080
TEDOC  --  REAL ARRAY  --  DECLARED IN SEGMENT 81 AT 07646400
       *07646490**07646510**07646545* 07646565 *07646570* 07646630  07646690
TEDOC  --  REAL ARRAY  --  DECLARED IN SEGMENT 83 AT 07656500
        07694000  07698500
TEDOC  --  REAL ARRAY  --  DECLARED IN SEGMENT 101 AT 08860100
        08860150  08860450
TEDOC  --  REAL ARRAY  --  DECLARED IN SEGMENT 141 AT 13777500
        13777600  13818500
TEDOC  --  REAL ARRAY  --  DECLARED IN SEGMENT 143 AT 14024000
        14049000  14595000
TEMP  --  STREAM VARIABLE  --  DECLARED IN SEGMENT 3 AT 02001070
        02001100  02001300  02001320  02001500  02001510
TEMP  --  REAL  --  DECLARED IN SEGMENT 81 AT 07646390
```

```
        *07646565* 07646575 *07646640* 07646670  07646680 *07646720* 07646
TEMPL  -- INTEGER  -- DECLARED IN SEGMENT 39 AT 04166000
        *04172000* 04175000  04186000
TEMP1  -- STREAM VARIABLE  -- DECLARED IN SEGMENT 3 AT 02001860
         02001866  02001868  02001890
TEMP2  -- STREAM VARIABLE  -- DECLARED IN SEGMENT 3 AT 02001860
TEN  -- REAL ARRAY  -- DECLARED IN SEGMENT 3 AT 01340000
         02796000  02799000  02803000  02826000  02833000  02835000  02838000  02840000  02843000 *09041000* 09314000
TENIL  -- REAL ARRAY  -- DECLARED IN SEGMENT 143 AT 14024100
         14049000  14594000
TEST  -- PROCEDURE  -- DECLARED IN SEGMENT 89 AT 08041000
         08051000  08131000  08220000
TESTCODE  -- DEFINE  -- DECLARED IN SEGMENT 162 AT 16192000
         16211300  16212500  16212700
TESTLOC  -- DEFINE  -- DECLARED IN SEGMENT 125 AT 10766000
         10801000  10926000  10928000
TESTVARB  -- DEFINE  -- DECLARED IN SEGMENT 124 AT 10526000
         10584000  10587000  10590000  10634000  10657000  10661000
TESTVARB  -- DEFINE  -- DECLARED IN SEGMENT 156 AT 15188000
         15244000  15250000
TEXT  -- ALPHA ARRAY  -- DECLARED IN SEGMENT 3 AT 01481200
         02212250  02723500 *09251010* 10251000  10254000 *12117000*
THENBRANCH  -- INTEGER  -- DECLARED IN SEGMENT 63 AT 06295000
        *06297000* 06301000
THENUMBEROFDECLAREDDIMENSIONS  -- INTEGER  -- DECLARED IN SEGMENT 156 AT 15184100
        *15233100* 15255000  15291100  15296000
THENV  -- DEFINE  -- DECLARED IN SEGMENT 3 AT 01256000
         06411000  16217000
THERE  -- STREAM VARIABLE  -- NAME PARAMETER  -- DECLARED IN SEGMENT 3 AT 01737350
         01737450
THI  -- REAL  -- DECLARED IN SEGMENT 3 AT 01689000
        *02250000* 02255000  02256000  02257000  02785000  02797000  02799000  02800000
THIRD  -- INTEGER  -- DECLARED IN SEGMENT 64 AT 06312500
        *06313700**06314200**06326500**06330600* 06331100  06334200  06334600  06336200  06336300
THISL  -- DEFINE  -- DECLARED IN SEGMENT 98 AT 08715000
         08737000  08747000
THISL  -- DEFINE  -- DECLARED IN SEGMENT 99 AT 08774000
         08800000  08817000
THISL  -- DEFINE  -- DECLARED IN SEGMENT 100 AT 08835000
         08857000  08858000  06061000
TIME1  -- REAL  -- DECLARED IN SEGMENT 3 AT 01300000  -- OCCURS AT 06073000
         01828000  09417000 *17000000**17091165*
TIMINGS  -- REAL ARRAY  -- DECLARED IN SEGMENT 167 AT 17002012
        *17002040**17002075**17022300* 17054800  17055010 *17080100**17091520**17094410**17095510**17096350*
TL  -- BOOLEAN  -- DECLARED IN SEGMENT 9 AT 02013178
        *02013179* 02013181
TL  -- INTEGER  -- DECLARED IN SEGMENT 38 AT 04118000
        *04119000* 04124000
TL  -- INTEGER  -- DECLARED IN SEGMENT 74 AT 07483000
        *07485000* 07487000
TL  -- REAL  -- DECLARED IN SEGMENT 126 AT 10957000
        *10958000* 10978500  10994000  11000000
TL  -- REAL  -- DECLARED IN SEGMENT 134 AT 13221000
        *13258000* 13277000
TLCR  -- INTEGER  -- DECLARED IN SEGMENT 3 AT 01330000
        *02202050* 02202060  02202070  02219500  02224500 *02456000*
TLCR  -- REAL  -- NAME PARAMETER  -- DECLARED IN SEGMENT 13 AT 02202750
```

```
        02202500  02203500  02204000  02208500
TLEVEL  --  INTEGER  --  DECLARED IN SEGMENT 66 AT 06357000
       *06359000* 06360000  06365000  06369000 *06374000* 06376000
TLO  --  REAL  --  DECLARED IN SEGMENT 3 AT 01689000
       *02250000* 02255000  02256000  02785000  02796000  02797000  02799000  02800000
TOGGLEV  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01245000
TOLOC  --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 104 AT 09005000
        09006000
TOTALNO  --  INTEGER  --  DECLARED IN SEGMENT 3 AT 01000860
        02192000 *02192500**02193000* 02234800 *02582000**17004500* 17045000
TOV  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01257000
        07505000  07552000  16252000  16437000
TOWARDS  --  INTEGER  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 03037000
        03036000
TOWARDS  --  INTEGER  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 04116000
        04114000  04115000  04120100  04120200  04120500
TP  --  INTEGER  --  DECLARED IN SEGMENT 142 AT 13903100
       *13909000* 13914000
TRANS  --  STREAM LABEL  --  DECLARED IN SEGMENT 3 AT 01758000  --  OCCURS AT 01778000
        01763000
TRB  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01682000
        04318000  06334600
TRNSFER  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01251000
        16379000
TRP  --  DEFINE  --  DECLARED IN SEGMENT 157 AT 16019000
        16383000
TRUTHV  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01208000
TRW  --  DEFINE  --  DECLARED IN SEGMENT 151 AT 14419000
        14456000
TS  --  BOOLEAN  --  DECLARED IN SEGMENT 9 AT 02013178
       *02013179* 02013181
TSSTREAMTOG  --  BOOLEAN  --  DECLARED IN SEGMENT 120 AT 10259400
       *10260050* 10284000
TSUBLEVEL  --  REAL  --  DECLARED IN SEGMENT 143 AT 14029000
       *14477000* 14492000
TURNONSTOPLIGHT  --  STREAM PROCEDURE  --  DECLARED IN SEGMENT 3 AT 02011000
        02013717  02202070  02214500  02218250  02235750  02369000  02717000  02905000
TWO  --  LABEL  --  DECLARED IN SEGMENT 138 AT 13452000  --  OCCURS AT 13474000
        13547000
TYP  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 05237000
        05237500  05241000
TYPE  --  REAL  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 03055000
TYPE  --  REAL  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 05259000
        05245000  05259000  05262000  05264500
TYPE  --  INTEGER  --  DECLARED IN SEGMENT 61 AT 06214000
       *06227520**06259000* 06263000  06265000
TYPE  --  INTEGER  --  DECLARED IN SEGMENT 63 AT 06295000
       *06299000* 06304000  06305000
TYPE  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 131 AT 13020000
        13021000  13025000
TYPE  --  REAL  --  DECLARED IN SEGMENT 131 AT 13039000
       *13081000**13086000**13088030**13088035**13088040**13088060**13088075**13088080**13088[illegible]02000
TYPE  --  REAL  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 13316000
        13314000  13315000  13330550  13346000  13349000
TYPE  --  INTEGER  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 13714000
        13726000
TYPE  --  SWITCH LABEL  --  DECLARED IN SEGMENT 157 AT 16476000
```

```
        16477000
TYPE  --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 3 AT 1[illegible]
        17049300
TYPEDPROC  --  BOOLEAN  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 1[illegible]
        15001000  15002000  15010010
TYPEREF  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01007420
        02001695  07598100  07687600  07835500  08178100  13310525  14310500  15091100  [illegible]  18500  17092000
        17092300  17092900
TYPEV  --  REAL  --  DECLARED IN SEGMENT 3 AT 01611000
       *13453000**13469000* 13471000  13567000 *14091000**14094000* [illegible]5000 *14281000**14284000* [illegible]85000 *14292000*
       *14294000**14294100* 14295000  14301000  14308000  14319000
TYPEV  --  INTEGER  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 03071000
        03069000  03070000
T1  --  STREAM VARIABLE  --  DECLARED IN SEGMENT 3 AT 01688020
       *01688030* 01688090  01688100
T1  --  REAL  --  DECLARED IN SEGMENT 3 AT 01693000
       *04284200* 04284300 *04285000* 04286000 *14098000* 14099000  14104000  [illegible]000
T1  --  STREAM VARIABLE  --  DECLARED IN SEGMENT 3 AT 02001860
       *02001874* 02001876
T1  --  INTEGER  --  DECLARED IN SEGMENT 43 AT 04502000
       *04550000* 04551000
T1  --  REAL  --  DECLARED IN SEGMENT 57 AT 06059000
       *06062000**06062120**06062165**06062190**06062210**06062220* 06062285 *06062420**06062430* 06062540  06062565
        06066000 *06073350**06073400**06073500* 06073550 *06073600**06073700**06073830* 06073850
T1  --  REAL  --  DECLARED IN SEGMENT 70 AT 07076000
       *07095000* 07096000 *07125000**07133000* 07145000 *07204000* 07210000 *07211000**07276000* 07288000 *07319000*
       *07361000* 07362000  07369000
T1  --  REAL  --  DECLARED IN SEGMENT 78 AT 07562000
       *07566000* 07572000  07573000  07575000 *07577000* 07579000  07583000  07586000
T1  --  REAL  --  DECLARED IN SEGMENT 84 AT 07658000
       *07663500* 07664500  07665500
T1  --  REAL  --  DECLARED IN SEGMENT 89 AT 08173000
       *08179000**08193000* 08202000  08203000  08204000  08205000  08206000  08207000  08225000
T1  --  REAL  --  DECLARED IN SEGMENT 122 AT 10291000
       *10301000* 10330000
T1  --  REAL  --  DECLARED IN SEGMENT 126 AT 10957000
       *10981000* 10984000 *10999000* 12000000
T1  --  STREAM VARIABLE  --  DECLARED IN SEGMENT 131 AT 13007000
       *13009000* 13015000  13017000
T1  --  REAL  --  DECLARED IN SEGMENT 138 AT 13449000
       *13454000* 13456000  13459000  13466000  13469000 *13476000**13478000* 13487000  13499000  13576000
T1  --  REAL  --  DECLARED IN SEGMENT 154 AT 15073000
        15095000  15097000  15098000  15100000  15112000  15124000  15304000  15308000  15341000  15369000
T2  --  STREAM VARIABLE  --  DECLARED IN SEGMENT 3 AT 01688020
       *01688030* 01688050  01688060
T2  --  REAL  --  DECLARED IN SEGMENT 3 AT 01693000
       *14104000**14107000**14108000* 14109000
T2  --  STREAM VARIABLE  --  DECLARED IN SEGMENT 3 AT 02001860
       *02001874* 02001876
T2  --  INTEGER  --  DECLARED IN SEGMENT 43 AT 04502000
       *04550000*
T2  --  REAL  --  DECLARED IN SEGMENT 57 AT 06059000
       *06062000* 06062110  06062125  06062170  06062185  06062210  06062280  06062470  06062540  06072080
T2  --  REAL  --  DECLARED IN SEGMENT 70 AT 07076000
       *07129000* 07138000 *07152000* 07165000 *07194000**07201000**07270000* 07272000 *07288000* 07290000 *07319000*
       *07362000* 07364000
T2  --  REAL  --  DECLARED IN SEGMENT 78 AT 07562000
       *07583000**07585000* 07586000
```

```
T2  --  REAL  --  DECLARED IN SEGMENT 84 AT 07658000
       *07663500* 07664500  07665500  07666500 *07672000* 07673000  07674000
T2  --  REAL  --  DECLARED IN SEGMENT 89 AT 08173000
       *08198000* 08208000
T2  --  REAL  --  DECLARED IN SEGMENT 122 AT 10291000
       *10317000* 10322000 *10330000*
T2  --  REAL  --  DECLARED IN SEGMENT 138 AT 13449000
       *13494000**13500000* 13501000 *13502000* 13547000 *13580000**13584000*
T2  --  REAL  --  DECLARED IN SEGMENT 154 AT 15074000
        15100000  15112000  15124000  15308000  15341000  15369000
T3  --  STREAM VARIABLE  --  DECLARED IN SEGMENT 3 AT 01688020
        01688030  01688060  01688070  01688100
T3  --  REAL  --  DECLARED IN SEGMENT 57 AT 06059000
       *06062150* 06062155
T3  --  REAL  --  DECLARED IN SEGMENT 70 AT 07076000
       *07130000* 07138000 *07194000* 07201000 *07203000**07270000* 07272000 *07290000* 07292000  07306000 *07319000*
       *07364000* 07365000
T3  --  REAL  --  DECLARED IN SEGMENT 84 AT 07658000
       *07665500* 07666500 *07673000*
T3  --  REAL  --  DECLARED IN SEGMENT 89 AT 08173000
       *08199000* 08209000
T3  --  REAL  --  DECLARED IN SEGMENT 122 AT 10291000
       *10327000* 10323000
T3  --  REAL  --  DECLARED IN SEGMENT 138 AT 13449000
       *13483000* 13486000  13494000  13508000  13530000  13531000  13540000 *13575000* 13580000
T4  --  REAL  --  DECLARED IN SEGMENT 70 AT 07076000
       *07106500* 07111230  07111320  07111325 *07292000**07306000*
T4  --  REAL  --  DECLARED IN SEGMENT 89 AT 08173000
       *08200000* 08210000
T4  --  REAL  --  DECLARED IN SEGMENT 138 AT 13449000
       *13517000* 13530000  13531000
T5  --  REAL  --  DECLARED IN SEGMENT 70 AT 07076000
       *07111230**07304000* 07312000
T6  --  REAL  --  DECLARED IN SEGMENT 70 AT 07076000
       *07173000* 07174000 *07197000* 07207000 *07305000* 07312000
ULITOG  --  BOOLEAN  --  DECLARED IN SEGMENT 138 AT 13450000
       *13518000**13523000* 13527000
UNHOOK  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 13353000  --  FORWARD AT 01626000
        13305300
UNKNOWNID  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01177000
        06092000  06227000  07462500
UNKNOWNSTMT  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 07800000  --  FORWARD AT 07[illegible]00
        07432000  07729000
UNTILV  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01226000
        07018000  07485000  07646595  08113000  08128000  08184000
UPDATETIMES  --  PROCEDURE  --  DECLARED IN SEGMENT 167 AT 17002050
        17022200  17091175  17119100
UPPER  --  REAL  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 0[illegible]00
        05297000  05298000  05304000
USEROPINX  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01001173
        02335000  02339000  02353000  02357000  02600000
USESWITCH  --  SWITCH LABEL  --  DECLARED IN SEGMENT 13 AT 022[illegible]
        02212000
USETHESWITCH  --  LABEL  --  DECLARED IN SEGMENT 13 AT 02210250  --  DECLARED[illegible]
        02224102  02229500  02232750  02234650  02235250
V  --  REAL  --  DECLARED IN SEGMENT 15 AT 02238112
       *02238340* 02238350  02238360 *02238390* 02238400 [illegible]
```

```
      *02238658**02238680**02238700* 02238704
V  --  OWN REAL  --  DECLARED IN SEGMENT 89 AT 08010000
       08078000  08099000  08181000  08188000 *08225000*  02270
VAL  --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEG
       01741400
VALTOG  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN
       17047450  17049430
VALUEV  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01258000
       14341000
VARIABLE  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 15030000  --
       06062400  06073500  06107000  06250000  07157000  07173000                    07687500  07738000
       07815000  07836000  07859040  07941000  08228000  08401020                    08728300  10317000
       12037000  12063100  15377000
VBIT  --  BOOLEAN  --  DECLARED IN SEGMENT 70 AT 07062000
      *07081000**07086000* 07108505  07111220  07111303  07117000  07124000                    07334000  07346000  07382000
VE  --  LABEL  --  DECLARED IN SEGMENT 70 AT 07065000  --  OCCURS AT 07125000
       07168000
VO  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01150000
       07086000  07290000  13203000  14116500
VOIDBIT  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01001150
       02231500  02233750  02412000  02415000
VOIDCR  --  REAL  --  DECLARED IN SEGMENT 3 AT 01785000
       02232000 *02233500* 02410000 *02417000* 02485000 *02492000*
VOIDING  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01001450
       02231500  02233750
VOIDPLACE  --  REAL  --  DECLARED IN SEGMENT 3 AT 01785000
      *02233500* 02410000 *02416000* 02417000  02485000 *02491000* 02492000
VOIDTAPE  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01001460
       02231500  02232500  02233750
VOIDTBIT  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01001160
       02231500  02232500  02233750  02487000  02490000
VONE  --  BOOLEAN  --  DECLARED IN SEGMENT 3 AT 01590000
       07341000 *13203000* 13330600 *13341000* 13373000 *13913100*
VP  --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 3 AT 01756000
       01781000  01780000 *01782000* 01783000
VR  --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 3 AT 01756000
       01783000
VRET  --  OWN REAL  --  DECLARED IN SEGMENT 89 AT 08010000
      *08054000* 08077000 *08176000* 08198000 *08208000* 08211000  08224000 *08228000*
W  --  REAL  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 01718200
W  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 01735000
       01736100  01736200
W  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 29 AT 02927240
       02927310
W  --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 53 AT 05309000
       05311000
W  --  REAL  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 10068000
       10071000  10074000
W  --  REAL  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 119 AT 10088000
       10086000  10087000  10090000 *10091000* 10099000
W  --  INTEGER  --  DECLARED IN SEGMENT 119 AT 10106000
      *10129000**10131000**10134530* 10134590  10143000  10146000  10148000 *10150145**10150220* 10150225  10150230
      *10150280**10150290* 10153000  10161500  10162000  10164000 *10165000**10165500**10165505* 10170000  10171000
WAITINGFORFWDREF  --  OWN BOOLEAN  --  DECLARED IN SEGMENT 177 AT 17091110
      *17092100* 17093000 *17093200*
WD  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 104 AT 09005000
       09006000
WHATISIT  --  LABEL  --  DECLARED IN SEGMENT 21 AT 02362000  --  OCCURS AT 02578000
```

```
        02364000  02365000  02379000  02398000  02475000  02517000  02568000  02574300
WHILESTMT  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 07490000
        07496000  07748000
WHILEV  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01224000
        08113000  08133000  08139000
WHIPOUT  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 10068000
        10082000  10099000  10127100  10143000  10148000  13797000
WHOLE  --  REAL  --  DECLARED IN SEGMENT 70 AT 07059000
       *07079000* 07099500  07122510  07122520  07215500  07270000  07271000 *07276000* 07290000  07316000 *07361000*
        07364000
WITHV  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01259000
        06504000  07688000  07812000
WOP  --  REAL ARRAY  --  DECLARED IN SEGMENT 3 AT 01371000
        02534000 *02536000* 04284300  04286000 *04602000* 04603000 *04606000*
WORD  --  REAL  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 04142000
        04147000  04149000
WORD  --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 3 AT 04270000
        04273000
WORD  --  REAL  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 05008000
        05009000
WORDCOUNT  --  REAL  --  DECLARED IN SEGMENT 135 AT 13301000
       *13302000* 13307000  13310000  13312000
WRITELINE  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 02181000
        02185000  02195750  02196590  02214200  02227250  02227500  02228750  04150000  05039600  05046000  05265000
        05326490  07025020  09302000  10071000
WRITEPRT  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 05325010
        05326500  05376100
WRITERROR  --  STREAM PROCEDURE  --  DECLARED IN SEGMENT 45 AT 05016000
        05031000  05044000
WRITESTMT  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 08578000  --  FORWARD AT 03062000
        07741000  08700000
WRITEV  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01217000
WRITNEW  --  BOOLEAN STREAM PROCEDURE  --  DECLARED IN SEGMENT 3 AT 02016000
       *02018000* 02019000  02194000
WRITXFORM  --  LABEL  --  DECLARED IN SEGMENT 97 AT 08591200  --  OCCURS AT 08647000
        08658020
WRTINTRST  --  STREAM PROCEDURE  --  DECLARED IN SEGMENT 104 AT 09010000
        09025000  09300200
W1  --  REAL  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 119 AT 10088000
        10086000  10087000  10093000
W1  --  INTEGER  --  DECLARED IN SEGMENT 119 AT 10106000
        10153000  10161500 *10164000**10165505**10167000**10169000**10[illegible]
W2  --  REAL  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 119 AT 10088[illegible]
        10086000  10087000  10094000
W2  --  INTEGER  --  DECLARED IN SEGMENT 119 AT 10106000
        10153000  10161500  10171000
X  --  INTEGER  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 01[illegible]
X  --  REAL  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 01[illegible]
X  --  REAL  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 13[illegible]
        13589000  13590000  13593000
X  --  REAL  --  DECLARED IN SEGMENT 154 AT 15075500
       *15273000* 15273100
XBIT  --  INTEGER  --  VALUE PARAMETER  --  DECLARED IN SE[illegible]
        02327000  02335000  02339000  02353000  02357000[illegible]
XCH  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01675000[illegible]
        04071000  04507000  06062170  06069000  0607[illegible]
        07470750  07634000  07851000  07859070  079[illegible]
```

```
        08728600  08974000  08976000  10851000  10[illegible]                                    [illegible]
        15230000  15309000
XINFO* --  REAL ARRAY  --  DECLARED IN SEGMENT 3 AT 01[illegible]
        02001660  02001695  02001795  02001815 *02001[illegible]
XIT  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01676000
        13747000  14557500
XIT  --  LABEL  --  DECLARED IN SEGMENT 13 AT 02210500  --
        02213250  02214750  02222250  02224000
XIT  --  STREAM LABEL  --  DECLARED IN SEGMENT 54 AT 05325080
        05325340  05325350  05325370  05325380  05325390  05325400
XIT  --  LABEL  --  DECLARED IN SEGMENT 81 AT 07646410  --  OCCURS AT
        07646440
XITR  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01430000
XLUN  --  INTEGER  --  DECLARED IN SEGMENT 3 AT 01007380
       *13310450* 17001000  17004500 *17004600* 17022300 *17025100**17025300*                    *
XMARK  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01007455
        07598100  07687600  07835500  08178100  13310525  14310500  15091100  [illegible]
XMODE  --  INTEGER  --  DECLARED IN SEGMENT 3 AT 01001530
        02328000 *02332000**02367000**02383000**02385000* 02408000 *02412000* 02419100  02425500 *02478000* 02483000
       *02487000* 02525003 *02525010* 02525012
XMODE0  --  LABEL  --  DECLARED IN SEGMENT 22 AT 02325000  --  OCCURS AT 02329000
        02326000
XMODE1  --  LABEL  --  DECLARED IN SEGMENT 22 AT 02325000  --  OCCURS AT 02333000
        02326000
XMODE2  --  LABEL  --  DECLARED IN SEGMENT 22 AT 02325000  --  OCCURS AT 02337000
        02326000
XMODE3  --  LABEL  --  DECLARED IN SEGMENT 22 AT 02325000  --  OCCURS AT 02341000
        02326000
XMODE4  --  LABEL  --  DECLARED IN SEGMENT 22 AT 02325000  --  OCCURS AT 02355000
        02326000
XREF  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01001461
        02882200  07598100  07687600  07835500  08178100  13310050  13310525  13310580  13348100  14310500  14469100
        15091100  15116000  15301100  15342300  16159100  16318500  17001000
XREFAY1  --  REAL ARRAY  --  DECLARED IN SEGMENT 3 AT 01007145
        02001805 *02001815**02001820* 02001821 *17084000* 17091900  17093850  17093950  17094050  17094100  17094700
        17095000  17095300  17095320
XREFAY2  --  REAL ARRAY  --  DECLARED IN SEGMENT 3 AT 01007030
        02001680 *02001695**07598100**07687600**07835500**08178100**13310525**14310500**15091100**16159100**16318500*
       *17003000* 17004000  17076000  17079000
XREFBIT  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01001170
        02419100  02882200  07598100  07687600  07835500  08178100  13310050  13310525  13310580  13348100  14310500
        14469100  15091100  15116000  15301100  15342300  16159100  16318500  17001000
XREFDUMP  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01007470
        13283500  14445100
XREFINFO  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01007481
        02001660  02001695  02001795  02001815  02001822  13310200  13310350  13310750  13310950
XREFINFO  --  DEFINE  --  DECLARED IN SEGMENT 167 AT 17002005
        17004710  17025100  17025300  17080000
XREFIT  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01007440
        02882200  13310575  14469100  15116000  15301100  15342300
XREFPT  --  INTEGER  --  DECLARED IN SEGMENT 3 AT 01007355
        02001670 *02001685* 02001695 *02001705* 07598100  07687600  07835500  08178100  13310525 *13348200* 14310500
        15091100  16159100  16318500 *17003000**17004600**17074000**17077000* 17079000 *17084000* 17091400  17094200
       *17094600* 17095900 *17096100**17096400**17118000*
XTOTHFI  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01571000
        04078000
XXX  --  LABEL  --  DECLARED IN SEGMENT 86 AT 07801000  --  OCCURS AT 07914000
```

```
        07809000  07810100  07818000  07825500  07826000  07827000  07830000  07839000  07841000  07845000  07850000
        07853000  07854000  07858000  07859030  07859090  07900000  07902000  07906000  07910000
Z   --  REAL  --  DECLARED IN SEGMENT 3 AT 01490000
        *07646660**13772000**13784000**14391000* 14457000
Z   --  REAL  --  DECLARED IN SEGMENT 154 AT 15075500
        *15273100* 15273200  15273300
ZEROUT  --  STREAM PROCEDURE  --  DECLARED IN SEGMENT 3 AT 01737000
        09395500
ZONEP   --  STREAM VARIABLE  --  DECLARED IN SEGMENT 3 AT 02013073
        02013074  02013135  02013136
ZP1     --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01677000


CROSS REFERENCE STATISTICS
----- --------- ----------

PHASE ONE - SORT  2248 IDENTIFIERS
        1:1  ELAPSED TIME (MIN:SEC)
        0:1  PROCESSOR TIME
        1:10  I/O TIME

PHASE TWO - SORT  19408 REFERENCES
        3:0  ELAPSED TIME (MIN:SEC)
        2:5  PROCESSOR TIME
        0:2  I/O TIME

PHASE THREE - PRINT CROSS REFERENCE (  LINES)
        2:59  ELAPSED TIME (MIN:SEC)
        1:24  PROCESSOR TIME
        0:54  I/O TIME


LABEL  00000000LINE   00177280? COMPILE ALGOL  DISK ALGOL LIBRARY                    ALGOL  /ALGOL
```