```
? EXECUTE PATCH/MERGE

            PACKET      1
            INPUT      25 CARDS FROM CRA
            TIME     1553
            DATE    77234 MONDAY, 08/22/77



        *** BURROUGHS B5700 DCMCP MARK XVI.0.73  AND INTRINSICS MARK XVI.0.00 ***

            #NO MESSAGES TODAY

         ? EXECUTE PATCH/MERGE
             ? FILE CARD= PMCARD
         ? DATA PMCARD
                  4:PATCH/MERGE= 1 BOJ 1553 05/12/77
                  CDA IN PMCARD:PATCH/MERGE= 1
                  PBD0002 OUT 011 LINE:PATCH/MERGE= 1
                  DKA OUT SER MASTERF U000000:PATCH/MERGE= 1
                  DKA IN SER PATCH ESPOL:PATCH/MERGE= 1
                  DKA OUT SER PATCHDE U000000:PATCH/MERGE= 1
                  CDA REL PMCARD:PATCH/MERGE= 1
         ? END
         DKA REL PATCHDE U000000:PATCH/MERGE= 1
         DKA REL PATCH ESPOL:PATCH/MERGE= 1
         PBD0002 REL 011 LINE 1808:PATCH/MERGE= 1
         DKA REL MASTERF U000000:PATCH/MERGE= 1
         PATCH/MERGE= 1 EOJ 1555
         FOR PATCH/MERGE= 1: PROCESS= 65 SECS, IO= 43 SECS, OLAY= 0
         PKT#0001 REMOVED
```

BURROUGHS B-5700 PATCH/MERGE PROGRAM MARK XVI.0.104   MONDAY, 08/22/77,   3:53 PM.

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

```
$@CARD CONFLICTS LIST; MERGE ZIP                                                C       0000000/PMCARD
$, 18 PATCHES FOR ESPOL                                                         C

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

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

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

       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 ESPOL 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 ESPOL 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 ESPOL 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 ESPOL CONTAINS 11 CARDS                                 P  105

LABEL ENDREADTAPE, EOFT;                                                02201510  P  105
READ (TAPE, 10, TBUFF[*])[EOFT];                                        02201750  P  105
                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
$;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 ESPOL CONTAINS 2 CARDS.                          00000000  P  106

                IF ERRNUM = 200 THEN I := I/0; % SEGMENT TOO LARGE.     05107100  P  106
                IF ERRNUM = 611 THEN I := I/0; % ERRMAX EXCEEDED.       05107200  P  106
$; DATE 6/27/75                                                         99990000  P  106
$; BY MWG - MSA CENTRAL.                                                99990100  P  106
$; THIS PATCH CAUSES ESPOL TO ABORT WITH A DIVIDE BY ZERO ERROR FOR      99990200  P  106
$; THE ERRORS "SEGMENT TOO LARGE" AND "ERROR LIMIT EXCEEDED".           99990300  P  106



$#PATCH NUMBER 107 FOR ESPOL CONTAINS 12 CARDS.                         00000000  P  107
```

```
                LABEL EXIT,NEXTCHK;                                                    06315000  P  107
                IF STEPI = RTBRKET THEN % WE WILL TAKE CARE OF THE RES                 06320000  P  107
                  IF (FIRST != ELBAT[I-3].ADDRESS) ×                                   06320100  P  107
                     (THIRD != ELBAT[I-1].ADDRESS) ≠ 0 THEN                            06320200  P  107
                    BEGIN                                                              06320300  P  107
                      SECOND := 48 - THIRD; % SOURCE IS RIGHT JUST.                    06320400  P  107
                      GO TO NEXTCHK;                                                   06320500  P  107
                    END                                                                06320600  P  107
                  ELSE % BAD BITS, FALL THROUGH TO ERROR.                              06320700  P  107
                ELSE % MAYBE A COLON.                                                  06320800  P  107
                  IF FLCLASS = COLON THEN %                                            06320900  P  107
  NEXTCHK:      IF FIRST + THIRD ≤ 48 THEN % SO FAR SO GOOD.                         06327000  P  107
$! DATE 1/22/76                                                                        99990000  P  107
$! BY JTC - MSA CENTRAL                                                               99990100  P  107
$! THIS PATCH ALTERS THE SYNTAX FOR THE CONCATENATE OPERATOR TO ALLOW                 99990200  P  107
$! THE SECOND VALUE IN THE CONCAT. LIST TO BE OMITTED (THE                           99990300  P  107
$! STARTING BIT NUMBER IN THE SOURCE WORD) IF THE RIGHT-MOST                         99990400  P  107
$! N BITS ARE BEING SELECTED FROM THE SOURCE WORD. THUS                              99990500  P  107
$!                                                                                    99990600  P  107
$!                I := I & J [2:20]                                                   99990700  P  107
$!                                                                                    99990800  P  107
$! IS THE SAME AS                                                                    99990900  P  107
$!                                                                                    99991000  P  107
$!                I := I & J [2:28:20]                                                99991100  P  107
$!                                                                                    99991200  P  107
$! THIS IS ESPECIALLY CONVENIENT FOR ESPOL SINCE THIS ALLOWS SOME                     99991300  P  107
$! FIELD DEFINES TO BE USED IN CONCATENATE EXPRESSIONS. THUS IF WE                    99991400  P  107
$! HAVE THE FOLLOWING FIELD DEFINITION                                               99991500  P  107
$!                                                                                    99991600  P  107
$!                PBIT = [2:1]                                                        99991700  P  107
$!                                                                                    99991800  P  107
$!                                                                                    99991900  P  107
$! WE CAN USE THE SAME FIELD DEFINITION IN A CONCATENATE STATEMENT                    99992000  P  107
$! AS FOLLOWS                                                                        99992100  P  107
$!                                                                                    99992200  P  107
$!                DESC := DESC & 1 PBIT                                               99992300  P  107
$!                                                                                    99992400  P  107
$!                                                                                    99992500  P  107
$! NOTE: FUTURE PATCHES TO THE MCP AND INTRINSICS WILL TAKE ADVANTAGE                 99992600  P  107
$! OF THIS PATCH.                                                                    99992700  P  107



$#PATCH NUMBER 108 FOR ESPOL CONTAINS 252 CARDS                                        00000000  P  108

        XREFBIT         = 26#,                                                         01001170  P  108
        BENDBIT         = 27#,                                                         01001171  P  108
        USEROPINX       = 28#;                                                         01001172  P  108
        XREF            = OPTIONWORD.[XREFBIT:1] #,                                    01001461  P  108
        BEND            = OPTIONWORD.[BENDBIT:1]#,                                     01001462  P  108
DEFINE XREFINFO[XREFINF01,XREFINF02]=                                                  01001750  P  108
        XINFO[XREFINF01,(XREFINF02)DIV 2]#,                                            01001760  P  108
      XMARK= IF XREF THEN XREFAY2[XREFPT-1].[1:1]=1#;                                  01001770  P  108
    ARRAY XREFAY2[0:29],XREFAY1[0:10],XINFO[0:31,0:127];                               01007100  P  108
INTEGER XREFPT,XLUN;                                                                  %DFB01007200  P  108
```

```
 ARRAY BEGINSTACK[0:255]; INTEGER BSPOINT;                                  01007600  P  108
 BOOLEAN DEFINING;                                                          01007650  P  108
     FILE DSK1 DISK SERIAL [20:816](2,10,30);                               01561085  P  108
     FILE DSK2 DISK SERIAL [20:450](2,30,30);                               01561087  P  108
 PROCEDURE BEGINPRINT;                                                      02196510  P  108
BEGIN                                                                       02196520  P  108
      STREAM PROCEDURE STUFF(N,L); VALUE N;                                 02196530  P  108
     BEGIN                                                                  02196540  P  108
          DI:=L; DS:=8 LIT " "; SI:=L; DS:=13 WDS;                          02196550  P  108
          SI:=LOC N; DS:=8 DEC;                                             02196560  P  108
     END;                                                                   02196570  P  108
      STUFF(BEGINSTACK[BSPOINT],LIN);                                       02196580  P  108
      IF NOHEADING THEN DATIME; WRITELINE;                                  02196590  P  108
END BEGINPRINT;                                                             02196610  P  108
 CARDNUMBER:=CONV(INFO[LASTSEQROW,LASTSEQUENCE-1],5,8);                     02214260  P  108
$                                                                           02228000  P  108
      CARDNUMBER:=IF SEQTOG THEN TOTALNO+ADDVALUE ELSE                      02234800  P  108
            CONV(INFO[LASTSEQROW,LASTSEQUENCE-1],5,8);                      02234900  P  108
 IF Q = "4XREFO" THEN BEGIN SWITCHIT(XREFBIT); IF BOOLEAN(XMODE) THEN       02419100  P  108
      DEFINING := BOOLEAN(REAL(DEFINING)&1[1:47:1]);                        02419110  P  108
                  GO AGAIN  END;                                            02419120  P  108
 IF Q = "4BENDO" THEN BEGIN SWITCHIT(BENDBIT); GO AGAIN END;                02419200  P  108
      IF XREF THEN                                                          02882100  P  108
      IF GT1#1 THEN                                                         02882200  P  108
      BEGIN                                                                 02882300  P  108
           IF XREFPT=30 THEN                                                02882400  P  108
           BEGIN                                                            02882500  P  108
                WRITE (DSK2,30,XREFAY2[*]);                                 02882600  P  108
                XREFPT ← 0;                                                 02882700  P  108
           END;                                                             02882800  P  108
           XREFAY2[XREFPT]←CARDNUMBER&XREFINFO[GT1,GT2][09:36:12];          02882900  P  108
           XREFPT ← XREFPT+1;                                               02882910  P  108
      END;                                                                  02882920  P  108
  IF NOT DEFINING THEN                                                      02910100  P  108
  IF T.CLASS = BEGINV THEN                                                  02910200  P  108
      BEGINSTACK[BSPOINT:=BSPOINT+1]:=CARDNUMBER ELSE                       02910300  P  108
  IF T.CLASS = ENDV THEN                                                    02910400  P  108
BEGIN                                                                       02910500  P  108
      IF LISTER THEN IF BEND THEN BEGINPRINT;                               02910600  P  108
      BSPOINT:=BSPOINT - REAL(BSPOINT > 0); % PREVENT INVALID INDEX         02910700  P  108
END;                                                                        02910800  P  108
                    "4BENDO",0,          % 50, 51                         09251285  P  108
                    "4XREFO",0,          % 52, 53                         09251286  P  108
STREAM PROCEDURE XREFMOVE(S,C,SN,SQ,L,D);                                %DFB13301100  P  108
  VALUE C,L;                                                             %DFB13301130  P  108
  BEGIN                                                                  %DFB13301160  P  108
      DI:=D;                                                             %DFB13301190  P  108
      DS:= 8 LIT" ";                                                     %DFB13301220  P  108
      SI:=D;                                                             %DFB13301250  P  108
      DS:=8 WDS;                                                         %DFB13301280  P  108
      DI:=D;                                                             %DFB13301310  P  108
      SI:=S;                                                             %DFB13301340  P  108
      SI:=SI+3;                                                          %DFB13301370  P  108
      DS:=C CHR;                                                         %DFB13301400  P  108
      DI:=D;                                                             %DFB13301430  P  108
       DI←DI+60;                                                         %DFB13301460  P  108
      SI:=SN;                                                            %DFB13301490  P  108
      DS:=4 DEC;                                                         %DFB13301520  P  108
```

```
        S1:=SQ;                                                          %DFB13301610  P  108
        DS:=8 DFC;                                                       %DFB13301640  P  108
        S1:=LOC L;                                                       %DFB13301670  P  108
        DS:=WDS;                                                         %DFB13301700  P  108
    END;                                                                 %DFB13301730  P  108
        IF XREF AND (NOT PTOG OR STREAMTOG) THEN                            13310100  P  108
    BEGIN                                                                %DFB13310200  P  108
    XREFMOVE(ACCUM[1],COUNT,SGNO ,CARDNUMBER,                            %DFB13310300  P  108
             XREFINFO[NEXTINFO.LINKR,NEXTINFO.LINKC]:=XLUN:=XLUN+1,      %DFB13310400  P  108
               XREFAY1);                                                    13310450  P  108
    WRITE(DSK1,10,XREFAY1[*]);                                           %DFB13310500  P  108
    END;                                                                 %DFB13310600  P  108
            IF XREF AND PTOG THEN                                           13341200  P  108
              XREFPT←XREFPT-REAL(ELBAT[I]≠0); % GET RID OF LAST CREF         13341300  P  108
$VOIDT 13366105                                                             13366001  P  108
        DEFINING := BOOLEAN(REAL(DEFINING) & 1[47:47:1]);                   14255500  P  108
        DEFINING := BOOLEAN(REAL(DEFINING) & 0[47:47:1]);                   14268500  P  108
L1:                                                                         15092000  P  108
       XMARK;                                                               15092010  P  108
       IF  TALL.FORMALNAME THEN                                             15092020  P  108
     ENDOFITALL;                                                            16495210  P  108
IF XREF OR  DEFINING.[1:1] THEN                                             16495300  P  108
     BEGIN DEFINE LSS= <#,GTR=>#,NEQ= ≠#,LEQ=≤#;                         %DFB17002000  P  108
               DEFINE XREFINFO=INFO#;                                       17002005  P  108
     CLOSE(CARD,RELEASE);                                                   17002490  P  108
     CLOSE(TAPE,RELEASE);                                                   17002500  P  108
                 LOCK(NEWTAPE);                                             17002510  P  108
     WRITE(LINE[PAGE]);                                                     17002520  P  108
    FOR XREFPT:=XREFPT STEP 1 UNTIL 29 DO XREFAY2[XREFPT]:=100000000;       17003000  P  108
    WRITE(DSK2,30,XREFAY2[*]);                                           %DFB17004000  P  108
          TOTALNO←REAL(XLUN >500)×1000+3000+XLUN;                           17004500  P  108
 XREFPT←XLUN←0;                                                         %DFB17004600  P  108
  BEGIN                                                                 %DFB17005000  P  108
  BOOLEAN PROCEDURE INPUT1(A);                                          %DFB17006000  P  108
     ARRAY A[0];                                                        %DFB17007000  P  108
     BEGIN                                                              %DFB17008000  P  108
     LABEL L,EOF;                                                       %DFB17009000  P  108
     READ(DSK1,10,A[*])[EOF];                                           %DFB17010000  P  108
     GO TO L;                                                           %DFB17011000  P  108
     EOF: INPUT1:=TRUE;                                                 %DFB17012000  P  108
          REWIND(DSK1);                                                 %DFB17013000  P  108
     L:                                                                 %DFB17014000  P  108
     END;                                                               %DFB17015000  P  108
PROCEDURE OUTPUT1(B,A);                                                 %DFB17016000  P  108
  VALUE B;                                                              %DFB17017000  P  108
  BOOLEAN B;                                                            %DFB17018000  P  108
  ARRAY A[0];                                                           %DFB17019000  P  108
     BEGIN                                                              %DFB17020000  P  108
     IF B THEN                                                          %DFB17021000  P  108
       REWIND(DSK1)                                                     %DFB17022000  P  108
     ELSE                                                               %DFB17023000  P  108
       BEGIN                                                            %DFB17024000  P  108
       A[9]:=XREFINFO[A[9],LINKR,A[9],LINKC]:=XLUN:=XLUN+1;             %DFB17025000  P  108
       WRITE(DSK1,10,A[*]);                                             %DFB17026000  P  108
       END;                                                             %DFB17027000  P  108
     END;                                                               %DFB17028000  P  108
BOOLEAN STREAM PROCEDURE COMPS1(A,B);                                   %DFB17029000  P  108
```

```
        BEGIN                                                          %DFB17030000  P  108
        SI:=A;                                                         %DFB17031000  P  108
        DI:=B;                                                         %DFB17032000  P  108
        IF 63 SC LSS DC THEN TALLY:=1;                                 %DFB17033000  P  108
        COMPS1:=TALLY;                                                 %DFB17034000  P  108
        END;                                                           %DFB17035000  P  108
      STREAM PROCEDURE HVS1(A);                                        %DFB17036000  P  108
        BEGIN                                                          %DFB17037000  P  108
        DI:=A;                                                         %DFB17038000  P  108
        DS:=8 LIT "9";                                                 %DFB17039000  P  108
        SI:=A;                                                         %DFB17040000  P  108
        DS:= 7 WDS;                                                    %DFB17041000  P  108
        END;                                                           %DFB17042000  P  108
      BOOLEAN PROCEDURE COMP1(A,B);                                    %DFB17042100  P  108
              ARRAY A,B[0];                                            %DFB17042200  P  108
              COMP1:=COMPS1(A,B);                                      %DFB17042300  P  108
      PROCEDURE HV1(A);                                                %DFB17042400  P  108
              ARRAY A[0];                                              %DFB17042500  P  108
              HVS1(A);                                                 %DFB17042600  P  108
      XLUN:=0;                                                         %DFB17043000  P  108
      REWIND(DSK1);                                                    %DFB17044000  P  108
      SORT(OUTPUT1,INPUT1,0,HV1,COMP1,10,TOTALNO    );                 %DFB17045000  P  108
END;                                                                   %DFB17046000  P  108
BEGIN                                                                  %DFB17047000  P  108
      STREAM PROCEDURE PUP(S,D);                                       %DFB17048000  P  108
        BEGIN                                                          %DFB17049000  P  108
        SI:=S;                                                         %DFB17050000  P  108
        DI:=D;                                                         %DFB17051000  P  108
      DS:=7 WDS; 8(DS:=8 LIT " ");                                         17052000  P  108
$VOIDT 17054551                                                            17053000  P  108
        END;                                                           %DFB17055000  P  108
      STREAM PROCEDURE PUP1(S,D);                                          17055100  P  108
      BEGIN                                                                17055200  P  108
      SI:=S; SI:=SI+32; SI:=SI+28;                                         17055300  P  108
      DI:=D; DS:=2 LIT " "; DS:=4 CHR; DS:=2 LIT " *";                     17055400  P  108
      DS:=8 CHR; DS:=LIT "*"; DS:=38 LIT " ";                              17055500  P  108
      END;                                                                 17055550  P  108
       STREAM PROCEDURE PUP2(STAR,S,C,D);                                  17056000  P  108
       VALUE STAR,S,C;                                                     17056500  P  108
        BEGIN                                                          %DFB17057000  P  108
        DI:=D;                                                         %DFB17058000  P  108
      C(DI:=DI+10);                                                        17059000  P  108
                                  STAR(DI+DI-1; DS+LIT"*");                17059100  P  108
        SI:=LOC S;                                                     %DFB17060000  P  108
        DS:= 8 DEC;                                                    %DFB17061000  P  108
      DS:=LIT " ";          STAR(DI:=DI-1; DS:=LIT "*");                   17061100  P  108
        END;                                                           %DFB17062000  P  108
      STREAM PROCEDURE BLANKET(A);                                     %DFB17063000  P  108
          BEGIN                                                        %DFB17064000  P  108
         DI:=A;                                                        %DFB17065000  P  108
         DS:= 8 LIT " ";                                               %DFB17066000  P  108
      SI:=A;                                                           %DFB17067000  P  108
      DS:= 14 WDS;                                                     %DFB17068000  P  108
      END;                                                             %DFB17069000  P  108
      ARRAY PAY[0:17];                                                 %DFB17069500  P  108
      BOOLEAN PROCEDURE INPUT2(A);                                     %DFB17070000  P  108
      ARRAY A[0];                                                      %DFB17071000  P  108
```

```
  BEGIN                                                                                %DFB17072000  P  108
  LABEL L,EOF;                                                                         %DFB17073000  P  108
   DEFINE I=XLUN#;                                                                     %DFB17073100  P  108
  IF XREFPT:=XREFPT+1=30 THEN                                                          %DFB17074000  P  108
    BEGIN                                                                              %DFB17075000  P  108
    READ(DSK2,30,XREFAY2[*])[EOF];                                                     %DFB17076000  P  108
    XREFPT:=0;                                                                         %DFB17077000  P  108
    END;                                                                               %DFB17078000  P  108
IF ( I  :=XREFAY2[XREFPT]).[21:27] GTR 99999999 THEN GO TO EOF;                        17079000  P  108
A[0]:= I&XREFINFO[I.[ 9:4],I,[13:8]][ 9:36:12];                                        %DFB17080000  P  108
GO TO L;                                                                               %DFB17081000  P  108
EOF: INPUT2:=TRUE;                                                                     %DFB17082000  P  108
     BLANKET(PAY);                                                                     %DFB17083000  P  108
XREFAY1[9]:=-1;       XREFPT←0;                                                        %DFB17084000  P  108
L:                                                                                     %DFB17085000  P  108
END;                                                                                   %DFB17086000  P  108
PROCEDURE OUTPUT2(B,A);                                                                %DFB17087000  P  108
  VALUE B;                                                                             %DFB17088000  P  108
  BOOLEAN B;                                                                           %DFB17089000  P  108
  ARRAY A[0];                                                                          %DFB17090000  P  108
  BEGIN      DEFINE PRINTER=LINE#;                                                     %DFB17091000  P  108
     LABEL LOOP,EOF2;                                                                      17091100  P  108
OWN BOOLEAN B2;                                                                          17091110  P  108
 LABEL DUN;                                                                              17091200  P  108
IF NOT B2 THEN                                                                           17091210  P  108
  IF B THEN                                                                            %DFB17092000  P  108
     WRITE(PRINTER,15,PAY[*])                                                          %DFB17093000  P  108
       ELSE                                                                            %DFB17094000  P  108
IF LASTADDRESS ≠ LASTADDRESS←A[0] AND LASTADDRESS.[9 :12]≠ 0 THEN                        17094100  P  108
          BEGIN                                                                        %DFB17095000  P  108
 LOOP:                                                                                 %DFB17095050  P  108
          IF  A[0].[ 9:12] GTR XREFAY1[9] THEN                                         %DFB17095100  P  108
          BEGIN                                                                        %DFB17095200  P  108
$                                                                                          17096000  P  108
      WRITE(PRINTER,15,PAY[*]);                                                            17097000  P  108
$                                                                                          17098000  P  108
          BLANKET(PAY ); XREFPT:=0;                                                        17099000  P  108
          READ(DSK1,10,XREFAY1[*])[EOF2];                                                  17099100  P  108
          PUP(XREFAY1,PAY);                                                                17099200  P  108
          WRITE(PRINTER[DBL]);                                                             17099900  P  108
          WRITE(PRINTER,10,PAY[*]);                                                    %DFB17100000  P  108
        PUP1(XREFAY1,PAY);                                                                 17100100  P  108
          WRITE(PRINTER[DBL],10,PAY[*]);                                                   17100200  P  108
          BLANKET(PAY);                                                                %DFB17101000  P  108
      GO LOOP;                                                                         %DFB17101100  P  108
          END;                                                                         %DFB17102000  P  108
DUN:                                                                                       17102100  P  108
                          PUP2(LASTADDRESS < 0,LASTADDRESS.[21:27],                    17103000  P  108
                               XREFPT,PAY[1]);                                             17103010  P  108
          IF XREFPT:=XREFPT+1 = 11 THEN                                                    17103100  P  108
            BEGIN                                                                      %DFB17103200  P  108
              XREFPT:=0;                                                               %DFB17103300  P  108
              WRITE(PRINTER,15,PAY[*]);                                                %DFB17103400  P  108
              BLANKET(PAY);                                                            %DFB17103500  P  108
            END;                                                                       %DFB17103600  P  108
          END;                                                                         %DFB17110000  P  108
        IF FALSE THEN EOF2: B2:=TRUE;                                                      17110500  P  108
```

```
          END;                                                          %DFB17111000  P  108
        PROCEDURE HV2(A);                                               %DFB17112000  P  108
          ARRAY A[0];                                                   %DFB17113000  P  108
          A[0]:=549755813887;                                           %DFB17114000  P  108
        BOOLEAN PROCEDURE COMP2(A,B);                                   %DFB17115000  P  108
          ARRAY A,B[0];                                                 %DFB17116000  P  108
         COMP2:= ABS(A[0]) LEQ ABS(B[0]);                                   17117000  P  108
        XREFPT:=29;       REWIND(DSK2);                                 %DFB17118000  P  108
        SORT(OUTPUT2,INPUT2,0,HV2,COMP2,1,TOTALNO    );               %DFB17119000  P  108
    END;                                                                %DFB17120000  P  108
    END;                                                                %DFB17121000  P  108
                  END MAIN BLOCK;                                           17121500  P  108
    END.                                                                %DFB17122000  P  108
$! BY JTC - MSC DETROIT                                                    99990000  P  108
$!**************************************************************************99990100  P  108



$#PATCH NUMBER 110 FOR FSPOL CONTAINS 503 CARDS                             00000000  P  110

$ VOID 01001771                                                             01001750  P  110
%***************************************************************************01007005  P  110
%              X R E F   S T U F F                                          01007010  P  110
%***************************************************************************01007015  P  110
%                                                                           01007020  P  110
ARRAY                                                                       01007025  P  110
  XREFAY2[0:29],        % ARRAY OF ONE WORD REFERENCE RECORDS.             01007030  P  110
                        % THE LAYOUT OF EACH WORD IS                      01007035  P  110
                        %                                                 01007040  P  110
                        %         .[1:5]  TYPE OF REFERENCE               01007045  P  110
                        %                   = 0 FOR FORWARD DECL          01007050  P  110
                        %                   = 1 FOR LABEL OCCURENCE       01007051  P  110
                        %                   = 2 FOR NORMAL DECL           01007055  P  110
                        %                   = 4 FOR NORMAL REFERENCE      01007060  P  110
                        %                   = 5 FOR ASSIGNMENT            01007065  P  110
                        %                                                 01007070  P  110
                        %                    NOTE: THE LOWER ORDER BIT    01007075  P  110
                        %                          OF THIS FIELD IS ON    01007080  P  110
                        %                          IF YOU WANT STARS      01007085  P  110
                        %                          AROUND THIS REFERENCE  01007090  P  110
                        %                          IN THE XREF            01007095  P  110
                        %                                                 01007100  P  110
                        %         .[6:15] IDENTIFIER ID. NO.              01007105  P  110
                        %                   THIS IS A UNIQUE NUMBER THAT  01007110  P  110
                        %                   IS ASSIGNED WHEN THE          01007115  P  110
                        %                   IDENTIFIER IS ENCOUNTERE      01007120  P  110
                        %                   FOR THE FIRST TIME.           01007125  P  110
                        %                                                 01007130  P  110
                        %         .[21:27] SEQUENCE NUMBER                01007135  P  110
                        %                                                 01007140  P  110
    XREFAY1[0:9],       % RECORD BUFFER AREA FOR WRITING OUT THE          01007145  P  110
                        % NAME INFORMATION RECORDS. ONE RECORD            01007150  P  110
                        % IS WRITTEN FOR EACH IDENTIFIER IN THE SYMBOL    01007155  P  110
                        % TABLE WHEN THE IDENTIFIER IS PURGED FROM THE    01007160  P  110
                        % SYMBOL TABLE, I.E., WHEN LEAVING THE BLOCK      01007165  P  110
                        % IN WHICH THE IDENTIFIER IS DECLARED.            01007170  P  110
                        %                                                 01007175  P  110
```

```
                % THE LAYOUT OF EACH RECORD IS:                          01007180  P  110
                %                                                        01007185  P  110
                %       WORDS 0-7    THE IDENTIFIER WITH BLANK           01007190  P  110
                %                      FILL ON THE RIGHT                 01007195  P  110
                %                                                        01007200  P  110
                %       WORD 8                                           01007205  P  110
                %         .[21:12]   SEGMENT NUMBER IN WHICH            01007210  P  110
                %                      THIS IDENTIFIER WAS DECLARED      01007215  P  110
                %                                                        01007220  P  110
                %         .[33:15]   IDENTIFIER ID. NO.                  01007225  P  110
                %                                                        01007230  P  110
                %       WORD 9       ELBAT WORD                          01007235  P  110
                %                                                        01007240  P  110
XINFO[0:31,0:127]; % THIS ARRAY CONTAINS ONE ENTRY FOR EACH ENTRY        01007245  P  110
                % IN THE INFO TABLE. IF YOU HAVE THE INDEX               01007250  P  110
                % OF THE ELBAT WORD FOR AN IDENTIFIER IN                 01007255  P  110
                % THE INFO TABLE YOU CAN FIND THE XINFO WORD             01007260  P  110
                % FOR THE IDENTIFIER  BY REFERRING TO:                   01007265  P  110
                %                                                        01007270  P  110
                %       XINFO[INDEX.LINKR,INDEX.LINKC DIV 2]             01007275  P  110
                %                                                        01007280  P  110
                % EACH ENTRY CONTAINS:                                   01007285  P  110
                %                                                        01007290  P  110
                %             .[21:12]  SEGMENT NUMBER IN WHICH          01007295  P  110
                %                         THIS IDENTIFIER WAS DECL       01007300  P  110
                %                                                        01007305  P  110
                %             .[33:15]  IDENTIFIER ID. NO.               01007310  P  110
                %                         IF THIS ID. NO. IS ZERO        01007315  P  110
                %                         THEN XREF WAS NOT ON           01007320  P  110
                %                         AT THE TIME THE IDENT          01007325  P  110
                %                         WAS DECLARED AND ALL           01007330  P  110
                %                         FUTURE REFERENCES WILL         01007335  P  110
                %                         BE DISCARDED.                  01007340  P  110
                %                                                        01007345  P  110
INTEGER         %                                                        01007350  P  110
  XREFPT,       % CONTAINS INDEX OF NEXT AVAILABLE SLOT IN              01007355  P  110
                % XREFAY2. WHEN THIS BECOMES GREATER                    01007360  P  110
                % THAN 30 THE CURRENT ARRAY IS DUMPED TO DISK           01007365  P  110
                % AND XREFPT IS RESET TO ZERO.                          01007370  P  110
                %                                                        01007375  P  110
  XLUN;         % THIS VARIABLE CONTROLS THE ASSIGNING OF                01007380  P  110
                % ID. NO. TO IDENTIFIERS. IT IS INCREMENTED              01007385  P  110
                % EACH TIME A NEW IDENTIFIER IS ENCOUNTERED.             01007390  P  110
                %                                                        01007395  P  110
DEFINE          %                                                        01007400  P  110
  SEGNOF = [21:12]#,    % FIELDS IN XINFO ENTRIES AND WORD 8 OF          01007405  P  110
  IDNOF  = [33:15]#,    % IDENTIFIER RECORDS.                            01007410  P  110
                %                                                        01007415  P  110
  TYPEREF = [1:5]#,     % FIELDS OF REFERENCE WORDS                      01007420  P  110
  REFIDNOF =[6:15]#,    %                                               01007425  P  110
  SEQNOF = [21:27]#,    %                                               01007430  P  110
                %                                                        01007435  P  110
  XREFIT(INDEX,SEQNO,REFTYPE) = % DEFINE TO ADD INFO TO REF TABLE        01007440  P  110
    BEGIN IF XREF THEN CROSSREFIT(INDEX,SEQNO,REFTYPE); END#,            01007445  P  110
                %                                                        01007450  P  110
  XMARK(REFTYPE) =      % DEFINE TO CHANGE LAST ENTRY IN REF TABLE TO A 01007455  P  110
    BEGIN IF XREF THEN XREFAY2[XREFPT-1].TYPEREF := REFTYPE END#,        01007460  P  110
```

```
                        %                                                       01007465  P  110
  XREFDUMP(INDEX) =     % DEFINE TO DUMP SYMBOL TABLE INFO FOR IDENTIFIER      01007470  P  110
    BEGIN IF DEFINING.[1:1] THEN CROSSREFDUMP(INDEX); END#,                     01007475  P  110
                        %                                                       01007480  P  110
  XREFINFO[INDEX] =     % DEFINE TO TRANSLATE INFO ROW AND COLUMN TO           01007481  P  110
    XINFO[(INDEX).LINKR,(INDEX).LINKC DIV 2]#, % XINFO ROW AND COL              01007482  P  110
                        %                                                       01007483  P  110
  FORWARDREF = 0#,      % DEFINES FOR DIFFERENT REFERENCE TYPES                 01007485  P  110
  LBLREF     = 1#,      %                                                      01007486  P  110
  DECLREF    = 2#,      %                                                      01007490  P  110
  NORMALREF  = 4#,      %                                                      01007495  P  110
  ASSIGNREF  = 5#;      %                                                      01007500  P  110
$VOIDT                                                                          01723000  P  110
%***************************************************************************02001605  P  110
%                                                                               02001610  P  110
%       MISCELLANEOUS CROSS REFFRENCE PROCEDURES                                02001615  P  110
%                                                                               02001620  P  110
$ VOIDT 02001831                                                                02001625  P  110
%***************************************************************************02001630  P  110
%                                                                               02001635  P  110
PROCEDURE CROSSREFIT(INDEX,SEQNO,REFTYPE);                                      02001640  P  110
  VALUE INDEX,SEQNO,REFTYPE;                                                    02001645  P  110
  REAL  INDEX,SEQNO,REFTYPE;                                                    02001650  P  110
BEGIN                                                                           02001655  P  110
 IF XREFINFO[INDEX].IDNOF # 0 THEN % SAVE                                       02001660  P  110
 BEGIN                                                                          02001665  P  110
  IF XREFPT > 29 THEN % NO SLOTS LEFT IN ARRAY, WRITE IT OUT,                   02001670  P  110
    BEGIN                                                                       02001675  P  110
      WRITE(DSK2,30,XREFAY2[*]);                                                02001680  P  110
      XREFPT := 0;                                                              02001685  P  110
    END;                                                                        02001690  P  110
  XREFAY2[XREFPT] := SEQNO & REFTYPE TYPEREF & XREFINFO[INDEX]                   02001695  P  110
                                     REFIDNOF;                                  02001700  P  110
  XREFPT := XREFPT + 1; % EVEN THOUGH THE ARRAY MAY BE FULL NOW WE              02001705  P  110
                        % CANT WRITE IT OUT BECAUSE SOME ROUTINES                02001710  P  110
                        % WILL LOOK BACK AT THE ENTRY WE JUST PUT               02001715  P  110
                        % IN AND FIX IT UP.                                     02001720  P  110
 END;                                                                           02001725  P  110
END OF CROSSREFIT;                                                              02001730  P  110
%                                                                               02001735  P  110
PROCEDURE CROSSREFDUMP(INDEX);                                                  02001740  P  110
  VALUE INDEX;                                                                  02001745  P  110
  REAL  INDEX;                                                                  02001750  P  110
BEGIN                                                                           02001755  P  110
  STREAM PROCEDURE MOVEXREFINFO(S,D,N);                                         02001760  P  110
    VALUE N;                                                                    02001765  P  110
  BEGIN                                                                         02001770  P  110
    SI := D; DI := D; DS := 8 LIT " "; DS := 7 WDS; % BLANK RECORD              02001775  P  110
    SI := S; SI := SI + 3; DI := D; DS := N CHR; % MOVE IDENTIFIER              02001780  P  110
  END OF MOVEXREFINFO;                                                          02001785  P  110
  %                                                                             02001790  P  110
 IF XREFINFO[INDEX].IDNOF # 0 THEN % DUMP IT                                    02001795  P  110
 BEGIN                                                                          02001800  P  110
  MOVEXREFINFO(INFO[INDEX.LINKR,INDEX.LINKC+1],XREFAY1[*],                      02001805  P  110
               TAKE(INDEX+1),[12:6]);                                           02001810  P  110
  XREFAY1[8] := XREFINFO[INDEX];                                                02001815  P  110
  XREFAY1[9] := TAKE(INDEX); % ELBAT WORD                                       02001820  P  110
```

```
  WRITE(DSK1,10,XREFAY1[1*])$                                          02001821  P  110
  XREFINFO[INDEX] := 0$                                                02001822  P  110
 END$                                                                  02001825  P  110
END OF CROSSREFDUMP$                                                   02001830  P  110
$VOIDT 02882921                                                        02882099  P  110
         IF GT1 #1 AND NOT MACROID THEN % NOT RESERVED WORD            02882100  P  110
           XREFIT(T.LINK,CARDNUMBER,NORMALREF)$ % BUILD XREF ENTRY     02882200  P  110
             XMARK(LBLREF)$ % THIS WILL SORT AHEAD OF DECLARATION      07598100  P  110
                            % WHEN WE GET AROUND TO THE XREF.          07598200  P  110
    XMARK(ASSIGNREF)$ % FILL STATEMENT                                 07660600  P  110
                 XMARK(ASSIGNREF)$ % FOR STATEMENT                     08178100  P  110
        XREFDUMP(POINTER)$ % DUMP XREF INFO                            13283500  P  110
$ VOIDT 13301731                                                       13301099  P  110
         IF XREF THEN % MAKE DECLARATION REFERENCE                     13310050  P  110
%         IF (ACCUM[0].CLASS # DEFINEDID OR NOT                        13310075  P  110
%             BOOLEAN(ACCUM[0].FORMAL)) THEN % NOT DEFINE PARAMETER    13310080  P  110
$ VOIDT 13310601                                                       13310099  P  110
           BEGIN                                                       13310100  P  110
             XREFINFO[NEXTINFO] :=                                     13310200  P  110
               IF SPECTOG THEN                                         13310300  P  110
                 XREFINFO[ELBAT[1]]                                    13310350  P  110
               ELSE                                                    13310400  P  110
                 ((XLUN := XLUN + 1) & SGNO SEGNOF)$                   13310450  P  110
             IF SPECTOG THEN % JUST GO BACK AND FIX UP XREF ENTRY      13310500  P  110
               XMARK(DECLREF)                                          13310525  P  110
             ELSE                                                      13310550  P  110
               XREFIT(NEXTINFO,CARDNUMBER,IF PTOG AND NOT STREAMTOG    13310575  P  110
                      THEN NORMALREF ELSE DECLREF)$                    13310580  P  110
           END                                                         13310600  P  110
%         ELSE % DEFINE PARAMETERS - DONT CROSS REF.                   13310700  P  110
%           XREFINFO[NEXTINFO] := 0                                    13310750  P  110
         ELSE                                                          13310800  P  110
           IF DEFINING.[1:1]  THEN % WE ARE DOING XREFING              13310900  P  110
             XREFINFO[NEXTINFO] := 0$                                  13310950  P  110
           IF XREF AND NOT SPECTOG THEN % ERASE PREVIOUS XREF ENTRY.   13341200  P  110
       XMARK(DECLREF)$ % PROCEDURE DECLARED FORWARD. MARK LAST         14312100  P  110
                       % XREF ENTRY AS A DECLARATION.                 14312101  P  110
         XREFDUMP(J)$ % DUMP XREF INFO                                 14445100  P  110
           XREFIT(PROINFO,0,FORWARDREF)$ % WE NEED THIS SO WE CAN FIND 14469100  P  110
                                 % THE FORWARD DECL. DURING XREF       14469101  P  110
       REAL REMEMBERSEQNO$ % REMEMBERS SEQUENCE NUMBER OF VARIABLE     15075550  P  110
                           % ON LEFT HAND SIDE OF ASSIGNMENT SO WE     15075551  P  110
                           % CAN XREF IT CORRECTLY.                    15075552  P  110
            REMEMBERSEQNO := CARDNUMBER$                               15085100  P  110
       XMARK(ASSIGNREF)$ % ASSIGNMENT TO SIMPLE VARIABLE,              15091100  P  110
$VOIDT                                                                 15092010  P  110
                         BEGIN                                         15113100  P  110
                           BEGIN                                       15115000  P  110
                             ERR(201)$ % PARTIAL WORD NOT LEFT-MOST    15115100  P  110
                             GO TO EXIT$                               15115200  P  110
                           END$                                        15115300  P  110
                           XREFIT(TALL,REMEMBERSEQNO,ASSIGNREF)$       15116000  P  110
                           GO TO L1$                                   15116100  P  110
                         END$                                          15116200  P  110
    JAZZ$          XREFIT(TALL,REMEMBERSEQNO,ASSIGNREF)$ STEPIT$ AEXP$ 15233012  P  110
                   XREFIT(TALL,REMEMBERSEQNO,ASSIGNREF)$ % ASSIGNMENT T015301100  P  110
                                 % SUBSCRIPTED VARIABLE.               15301200  P  110
```

```
                    IF STEP1 = ASSIGNOP THEN                                    15342000  P  110
                      IF P1 = FS THEN % PARTIAL WORD IS LEFT-MOST                15342100  P  110
                        BEGIN                                                    15342200  P  110
                          XREFIT(TALL,REMEMBERSEQNO,ASSIGNREF); % PARTIAL        15342300  P  110
                                % WORD ASSIGNMENT TO SUBSCR. VAR.                15342400  P  110
                          GO TO LAST;                                            15342500  P  110
                        END                                                      15342600  P  110
          XMARK(LBLREF); % MARK LABEL OCCURENCE FOR XREF                         16159100  P  110
            XMARK(ASSIGNREF);                                                    16318500  P  110
IF (XREF OR DEFINING.[1:1]) AND XLUN > 0 THEN                                    16495300  P  110
    DEFINE XREFINFO[INDEX] = INFO((INDEX).CF DIV 2).[33:7],                      17002005  P  110
                             ((INDEX).CF DIV 2).LINKC]#,                         17002006  P  110
           CF = [33:15]#,                                                        17002007  P  110
           FF = [18:15]#,                                                        17002008  P  110
           NEWID[INDEX] = (IF BOOLEAN(INDEX) THEN XREFINFO[INDEX].FF             17002009  P  110
                           ELSE XREFINFO[INDEX].CF)#;                            17002010  P  110
    ARRAY TIMINGS[0:2,0:3];                                                      17002012  P  110
    PROCEDURE SAVETIMES(I);                                                      17002015  P  110
      VALUE I; INTEGER I;                                                        17002020  P  110
    BEGIN                                                                        17002025  P  110
      INTEGER J;                                                                 17002030  P  110
      FOR J := 1 STEP 1 UNTIL 3 DO                                               17002035  P  110
        TIMINGS[I,J] := TIME(J);                                                 17002040  P  110
    END;                                                                         17002045  P  110
    PROCEDURE UPDATETIMES(I);                                                    17002050  P  110
      VALUE I; INTEGER I;                                                        17002055  P  110
    BEGIN                                                                        17002060  P  110
      INTEGER J;                                                                 17002065  P  110
      FOR J := 1 STEP 1 UNTIL 3 DO                                               17002070  P  110
        TIMINGS[I,J] := TIME(J) - TIMINGS[I,J];                                  17002075  P  110
    END;                                                                         17002080  P  110
    SAVETIMES(0); % SAVE TIMES FOR START OF IDENTIFIER SORT.                     17002525  P  110
    TOTALNO := XLUN; % REMEMBER NUMBER OF IDENTIFIERS.                           17004500  P  110
    FOR I := 0 STEP 1 UNTIL 8191 DO                                              17004700  P  110
      PUT(0,I);                                                                  17004710  P  110
              BEGIN                                                              17022000  P  110
                REWIND(DSK1);                                                    17022100  P  110
                UPDATETIMES(0); % UPDATE TIMES FOR IDENTIFIER SORT.              17022200  P  110
                TIMINGS[0,0] := XLUN; % NUMBER OF IDENTIFIERS SORTED.            17022300  P  110
              END                                                                17022400  P  110
              IF BOOLEAN(A[8]) THEN                                              17025000  P  110
                XREFINFO[A[8]].FF := XLUN := XLUN + 1                            17025100  P  110
              ELSE                                                               17025200  P  110
                XREFINFO[A[8]].CF := XLUN := XLUN + 1;                           17025300  P  110
              A[8].IDNOF := XLUN;                                                17025400  P  110
            IF 63 SC < DC THEN                                                   17033000  P  110
              TALLY := 1                                                         17033100  P  110
            ELSE                                                                 17033200  P  110
              BEGIN                                                              17033300  P  110
                S1 := A;                                                         17033400  P  110
                D1 := B;                                                         17033500  P  110
                IF 63 SC = DC THEN                                               17033600  P  110
                  TALLY := 2;                                                    17033700  P  110
              END;                                                               17033800  P  110
            DS := 8 LIT 3"777777777"; % ID.NO. AND SEG.NO. FIELDS                17041100  P  110
                IF REAL(COMP1:=COMPS1(A,B)) = 2 THEN % IDS EQUAL                 17042300  P  110
                  COMP1 := A[8].IDNOF < B[8].IDNOF;                              17042350  P  110
```

```
        SORT(OUTPUT1,INPUT1,0,HV1,COMP1,10,IF TOTALNO < 1000 THEN      17045000  P  110
                    7000 ELSE 10000);                                  17045100  P  110
$ VOIDT 17069001                                                       17047001  P  110
    ARRAY IDTYPE[0:(IDMAX+3)*4-1];                                     17047100  P  110
    STREAM PROCEDURE SETUPHEADING(S,D,SEG,SEQNO,FWDTOG,LBLTOG,         17047200  P  110
                                  FWDSEQNO,TYPE,OWNTOG,PARAMTOG,       17047300  P  110
                                  VALTOG);                             17047350  P  110
      VALUE SEG,SEQNO,FWDTOG,LBLTOG,FWDSEQNO,OWNTOG,PARAMTOG,          17047400  P  110
        VALTOG;                                                        17047450  P  110
    BEGIN                                                              17047500  P  110
      SI := S;                                                         17047700  P  110
      DI := D;                                                         17047800  P  110
      63 (IF SC = " " THEN JUMP OUT ELSE DS := CHR);                   17047900  P  110
      DS := 6 LIT "  --  ";                                           17048000  P  110
      OWNTOG (DS := 4 LIT "OWN ");                                     17048100  P  110
      SI := TYPE;                                                      17049300  P  110
      32 (IF SC = "," THEN JUMP OUT ELSE DS := CHR);                   17049400  P  110
      PARAMTOG (DS := 6 LIT "  --  ";                                 17049410  P  110
                DS := 4 LIT "NAME";                                    17049420  P  110
                VALTOG (DI := DI - 4; DS := 5 LIT "VALUE");            17049430  P  110
                DS := 10 LIT " PARAMETER");                            17049440  P  110
      DS := 18 LIT "  --  DECLARED AT ";                               17049500  P  110
      SI := LOC SEQNO;                                                 17050400  P  110
      DS := 8 DEC;                                                     17050500  P  110
      FWDTOG (DS := 17 LIT "  --  FORWARD AT ";                       17050600  P  110
              SI := LOC FWDSEQNO;                                      17050700  P  110
              DS := 8 DEC);                                            17050800  P  110
      LBLTOG (DS := 16 LIT "  --  OCCURS AT ";                        17050900  P  110
              SI := LOC FWDSEQNO;                                      17051000  P  110
              DS := 8 DEC);                                            17051100  P  110
    END OF SETUPHEADING;                                               17051200  P  110
                                                                       17051300  P  110
    STREAM PROCEDURE ADDASEQNO(SEQNO,N,STARS,D);                       17051400  P  110
      VALUE SEQNO,N,STARS;                                             17051500  P  110
    BEGIN                                                              17051600  P  110
      DI := D;                                                         17051700  P  110
      DI := DI + 8;                                                    17051800  P  110
      N (DI := DI + 10);                                               17051900  P  110
      STARS(DI := DI - 1; DS := LIT "*");                              17052000  P  110
      SI := LOC SEQNO;                                                 17052100  P  110
      DS := 8 DEC;                                                     17052200  P  110
      DS := LIT " ";                                                   17052300  P  110
      STARS (DI := DI - 1; DS := LIT "*");                             17052400  P  110
    END;                                                               17052500  P  110
    STREAM PROCEDURE BLANKET(D);                                       17052600  P  110
    BEGIN                                                              17052700  P  110
      DI := D;                                                         17052800  P  110
      DS := 8 LIT " ";                                                 17052900  P  110
      SI := D;                                                         17053000  P  110
      DS := 16 WDS;                                                    17053100  P  110
    END OF BLANKET;                                                    17053200  P  110
    PROCEDURE PRINTXREFSTATISTICS;                                     17053300  P  110
    BEGIN                                                              17053400  P  110
      SWITCH FORMAT STATS :=                                           17053500  P  110
        (///, "CROSS REFERENCE STATISTICS", /,                         17053600  P  110
              "------ -------- ----------", /),                       17053700  P  110
        ("PHASE ONE - SORT",I6," IDENTIFIERS"),                        17053800  P  110
```

```
    ("PHASE TWO - SORT",I7," REFERENCES"),                                  17053900  P  110
    ("PHASE THREE - PRINT CROSS REFERENCE (",I7," LINES)"),                 17054000  P  110
    (X5,I4,":",2I1," ELAPSED TIME (MIN:SEC)"),                              17054100  P  110
    (X5,I4,":",2I1," PROCESSOR TIME"),                                      17054200  P  110
    (X5,I4,":",2I1," I/O TIME",/)];                                         17054300  P  110
  INTEGER I,J,K;                                                            17054400  P  110
  WRITE(LINE,STATS[0]);                                                     17054500  P  110
  FOR I := 0 STEP 1 UNTIL 2 DO                                              17054600  P  110
    BEGIN                                                                   17054700  P  110
      WRITE(LINE,STATS[I+1],TIMINGS[I,0]);                                  17054800  P  110
      FOR J := 1 STEP 1 UNTIL 3 DO                                          17054900  P  110
        BEGIN                                                               17055000  P  110
          K := (TIMINGS[I,J] + 30) DIV 60; % ROUND TO NEAREST SECON         17055010  P  110
          WRITE(LINE,STATS[J+3],K DIV 60,(K:=K MOD 60) DIV 10,              17055020  P  110
                                         K MOD 10);                         17055025  P  110
        END;                                                                17055030  P  110
    END;                                                                    17055100  P  110
END PRINTXREFSTATISTICS;                                                    17055200  P  110
    DEFINE REFCOUNT = TIMINGS[1,0]#; % NUMBER OF REFERENCES SORTED.         17069300  P  110
    BOOLEAN FIRSTTIME; % TRUE ON FIRST CALL OF OUTPUT PROCEDURE.            17069400  P  110
    REAL LASTADDRESS;                                                       17069600  P  110
      DEFINE I = LASTADDRESS#;                                              17073100  P  110
    A[0] := I & NEWIDX[I,REFIDNOF] REFIDNOF;                                17080000  P  110
    REFCOUNT := REFCOUNT + 1;                                               17080100  P  110
    XREFAY1[8] := XREFPT := LASTADDRESS := 0;                               17084000  P  110
    FILL IDTYPE[*] WITH                                                     17084010  P  110
      "UNKNOWN.                          ", % 0                             17084020  P  110
      "STREAM LABEL.                     ", % 1                             17084030  P  110
      "STREAM VARIABLE.                  ", % 2                             17084040  P  110
      "DEFINE.                           ", % 3                             17084050  P  110
      "LIST.                             ", % 4                             17084060  P  110
      "FORMAT.                           ", % 5                             17084070  P  110
      "SWITCH FORMAT.                    ", % 6                             17084080  P  110
      "REAL SUBROUTINE.                  ", % 7                             17084090  P  110
      "SUBROUTINE.                       ", % 8                             17084100  P  110
      "SWITCH LABEL.                     ", % 9                             17084110  P  110
      "PROCEDURE.                        ", % 10                            17084120  P  110
      "INTRINSIC.                        ", % 11                            17084130  P  110
      "STREAM PROCEDURE.                 ", % 12                            17084140  P  110
      "BOOLEAN STREAM PROCEDURE.         ", % 13                            17084150  P  110
      "REAL STREAM PROCEDURE.            ", % 14                            17084160  P  110
      "INTEGER STREAM PROCEDURE.         ", % 15                            17084170  P  110
      "INTEGER STREAM PROCEDURE.         ", % 16                            17084180  P  110
      "BOOLEAN PROCEDURE.                ", % 17                            17084182  P  110
      "REAL PROCEDURE.                   ", % 18                            17084184  P  110
      "INTEGER PROCEDURE.                ", % 19                            17084186  P  110
      "INTEGER PROCEDURE.                ", % 20                            17084188  P  110
      "BOOLEAN.                          ", % 21                            17084190  P  110
      "REAL.                             ", % 22                            17084200  P  110
      "INTEGER.                          ", % 23                            17084210  P  110
      "INTEGER.                          ", % 24                            17084220  P  110
      "BOOLEAN ARRAY.                    ", % 25                            17084230  P  110
      "REAL ARRAY.                       ", % 26                            17084240  P  110
      "INTEGER ARRAY.                    ", % 27                            17084250  P  110
      "INTEGER ARRAY.                    ", % 28                            17084260  P  110
      "                                  ", % 29                            17084270  P  110
      "NAME.                             ", % 30                            17084280  P  110
```

```
          "INTEGER NAME.                          ", % 31                    17084290  P  110
          "LABEL.                                 "; % 32                    17084300  P  110
        LABEL EOF2, SKIP;                                                    17091100  P  110
        OWN BOOLEAN B2, FWDTOG, LBLTOG, WAITINGFORFWDREF;                    17091110  P  110
        DEFINE MATCH(A,B) = REAL(BOOLEAN(A) EQV BOOLEAN(B)) =                17091115  P  110
                            REAL(NOT FALSE)#;                                17091116  P  110
        REAL I;                                                              17091120  P  110
        DEFINE LINECOUNT = TIMINGS[2,0]#; % NUMBER OF LINES PRINTED.         17091140  P  110
        OWN REAL FWDSEQNO;                                                   17091150  P  110
        IF FIRSTTIME THEN % PRINT HEADINGS AND SAVE TIMINGS.                 17091155  P  110
          BEGIN                                                              17091160  P  110
            FIRSTTIME := FALSE;                                              17091162  P  110
            TIME1 := TIME(1);                                                17091165  P  110
            DATIME;                                                          17091170  P  110
            UPDATETIMES(1);                                                  17091175  P  110
            SAVETIMES(2); % SAVE TIMES FOR START OF XREF PRINT.              17091180  P  110
          END;                                                               17091200  P  110
$VOIDT 17111001                                                              17091201  P  110
        IF NOT B2 THEN                                                       17091210  P  110
          IF B THEN % END OF SORT - LIST OUT REST OF SEQ. NO.                 17091300  P  110
            IF XREFPT ≠ 0 THEN % WE GOT SOME TO LIST OUT                     17091400  P  110
              BEGIN                                                          17091500  P  110
                WRITE(LINE[DBL],15,PAY[*]);                                  17091510  P  110
                LINECOUNT := LINECOUNT + 1;                                  17091520  P  110
              END                                                            17091530  P  110
            ELSE % NOTHING TO LIST OUT                                       17091600  P  110
          ELSE % NOT END OF SORT                                             17091700  P  110
            IF NOT MATCH(LASTADDRESS,A[0]) AND A[0].REFIDNOF ≠ 0 AND         17091800  P  110
               A[0].REFIDNOF ≥ XREFAY1[8].IDNOF THEN                         17091900  P  110
              IF A[0].TYPEREF = FORWARDREF THEN %                            17092000  P  110
                WAITINGFORFWDREF := TRUE                                     17092100  P  110
              ELSE                                                           17092200  P  110
                IF A[0].TYPEREF = LBLREF THEN %                              17092300  P  110
                  BEGIN                                                      17092400  P  110
                    LBLTOG := TRUE;                                          17092500  P  110
                    FWDSEQNO := A[0].SEQNOF;                                 17092600  P  110
                  END                                                        17092700  P  110
                ELSE                                                         17092800  P  110
                  IF A[0].TYPEREF = DECLREF THEN                             17092900  P  110
                    IF WAITINGFORFWDREF THEN % THIS MUST BE IT               17093000  P  110
                      BEGIN                                                  17093100  P  110
                        WAITINGFORFWDREF := FALSE;                           17093200  P  110
                        FWDTOG := TRUE;                                      17093300  P  110
                        FWDSEQNO := A[0].SEQNOF;                             17093400  P  110
                      END                                                    17093500  P  110
                    ELSE % ITS A NORMAL DECLARATION - NOT FORWARD            17093600  P  110
                      BEGIN                                                  17093700  P  110
                        IF A[0].REFIDNOF > XREFAY1[8].IDNOF THEN             17093850  P  110
                          DO                                                 17093900  P  110
                            READ(DSK1,10,XREFAY1[*]) [EOF2]                  17093950  P  110
                          UNTIL                                              17094000  P  110
                            A[0].REFIDNOF ≤ XREFAY1[8].IDNOF;                17094050  P  110
                        IF A[0]. REFIDNOF < XREFAY1[8].IDNOF THEN            17094100  P  110
                          GO TO SKIP;                                        17094150  P  110
                        IF XREFPT > 0 THEN % THERE IS STUFF TO PRINT         17094200  P  110
                         BEGIN                                               17094240  P  110
                          IF SINGLTOG THEN                                   17094250  P  110
                            WRITE(LINE,15,PAY[*])                            17094300  P  110
```

```
      ELSE                                                          17094350  P  110
        WRITE(LINE[DBL],15,PAY[*]);                                 17094400  P  110
      LINECOUNT := LINECOUNT + 1;                                   17094410  P  110
     END                                                            17094420  P  110
    ELSE                                                            17094450  P  110
      IF NOT SINGLTOG THEN                                          17094500  P  110
        WRITE(LINE);                                                17094550  P  110
    XREFPT := 0;                                                    17094600  P  110
    BLANKFT(PAY[*]);                                                17094650  P  110
     SETUPHEADING(XREFAY1[*],PAY[*],XREFAY1[8],                     17094700  P  110
       SEGNOF,A[0].SEQNOF,FWDTOG,LBLTOG,                            17094800  P  110
      FWDSEQNO,IDTYPE[(IF (I :=                                     17094900  P  110
      XREFAY1[9].CLASS) > IDMAX THEN                                17095000  P  110
      0 ELSE I) * 4],                                               17095100  P  110
      REAL(I ≥ BOOID AND XREFAY1[9].[9:2] = 1),                     17095300  P  110
      REAL((I ≥ BOOID OR I = LOCLID) AND BOOLEAN                    17095310  P  110
      (XREFAY1[9].[9:1])), XREFAY1[9].[10:1]);                      17095320  P  110
    FWDTOG := LBLTOG := FALSE;                                      17095400  P  110
    WRITE(LINE,15,PAY[*]);                                          17095500  P  110
    LINECOUNT := LINECOUNT + 1;                                     17095510  P  110
    BLANKFT(PAY[*]);                                                17095550  P  110
  END                                                               17095600  P  110
ELSE % IT MUST BE A NORMAL REFERENCE                                17095700  P  110
  IF A[0].SEQNOF ≠ LASTADDRESS.SEQNOF THEN                          17095750  P  110
    BEGIN                                                           17095800  P  110
      ADDASEQNO(A[0].SEQNOF,XREFPT,A[0].[5:1],                      17095900  P  110
                PAY[*]);                                            17096000  P  110
      IF (XREFPT := XREFPT + 1) = 11 THEN %FULL                     17096100  P  110
        BEGIN                                                       17096200  P  110
          WRITE(LINE,15,PAY[*]);                                    17096300  P  110
         LINECOUNT := LINECOUNT + 1;                                17096350  P  110
          XREFPT := 0;                                              17096400  P  110
          BLANKET(PAY[*]);                                          17096450  P  110
        END                                                         17096500  P  110
    END                                                             17096550  P  110
  ELSE % REFERENCE TO SAME SEQ. NO, SKIP IT                         17096575  P  110
ELSE % THIS IS A REFERENCE TO THE SAME SEQ. NO. - SKIP              17096600  P  110
FLSE % HIT END OF IDENTIFIER FILE - JUST SKIP OVER REFERENCES       17096700  P  110
  EOF2: B2 := TRUE; %  SO SORT CAN GO TO NORMAL EOJ                 17096800  P  110
IF NOT B THEN SKIP; LASTADDRESS := A[0];                            17096850  P  110
END OF OUTPUT2;                                                     17096900  P  110
  A[0] := 3"777777777777777"; % BIGGEST FLOATING PT. NO.            17114000  P  110
    COMP2 := IF A[0].REFIDNOF < B[0].REFIDNOF THEN % DIF IDS        17117000  P  110
               TRUE                                                 17117100  P  110
             ELSE                                                   17117200  P  110
               IF A[0].REFIDNOF = B[0].REFIDNOF THEN                17117300  P  110
                 IF A[0].[1:4] LSS B[0].[1:4] THEN                  17117400  P  110
                   TRUE                                             17117500  P  110
                 ELSE                                               17117600  P  110
                   IF A[0].[1:4] = B[0].[1:4] THEN                  17117700  P  110
                     IF A[0].SEQNOF < B[0].SEQNOF THEN              17117702  P  110
                       TRUE                                         17117704  P  110
                     ELSE                                           17117706  P  110
                       IF A[0].SEQNOF = B[0].SEQNOF THEN            17117708  P  110
                         BOOLEAN(A[0].[5:1])                        17117710  P  110
                       ELSE                                         17117712  P  110
                         FALSE                                      17117714  P  110
```

```
                                FLSE                                    17117720  P  110
                                  FALSE                                 17117730  P  110
                              ELSE                                      17117800  P  110
                                FALSE;                                  17117900  P  110
        SAVETIMES(1); % SAVE TIMES FOR START OF REFERENCES SORT         17117910  P  110
        FIRSTTIME := TRUE; % LET OUTPUT PROCEDURE KNOW ABOUT FIRST CAL  17117920  P  110
        SORT(OUTPUT2,INPUT2,0,HV2,COMP2,1,6000);                        17119000  P  110
       UPDATETIMES(2); % UPDATE TIMES FOR PRINTING CROSS REFERENCE      17119100  P  110
       PRINTXREFSTATISTICS;                                             17119200  P  110
$! DATE 2/1/76                                                          99990000  P  110
$! BY JTC - MSA CENTRAL.                                               99990100  P  110
$!                                                                      99990200  P  110
$! THIS PATCH IS NEARLY A COMPLETE REWRITE OF THE ALGOL                 99990300  P  110
$! CROSS REFERENCE ROUTINES IN AN ATTEMPT TO REDUCE                     99990400  P  110
$! THE AMOUNT OF PAPER PRODUCED AND ALSO TO PROVICE MORE                99990500  P  110
$! INFORMATION ABOUT THE IDENTIFIERS.  IN PARTICULAR,                   99990600  P  110
$! THE FOLLOWING CHANGES HAVE BEEN MADE:                               99990700  P  110
$!                                                                      99990800  P  110
$! 1). ONE LINE IS LISTED IN THE CROSS REFERENCE FOR EACH               99990900  P  110
$!      IDENTIFIER WHICH INCLUDES THE FOLLOWING INFORMATION:            99991000  P  110
$!        A. THE NAME OF THE IDENTIFIER.                                99991100  P  110
$!        B. THE CLASS OF THE IDENTIFIER, E.G., REAL,                   99991200  P  110
$!           INTEGER, BOOLEAN, PROCEDURE, ETC.                          99991300  P  110
$!        C. IF THE IDENTIFIER IS A FORMAL PARAMETER IT IS              99991400  P  110
$!           MARKED AS A "NAME PARAMETER" OR "VALUE PARAMETER".         99991500  P  110
$!        D. THE SEGMENT AND SEQUENCE NUMBER AT WHICH THE               99991600  P  110
$!           IDENTIFIER IS DECLARED.                                    99991700  P  110
$!        E. IF THE IDENTIFIER IS A PROCEDURE IDENTIFIER, THEN          99991800  P  110
$!           IF THE PROCEDURE IS DECLARED FORWARD THE SEQUENCE          99991900  P  110
$!           NUMBER OF THE FORWARD DECLARATION IS LISTED.               99992000  P  110
$!        F. IF THE IDENTIFIER IS A LABEL IDENTIFIER, THE SEQUENCE      99992100  P  110
$!           NUMBER AT WHICH THE LABEL OCCURS IS LISTED.                99992200  P  110
$!                                                                      99992300  P  110
$! 2). ASSIGNMENTS TO IDENTIFIERS ARE NOW STARRED IN THE                99992400  P  110
$!      CROSS REFERENCE IN THE FOLLOWING ADDITIONAL CASES:              99992500  P  110
$!      A. WHEN A SUBSCRIPTED VARIABLE IS USED IN ASSIGNMENT.           99992600  P  110
$!      B. WHEN AN ARRAY ROW IS USED IN A FILL STATEMENT.               99992700  P  110
$!      C. WHEN AN ARRAY ROW IS USED IN A SEARCH STATEMENT.             99992800  P  110
$!      C. WHEN A SIMPLE VARIABLE IS USED IN A FOR STATEMENT.           99992900  P  110
$!      D. WHEN A LOCAL VARIABLE OR STREAM PARAMETER IS CHANGED         99993000  P  110
$!         BY BEING ASSIGNED THE VALUE OF SI,DI,CI OR TALLY.            99993100  P  110
$!                                                                      99993200  P  110
$! 3). DEFINE PARAMETERS ARE NO LONGER CROSS REFERENCED.                99993300  P  110
$!                                                                      99993400  P  110
$! 4). IDENTIFIERS REFERRED TO IN THE RIGHT HAND SIDE OF A              99993500  P  110
$!      DEFINE DECLARATION ARE NO LONGER CROSS                          99993600  P  110
$!      REFERENCED AGAINST THE DEFINE DECLARATION.                      99993700  P  110
$!                                                                      99993800  P  110
$! 5). IF NO IDENTIFIERS WERE CROSS REFERENCED, THE IDENTIFIER          99993900  P  110
$!      SORT WOULD BLOW UP WITH AN I/O ERROR 86.  NOW, XREF             99994000  P  110
$!      CHECKS TO MAKE SURE THERE IS SOMETHING TO SORT BEFORE           99994100  P  110
$!      BEGINNING.                                                      99994200  P  110
$!                                                                      99994300  P  110
$! 6). IT SHOULD NOW BE POSSIBLE TO SET AND RESET XREF                  99994400  P  110
$!      INDISCRIMINANTLY DURING A COMPILE WITHOUT CAUSING THE XREF      99994500  P  110
$!      TO GET CONFUSED.  PREVIOUSLY, REFERENCES WOULD GET              99994600  P  110
$!      ASSOCIATED WITH THE WRONG IDENTIFIERS.  THE RULE NOW IS THAT    99994700  P  110
```

```
$!       XREF MUST BE ON AT THE POINT OF DECLARATION FOR AN                 99994800  P  110
$!       IDENTIFIER TO BE CROSS REFERENCED AT ALL.                          99994900  P  110
$!                                                                          99995000  P  110
$! 7). PREVIOUSLY, WHEN THE SAME IDENTIFIER OCCURRED IN                     99995100  P  110
$!       DIFFERENT BLOCKS, THE IDENTIFIERS WOULD BE LISTED                  99995200  P  110
$!       IN NO PARTICULAR ORDER. NOW, THE IDENTIFIER THAT                   99995300  P  110
$!       APPEARS FIRST WILL BE LISTED FIRST IN THE XREF.                    99995400  P  110
$!                                                                          99995500  P  110



$#PATCH NUMBER 111 FOR ESPOL CONTAINS 3 CARDS;                             00000000  P  111

    IF MERGETOG THEN % INDICATE NAME OF SOURCE FILE.                       01835600  P  111
      WRITE(LINE,<X40,"SOURCE FILE: ",A1,A6,"/",A1,A6,//>,                 01835700  P  111
        (N1:=TAPE.MFID).[6:6],N1,(N2:=TAPE.FID).[6:6],N2);                 01835800  P  111
$! DATE 2/6/76                                                             99990000  P  111
$! BY JTC - MSA CENTRAL;                                                   99990100  P  111
$!                                                                         99990200  P  111
$! THIS PATCH PRINTS THE NAME OF THE SOURCE FILE BELOW                     99990300  P  111
$! THE NAME OF THE OBJECT FILE ON THE PRINTED LISTING IF TAPE              99990400  P  111
$! IS SET AT THE TIME THE HEADING IS PRINTED.                              99990500  P  111



$#PATCH NUMBER 112 FOR ESPOL CONTAINS 116 CARDS.                           00000000  P  112

094          PARSE: MISSING RIGHT BRACKET                                  00069975  P  112
618      BLOCK: AUXMEM APPEARS IMMEDIATELY BEFORE IDENTIFIER (NO TYPE)     00418000  P  112
                  SBITF =[21:6]#, % STARTING BIT FOR FIELD ID.             01154200  P  112
                  NBITF =[27:6]#, % NUMBER OF BITS FOR FIELD ID.           01154300  P  112
         FIELDID        =125#,    COMMENT 175;                            01278700  P  112
         DEFINEV        =19#,     COMMENT  23;                            01298000  P  112
         AUXMEMV        =20#,     COMMENT 24;                             01298500  P  112
         FIELDV         =21#;     COMMENT 25;                             01298600  P  112
DEFINE LASTSEQUENCE = 147#,                                                01569000  P  112
                IF STEPI = FIELDID THEN % GET INFO FROM INFO               05273100  P  112
                  BEGIN                                                    05273200  P  112
                    FIRST := ELBAT[I].SBITF;                               05273300  P  112
                    SECOND := ELBAT[I].NBITF;                              05273400  P  112
                    GO TO EXIT;                                            05273500  P  112
                  END                                                      05273600  P  112
                ELSE                                                       05273700  P  112
                  IF ELCLASS = LFTBRKET THEN                               05273800  P  112
                    IF STEPI = FIELDID THEN                                05273900  P  112
                      BEGIN                                                05274000  P  112
                        FIRST := ELBAT[I].SBITF;                           05274100  P  112
                        SECOND := ELBAT[I].NBITF;                          05274200  P  112
                        IF STEPI = RTBRKET THEN                            05274300  P  112
                          GO TO EXIT;                                      05274400  P  112
                      END                                                  05274500  P  112
                    ELSE                                                   05274600  P  112
                      IF ELCLASS = LITNO THEN                              05275000  P  112
         IF ELCLASS = FIELDID THEN                                         06315100  P  112
           BEGIN                                                           06315200  P  112
```

```
            FIRST := ELBAT[I].SBITF;                                    06315300  P  112
            SECOND := 48 - (THIRD := ELBAT[I].NBITF);                   06315400  P  112
            GO TO NEXTCHK;                                              06315500  P  112
          END                                                           06315600  P  112
        ELSE                                                            06315700  P  112
        IF STEPI = FIELDID THEN                                         06316100  P  112
          BEGIN                                                         06316200  P  112
            FIRST := ELBAT[I].SBITF;                                    06316300  P  112
            SECOND := 48 - (THIRD := ELBAT[I].NBITF);                   06316400  P  112
            IF STEPI ≠ RTBRKET THEN                                     06316500  P  112
              BEGIN                                                     06316600  P  112
                ERR(94);                                                06316700  P  112
                GO TO EXIT;                                             06316800  P  112
              END;                                                      06316900  P  112
            GO TO NEXTCHK;                                              06317000  P  112
          END                                                           06317100  P  112
        ELSE                                                            06317200  P  112
          IF ELCLASS = LITNO THEN                                       06317300  P  112
    OCT1310157730000002, "3SCS00",                          %652        09128600  P  112
    OCT0500000250000000, "5FIELD",                          %654        09128700  P  112
              IF ELCLASS>IDMAX AND ELCLASS≤MULOP                        13337000  P  112
      GOTSCHK,FIELDDEC,AUXMEMERR,                                       14016000  P  112
              STREAMERR,DEFINEDEC,AUXMEMERR,FIELDDEC;                   14021000  P  112
AUXMEMERR:FLAG(618);J:=J+1;GO TO REALDEC;                               14136100  P  112
FIELDDEC:                                                               14269020  P  112
  BEGIN                                                                 14269040  P  112
    REAL SAVEINFO, SB, NB;                                              14269060  P  112
    BOOLEAN FOUNDLB; % TRUE IF LEFT-BRACKET WAS USED IN FIELD SPEC.     14269080  P  112
    LABEL EXIT, SAVEIT;                                                 14269100  P  112
    STOPENTRY := STOPGSP := TRUE;                                       14269120  P  112
    I := I - 1;                                                         14269140  P  112
    DO                                                                  14269160  P  112
      BEGIN                                                             14269180  P  112
        STOPDEFINE := TRUE;                                             14269200  P  112
        STEPIT;                                                         14269220  P  112
        ENTRY(FIELDID);                                                 14269240  P  112
        SAVEINFO := LASTINFO;                                           14269260  P  112
        IF ELCLASS = RELOP AND ACCUM[1] = "1=0000" THEN                 14269280  P  112
          BEGIN                                                         14269300  P  112
            IF STEPI = LFTBRKET THEN % REMEMBER THIS                    14269320  P  112
              BEGIN                                                     14269340  P  112
                FOUNDLB := TRUE;                                        14269360  P  112
                STEPIT;                                                 14269380  P  112
              END                                                       14269400  P  112
            ELSE                                                        14269420  P  112
              FOUNDLB := FALSE;                                         14269440  P  112
            IF ELCLASS = FIELDID THEN                                   14269442  P  112
              BEGIN                                                     14269444  P  112
                SB := ELBAT[I].SBITF;                                   14269446  P  112
                NB := ELBAT[I].NBITF;                                   14269448  P  112
                GO TO SAVEIT;                                           14269450  P  112
              END;                                                      14269452  P  112
            IF ELCLASS = LITNO THEN                                     14269460  P  112
              IF STEPI = COLON THEN                                     14269480  P  112
                IF STEPI = LITNO THEN                                   14269500  P  112
                  IF (SB := ELBAT[I-2].ADDRESS) ×                       14269520  P  112
                      (NB := ELBAT[I].ADDRESS) ≠ 0 AND                  14269540  P  112
                      SB + NB ≤ 48 THEN                                 14269560  P  112
```

```
                BEGIN                                                    14269580  P  112
            SAVEIT:                                                      14269590  P  112
                    PUT(TAKE(SAVEINFO) & SB SBITF & NB NBITF,            14269600  P  112
                      SAVEINFO);                                         14269620  P  112
                    STEPIT;                                              14269640  P  112
                    IF FOUNDLB THEN % BETTER HAVE RIGHT BRACKET.         14269660  P  112
                      IF ELCLASS = RTBRKET THEN                          14269680  P  112
                        BEGIN                                            14269700  P  112
                          STEPIT;                                        14269705  P  112
                          GO TO EXIT;                                    14269710  P  112
                        END                                              14269715  P  112
                      ELSE % MISSING RIGHT BRACKET.                      14269720  P  112
                    ELSE                                                 14269740  P  112
                      GO TO EXIT;                                        14269760  P  112
                  END;                                                   14269780  P  112
          END;                                                           14269800  P  112
        FLAG(114);                                                       14269820  P  112
        DO STEPIT UNTIL ELCLASS = COMMA OR ELCLASS = SEMICOLON;          14269840  P  112
    EXIT:                                                                14269860  P  112
      END                                                                14269880  P  112
    UNTIL                                                                14269900  P  112
      ELCLASS # COMMA;                                                   14269920  P  112
    STOPENTRY := STOPGSP := FALSE;                                       14269940  P  112
END;                                                                     14269960  P  112
GO TO START;                                                             14269980  P  112
    ARRAY IDTYPE[0:(IDMAX+4)*4-1];                                       17047100  P  112
          "LABEL.                                ", % 32                 17084300  P  112
          "FIELD.                                "; % 33 (CLASS = 125)   17084400  P  112
                        XREFAY1[9].CLASS) > IDMAX THEN IF I =            17095000  P  112
                        FIELDID THEN 33 ELSE 0 ELSE I) * 4];             17095100  P  112
$: DATE 2/2/76                                                           99990000  P  112
$: BY JTC - MSA CENTRAL                                                  99990100  P  112
$:                                                                       99990200  P  112
$: THIS PATCH IMPLEMENTS A NEW DECLARATION TYPE OF "FIELD" WHICH         99990300  P  112
$: PROVIDES A CONVENIENT METHOD OF DESIGNATING A PARTIAL WORD FIELD      99990400  P  112
$: A LA THE 6700 ESPOL AND ALGOL MECHANISM. THIS METHOD IS CONSIDERABLY  99990500  P  112
$: MORE EFFICIENT THAN A DEFINE BECAUSE THE PARTIAL WORD FIELD INFOR-    99990600  P  112
$: MATION IS STORED OFF WITH THE IDENTIFIER IN THE SYMBOL TABLE THUS     99990700  P  112
$: MAKING REFERENCES TO THE PARTIAL WORD FIELD VERY INEXPENSIVE.         99990800  P  112
$:                                                                       99990900  P  112
$: THE SYNTAX FOR A PARTIAL WORD DECLARATION IS AS FOLLOWS:              99991000  P  112
$:                                                                       99991100  P  112
$:  <FIELD DECLARATION> ::= FIELD <FIELD LIST>                           99991200  P  112
$:                                                                       99991300  P  112
$:  <FIELD LIST> ::= <FIELD> / <FIELD LIST> , <FIELD>                    99991400  P  112
$:                                                                       99991500  P  112
$:  <FIELD> ::= <FIELD IDENTIFIER> = <FIELD SPECIFICATION>               99991600  P  112
$:                                                                       99991700  P  112
$:  <FIELD IDENTIFIER> ::= <IDENTIFIER>                                  99991800  P  112
$:                                                                       99991900  P  112
$:  <FIELD SPECIFICATION> ::= <INTEGER> : <INTEGER> /                    99992000  P  112
$:                            [ <INTEGER> : <INTEGER> ] /                99992100  P  112
$:                            <FIELD IDENTIFIER> /                       99992200  P  112
$:                            [ <FIELD IDENTIFIER> ]                     99992300  P  112
$:                                                                       99992400  P  112
$: SOME EXAMPLES:                                                        99992500  P  112
$:                                                                       99992600  P  112
```

```
$!       FIELD CF = [33:15];                                                    99992700  P  112
$!                                                                              99992800  P  112
$!       FIELD FF = 18:15,                                                      99992900  P  112
$!           SIZE = 08:10;                                                      99993000  P  112
$!                                                                              99993100  P  112
$! FIELDS MAY BE USED ANYWHERE PARTIAL WORD DESIGNATORS MAY BE USED             99993200  P  112
$! AND MAY BE ENCLOSED IN BRACKETS OR NOT AS DESIRED. ENCLOSING                99993300  P  112
$! THE FIELD IDENTIFIER IN BRACKETS MAY HELP THE READABILITY OF                99993400  P  112
$! THE CODE BUT OF COURSE REQUIRES EXTRA SCANNING  BY THE                      99993500  P  112
$! COMPILER TO PROCESS THE BRACKETS.                                           99993600  P  112
$!                                                                             99993700  P  112
$! SOME EXAMPLES:                                                             99993800  P  112
$!                                                                             99993900  P  112
$!        I := I.FF;                                                           99994000  P  112
$!                                                                             99994100  P  112
$!        J := I.CF & 10 [SIZE] & K FF;                                        99994200  P  112
$!                                                                             99994300  P  112
$!        I.FF := J.[CF];                                                      99994400  P  112
$!                                                                             99994500  P  112
$!*****************************************************************************99999999  P  112


$#PATCH NUMBER 113 FOR ESPOL CONTAINS 2 CARDS.                                 00000000  P  113
 $SET OMIT LISTA = LIST                                                        00000999  P  113
 $POP OMIT LISTA                                                               00499999  P  113
$! DATE 3/2/76                                                                 99990000  P  113
$! BY JTC - MSA CENTRAL                                                        99990100  P  113
$!                                                                             99990200  P  113
$! THIS PATCH SETS OMIT AROUND THE COMMENT AT THE BEGINNING OF THE             99990300  P  113
$! ESPOL COMPILER TO SPEED UP SCANNING OF THE COMMENT.                         99990400  P  113
$!*****************************************************************************99999999  P  113


$#PATCH NUMBER 114 FOR ESPOL CONTAINS 222 CARDS.                               00000000  P  114

$ VOIDT 01476001                                                               01471000  P  114
              LABEL EXIT, NEXT, LASS;                                          15076000  P  114
              DEFINE                                                           15076100  P  114
                FORMALNAME = [9:2] = 2#,                                       15076110  P  114
                LONGID = NAMEID#,                                              15076120  P  114
                PARTIALWORD = (T1 ≠ 0)#,                                       15076130  P  114
                ERREXIT(NUM) = BEGIN ERR(NUM); GO TO EXIT; END#;               15076140  P  114
              BOOLEAN                                                          15076200  P  114
                SPCLMON,                                                       15076300  P  114
                DUPIT;  % TRUE IF WE ARE DOING UPDATE TYPE ASSIGN-             15076400  P  114
                        % MENT, I.E., I := * + 1..                             15076500  P  114
$ VOIDT 15127001                                                               15085150  P  114
              IF TALL.CLASS ≤ INTID THEN % SIMPLE VAR OR FORMAL NAME           15085200  P  114
                BEGIN                                                          15085400  P  114
                  IF STEP1 ≠ ASSIGNOP THEN                                     15085600  P  114
                    IF P1 = FL THEN                                            15085800  P  114
                      BEGIN                                                    15086000  P  114
```

```
      IF ELCLASS < AMPERSAND THEN                                    15086200  P  114
        EMITN(TALL.ADDRESS)                                          15086400  P  114
      ELSE                                                           15086600  P  114
        EMITV(TALL.ADDRESS);                                         15086800  P  114
      GO TO EXIT;                                                    15087000  P  114
    END                                                              15087200  P  114
  ELSE                                                               15087400  P  114
    IF ELCLASS = PERIOD THEN % PARTIAL WORD                          15087600  P  114
      IF DOTSYNTAX(T1,T2) THEN % ERROR                               15087800  P  114
        GO TO EXIT                                                   15088000  P  114
      ELSE                                                           15088200  P  114
        STEPIT;                                                      15088400  P  114
IF ELCLASS = ASSIGNOP THEN                                           15088600  P  114
  BEGIN                                                              15088800  P  114
    STACKCT != 1;                                                    15089000  P  114
    IF PARTIALWORD THEN % MAKE SURE LEFT-MOST                        15089200  P  114
      BEGIN                                                          15089400  P  114
        IF P1 # FS THEN % NOT LEFT-MOST                              15089600  P  114
          ERREXIT(201);                                              15089800  P  114
        XREFIT(TALL,REMEMBERSEQNO,ASSIGNREF);                        15090600  P  114
      END                                                            15090800  P  114
    ELSE                                                             15091000  P  114
      XMARK(ASSIGNREF);                                              15091200  P  114
    IF TABLE(I+1) = ASTRISK THEN % MIGHT BE UPDATE                   15091400  P  114
      IF (DUPIT!=(TABLE(I+2) ≥ EQVOP AND TABLE(I+2)                  15091600  P  114
          ≤ MULOP) OR TABLE(I+2) = AMPERSAND) THEN                   15091800  P  114
        STEPIT; % STEP OVER ASTERISK                                 15092000  P  114
    IF TALL.FORMALNAME THEN % FORMAL PARAMETER                       15092200  P  114
      BEGIN                                                          15092400  P  114
        EMITN(TALL.ADDRESS);                                         15092600  P  114
        IF PARTIALWORD OR DUPIT THEN % NEED VALUE                    15092800  P  114
          BEGIN                                                      15093000  P  114
            EMITO(DUP);                                              15093200  P  114
            EMITO(COC);                                              15093400  P  114
          END;                                                       15093600  P  114
      END                                                            15094800  P  114
    ELSE % ITS A SIMPLE VARIABLE                                     15095000  P  114
      IF PARTIALWORD OR DUPIT THEN                                   15095400  P  114
        EMITV(TALL.ADDRESS);                                         15095600  P  114
    IF PARTIALWORD AND DUPIT THEN                                    15095700  P  114
      BEGIN                                                          15095800  P  114
        EMITO(DUP);                                                  15095900  P  114
        EMITI(0,T1,T2);                                              15096000  P  114
      END;                                                           15096100  P  114
    STACKCT != REAL(PARTIALWORD OR DUPIT);                           15096400  P  114
    STEPIT;                                                          15096600  P  114
    IF DUPIT THEN % ALREADY GOT FIRST PRIMARY                        15096800  P  114
      SIMPARITH                                                      15097000  P  114
    ELSE                                                             15097200  P  114
      AEXP;                                                          15097400  P  114
    EMITD(48-T2,T1,T2);                                              15097600  P  114
    STACKCT != 0;                                                    15097800  P  114
    GT1 != IF TALL.CLASS = INTID THEN                                15098000  P  114
             IF P1 = FS THEN                                         15098200  P  114
               ISD                                                   15098400  P  114
             ELSE                                                    15098600  P  114
               ISN                                                   15098800  P  114
```

```
                        ELSE                                                15099000  P  114
                          IF P1 = FS THEN                                   15099200  P  114
                            STD                                             15099400  P  114
                          ELSE                                              15099600  P  114
                            SND;                                            15099800  P  114
                      IF TALL.FORMALNAME THEN                               15100000  P  114
                        BEGIN                                               15100200  P  114
                          EMITO(XCH); % TO GET DESCRIPTOR ON TOP            15100400  P  114
                          IF TALL.ADDRESS > 1023 THEN % SET VARIANT         15100600  P  114
                            EMITO(PRTE);                                    15100800  P  114
                          EMITO(GT1);                                       15101000  P  114
                        END                                                 15101200  P  114
                      ELSE                                                  15101400  P  114
                        EMITPAIR(TALL.ADDRESS,GT1);                         15101600  P  114
                    END                                                     15101800  P  114
                  ELSE % NOT ASSIGNMENT TO SIMPLE VARIABLE                  15102000  P  114
                    BEGIN                                                   15102200  P  114
                      IF P1 ≠ FP THEN % EXPECTED ASSIGNMENT                 15102400  P  114
                        ERREXIT(202);                                       15102600  P  114
                      EMITI(TALL,T1,T2); % EMIT OP CALL AND PARTIAL         15103400  P  114
                    END;                    % WORD CODE                    15103600  P  114
                END OF SIMPLE VARIABLES                                     15103800  P  114
$ VOIDT 15233028                                                            15183050  P  114
            IF STEP1 ≠ LFTBRKET THEN % ARRAY ITEM NOT FOLLOWED BY           15183100  P  114
              BEGIN                  % A SUBSCRIPT                          15183200  P  114
                IF ELCLASS = PERIOD THEN                                    15183300  P  114
                  IF DOTSYNTAX(T1,T2) THEN % ERROR IN PARTIAL WORD          15183400  P  114
                    GO TO EXIT                                              15183500  P  114
                  ELSE                                                      15183600  P  114
                    STEPIT;                                                 15183700  P  114
                ELSE                                                        15183600  P  114
                  STEPIT;                                                   15183700  P  114
              IF ELCLASS = ASSIGNOP THEN                                    15183800  P  114
                BEGIN                                                       15183900  P  114
                  IF PARTIALWORD THEN                                       15184000  P  114'
                    BEGIN                                                   15184100  P  114
                      IF P1 ≠ FS THEN % PARTIAL WORD NOT LEFT-MOST          15184200  P  114
                        ERREXIT(209);                                       15184300  P  114
                      XREFIT(TALL.REMEMBERSEQNO,ASSIGNREF);                 15184400  P  114
                    END                                                     15184500  P  114
                  ELSE % ASSIGNMENT TO ID WITH NO PARTIAL WORD              15184600  P  114
                    XMARK(ASSIGNREF);                                       15184700  P  114
                   IF TABLE(I+1) = ASTRISK THEN                             15184750  P  114
                    IF (DUPIT := (TABLE(I+2) ≥ EQVOP AND TABLE(I+2)         15184800  P  114
                        ≤ MULOP) OR TABLE(I+2) = AMPERSAND) THEN            15184850  P  114
                      STEPIT;                                               15185000  P  114
                  IF PARTIALWORD OR DUPIT THEN % NEED VALUE ON STACK        15185100  P  114
                    IF TALL.CLASS ≤ INTARRAYID THEN % NOT NAME ITEM         15185200  P  114
                      EMITPAIR(TALL.ADDRESS,LOD)                            15185300  P  114
                    ELSE                                                    15185400  P  114
                      EMITN(TALL.ADDRESS);                                  15185500  P  114
                    IF PARTIALWORD AND DUPIT THEN                           15185600  P  114
                      BEGIN                                                 15185700  P  114
                        EMITO(DUP);                                         15185800  P  114
                        EMITI(0,T1,T2);                                     15185900  P  114
                      END;                                                  15186000  P  114
                    STACKCT := STACKCT + REAL(PARTIALWORD OR DUPIT);        15186100  P  114
                    STEPIT;                                                 15186200  P  114
                    IF DUPIT THEN % WE HANDLED FIRST PRIMARY                15186300  P  114
                      SIMPARITH                                             15186400  P  114
```

```
                    ELSE                                                    15186500  P  114
                      AEXP;                                                 15186600  P  114
                    EMITD(48-T2,T1,T2);                                     15186700  P  114
                    EMITPAIR(TALL.ADDRESS,IF P1 = FS THEN STD ELSE          15186800  P  114
                                                          SND);             15186900  P  114
                    STACKCT := 0; % A AND B ARE EMPTY                       15187000  P  114
                  END                                                       15187100  P  114
                ELSE % NOT ASSIGNMENT                                       15187200  P  114
                  EMIT1(TALL,T1,T2);                                        15187300  P  114
                GO TO EXIT;                                                 15187400  P  114
              END OF ASSIGNMENT TO NON SIMPLE NON SUBSCRIPTED               15187500  P  114
                VARIABLE;                                                   15187600  P  114
$ VOIDT                                                                     15273000  P  114
$ VOIDT 15370001                                                            15299050  P  114
            IF STEP1 = PERIOD THEN % PARTIAL WORD                           15300000  P  114
              IF DOTSYNTAX(T1,T2) THEN % ERROR                              15300100  P  114
                GO TO EXIT                                                  15300200  P  114
              ELSE                                                          15300300  P  114
                STEPIT;                                                     15300400  P  114
            IF ELCLASS = ASSIGNOP THEN % ASSIGNMENT TO SUBSCRIPTED          15300500  P  114
              BEGIN                    % VARIABLE                           15300600  P  114
                IF PARTIALWORD THEN                                         15300700  P  114
                  IF P1 ≠ FS THEN                                           15300800  P  114
                    ERREXIT(209); % PARTIALWORD NOT LEFT-MOST               15300900  P  114
                XREFIT(TALL,REMEMBERSEQNO,ASSIGNREF);                       15301000  P  114
                IF J = 1 THEN % SINGLE-DIMENSIONED                          15301100  P  114
                  EMITN(TALL.ADDRESS)                                       15301200  P  114
                ELSE                                                        15301300  P  114
                  EMITO(CDC);                                               15301400  P  114
                IF TALL.CLASS ≥ LONGID THEN % EXPLICIT INDEX OP             15301500  P  114
                  EMITO(INX);                % REQUIRED                    15301600  P  114
            IF P1 = FR THEN % CALLED FROM FOR STATEMENT                     15301700  P  114
              GO TO EXIT;                                                   15301800  P  114
            IF TABLE(I+1) = ASTRISK THEN                                    15301900  P  114
              IF (DUPIT:=(TABLE(I+2) ≥ FQVOP AND TABLE(I+2)                 15302000  P  114
                 ≤ MULOP) OR TABLE(I+2) = AMPERSAND) THEN                   15302100  P  114
                STEPIT;                                                     15302200  P  114
            IF PARTIALWORD OR DUPIT THEN % NEED VALUE ON STACK              15302300  P  114
              BEGIN                                                         15302400  P  114
                EMITO(DUP);                                                 15302500  P  114
                EMITO(LOD);                                                 15302600  P  114
              END;                                                          15302700  P  114
            IF PARTIALWORD AND DUPIT THEN                                   15302800  P  114
              BEGIN                                                         15302900  P  114
                EMITO(DUP);                                                 15303000  P  114
                EMIT1(0,T1,T2);                                             15303100  P  114
              END;                                                          15303200  P  114
            STEPIT;                                                         15303300  P  114
            IF DUPIT THEN                                                   15303400  P  114
              SIMPARITH                                                     15303500  P  114
            ELSE                                                            15303600  P  114
              AEXP;                                                         15303700  P  114
            EMITD(48-T2,T1,T2);                                             15303800  P  114
            EMITO(XCH);                                                     15303900  P  114
            IF TALL.ADDRESS > 1023 THEN                                     15304000  P  114
              EMITO(PRTE);                                                  15304100  P  114
            EMITO(IF TALL.CLASS MOD 2 = INTARRAYID MOD 2 THEN               15304200  P  114
```

```
                IF P1 = FS THEN ISD ELSE ISN                          15304300  P  114
              ELSE IF P1 = FS THEN STD ELSE SND);                     15304400  P  114
         STACKCT := 0; % A & B ARE EMPTY                              15304500  P  114
         P1 := 0;                                                     15304600  P  114
       END OF ASSIGNMENT TO SUBSCRIPTED VARIABLE                      15304700  P  114
     ELSE                                                             15304800  P  114
       BEGIN % HANDLING OF NO ASSIGNMENT CASE                         15304900  P  114
         SPCLMON := P1 = FP OR PARTIALWORD OR ELCLASS ≥               15305000  P  114
                    AMPERSAND;                                        15305100  P  114
         IF J = 1 THEN % SINGLE DIMENSIONED                           15305200  P  114
           IF TALL.CLASS ≥ LONGID THEN % NAME ITEM                    15305300  P  114
             BEGIN                                                    15305400  P  114
               EMITN(TALL.ADDRESS);                                   15305500  P  114
               EMITO(INX);                                            15305600  P  114
               IF SPCLMON THEN                                        15305700  P  114
                 EMITO(LOD);                                          15305800  P  114
             END                                                      15305900  P  114
           ELSE % REFERENCE TO SINGLE DIMENSIONED ARRAY               15306000  P  114
             IF SPCLMON THEN                                          15306100  P  114
               EMITV(TALL.ADDRESS)                                    15306200  P  114
             ELSE                                                     15306300  P  114
               EMITN(TALL.ADDRESS)                                    15306400  P  114
         ELSE % MULTI-DIMENSIONED CASE                                15306500  P  114
           EMITO(IF SPCLMON THEN COC ELSE CDC);                       15306600  P  114
         IF P1 = FS THEN % EXPECTED AN ASSIGNMENT                     15306700  P  114
           ERREXIT(210);                                              15306800  P  114
         STACKCT := 1; % BECAUSE REGISTERS ARE NON-EMPTY              15306900  P  114
         IF PARTIALWORD THEN                                          15307000  P  114
           BEGIN                                                      15307100  P  114
             EMITI(0,T1,T2);                                          15307200  P  114
             P1 := 0;                                                 15307300  P  114
           END;                                                       15307400  P  114
       END OF NO ASSIGNMENT CASE;                                     15307500  P  114


$#PATCH NUMBER 115 FOR ESPOL CONTAINS 150 CARDS                       00000000  P  115

         CASEV           =49#,     COMMENT 061;                       01229000  P  115
       DEFINE LASTSEQUENCE = 149#,                                    01569000  P  115
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%07646010  P  115
%*********************************************************************%07646015  P  115
%*                                                                   *%07646020  P  115
%*                           C A S E                                 *%07646025  P  115
%*                           = = = =                                 *%07646030  P  115
%*                                                                   *%07646035  P  115
%*                      S T A T E M E N T                            *%07646040  P  115
%*                      = = = = = = = = =                            *%07646045  P  115
%*                                                                   *%07646050  P  115
%*                                                                   *%07646055  P  115
%* THIS PROCEDURE HANDLES THE CASE STATEMENT. THE SYNTAX FOR THE CASE *%07646060  P  115
%* STATMENT IS:                                                      *%07646065  P  115
%*                                                                   *%07646070  P  115
%*      CASE <AEXP> OF BEGIN <COMPOUND TAIL>                         *%07646075  P  115
%*                                                                   *%07646080  P  115
%* THE CODE GENERATED FOR THIS STATEMENT IS:                         *%07646085  P  115
%*                                                                   *%07646090  P  115
```

```
%*                          <AEXP>                                                  *%07646095  P  115
%*                          LITC  ?                                                 *%07646100  P  115
%*                          BFW         BRANCH TO BRANCH TABLE                      *%07646105  P  115
%*                          STMT 0                                                  *%07646110  P  115
%*                          LITC  ?                                                 *%07646115  P  115
%*                          BFW         BRANCH TO RESUME                            *%07646120  P  115
%*                          STMT 1                                                  *%07646125  P  115
%*                          LITC  ?                                                 *%07646130  P  115
%*                          BFW         BRANCH TO RESUME                            *%07646135  P  115
%*                          .                                                       *%07646140  P  115
%*                          .                                                       *%07646145  P  115
%*                          .                                                       *%07646150  P  115
%*      BRANCH TABLE:                                                               *%07646155  P  115
%*                          DUP                                                     *%07646160  P  115
%*                          ADD                                                     *%07646165  P  115
%*                          BFW         BRANCH INTO TABLE                           *%07646170  P  115
%*                          LITC  ?                                                 *%07646175  P  115
%*                          BBW         BRANCH TO STATEMENT 0                       *%07646180  P  115
%*                          LITC  ?                                                 *%07646185  P  115
%*                          BBW         BRANCH TO STATEMENT 1                       *%07646190  P  115
%*                          .                                                       *%07646195  P  115
%*                          .                                                       *%07646200  P  115
%*                          .                                                       *%07646205  P  115
%*                          LITC  ?                                                 *%07646210  P  115
%*                          BBW         BRANCH TO STATEMENT N                       *%07646215  P  115
%*      RESUME:                                                                     *%07646220  P  115
%*                                                                                  *%07646225  P  115
%*                                                                                  *%07646230  P  115
%* NOTICE THAT:                                                                     *%07646235  P  115
%*                                                                                  *%07646240  P  115
%*  1)  THE  CASE INDEX IS NOT INTEGERIZED. IF YOU HAVE ANYTHING THAT               *%07646245  P  115
%*      MIGHT CAUSE THE  INDEX  TO  BE  UNNORMALIZED  YOU WILL HAVE TO             *%07646250  P  115
%*      INTEGERIZE IT YOURSELF.                                                     *%07646255  P  115
%*                                                                                  *%07646260  P  115
%*  2) NO RANGE CHECKING IS DONE ON THE  INDEX.  IF  YOU USE A NUMBER               *%07646265  P  115
%*      OUT OF RANGE THEN YOU WILL FALL OFF THE END OF THE WORLD.                   *%07646270  P  115
%*                                                                                  *%07646275  P  115
%*  3) IF A STATEMENT IN THE CASE COMPOUND TAIL IS NULL,  NO  CODE IS               *%07646280  P  115
%*      GENERATED  FOR  THE  STATEMENT.  INSTEAD, THE BRANCH TABLE GOES             *%07646285  P  115
%*      DIRECTLY TO RESUME.                                                         *%07646290  P  115
%*                                                                                  *%07646295  P  115
%*  4) IF A STATEMENT  IN  THE  CASE COMPOUND TAIL IS A SIMPLE GO TO,               *%07646300  P  115
%*      THE STATEMENT DOES NOT HAVE A BRANCH TO RESUME AFTER IT.                    *%07646305  P  115
%*                                                                                  *%07646310  P  115
%*************************************************************************************%07646315  P  115
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%07646320  P  115
PROCEDURE CASESTATEMENT;                                                             07646325  P  115
BEGIN                                                                                07646330  P  115
  ARRAY CASEADDRESS[0:99];     % THIS ARRAY HOLDS THE RELATIVE                        07646335  P  115
                               % ADDRESS OF THE BEGINNING OF EACH                     07646340  P  115
                               % CASE.                                               07646345  P  115
  INTEGER                                                                            07646350  P  115
    LINK,                      % HOLDS RELATIVE ADDRESS OF FIRST OF                   07646355  P  115
                               % BRANCHES THAT MUST BE FIXED UP TO                   07646360  P  115
                               % BRANCH TO THE RESUME POINT.                         07646365  P  115
    ADR,                       % HOLDS RELATIVE ADDRESS OF THE BRANCH                 07646370  P  115
                               % AROUND THE CASE STATEMENTS TO THE                   07646375  P  115
```

```
                        % BRANCH TABLE. THIS BRANCH GETS FIXED      07646380  P  115
                        % UP WHEN WE FIND WHERE THE BRANCH TABLE    07646385  P  115
                        % GOES.                                    07646390  P  115
    N,                  % COUNT OF NUMBER OF CASE STATEMENT         07646395  P  115
                        % ENCOUNTERED.                             07646400  P  115
    ENDOFIT,            % ADDRESS OF RESUME POINT                   07646405  P  115
    J,                  % TEMPORARY                                 07646410  P  115
    K;                  % TEMPORARY                                 07646415  P  115
                        %                                          07646420  P  115
  BOOLEAN                                                          07646425  P  115
    GOTOG;              % TRUE IF WE ARE COMPILING A SIMPLE         07646430  P  115
                        % GO TO.                                    07646435  P  115
  LABEL XIT;                                                       07646436  P  115
%                                                                  07646440  P  115
%                                                                  07646445  P  115
    STEPIT;    % STEP OVER "CASE"                                  07646450  P  115
    AEXP;      % GENERATE CODE FOR CASE INDEX                      07646455  P  115
    IF STEPI ≠ BEGINV THEN % NOTICE WE JUST JUMPED OVER "OF"       07646456  P  115
      BEGIN                                                        07646457  P  115
        ERR(70); GO TO XIT;                                        07646458  P  115
      END;                                                         07646459  P  115
    EMIT(0);  % GENERATE DUMMY BRANCH TO BRANCH TABLE. WILL FIX    07646460  P  115
    EMITO(BFW);  %     IT UP LATER.                                07646465  P  115
    ADR := L;                                                      07646470  P  115
    WHILE STEPI ≠ ENDV DO                                          07646475  P  115
      BEGIN   % PROCESSING CASE STATEMENTS                         07646480  P  115
        ERRORTOG := TRUE;                                          07646485  P  115
        IF ELCLASS = SEMICOLON THEN % NULL STATEMENT, NO CODE NEEDED 07646490  P  115
          N := N + 1  % THIS LEAVES A ZERO IN CASEADDRESS[N]        07646495  P  115
        ELSE                                                       07646500  P  115
          BEGIN                                                    07646505  P  115
            CASEADDRESS[N] := L; % REMEMBER BEGINNING ADDRESS OF   07646510  P  115
            N := N + 1;          % THIS CASE.                      07646515  P  115
            IF (GOTOG := SIMPGO) THEN                              07646520  P  115
              ELBAT[I:=I-1] := ELCLASS := GOV; % REMEMBER IF SIMPLE 07646525  P  115
            STMT; % PROCESS THE STATEMENT                          07646530  P  115
            IF ELCLASS = SEMICOLON THEN                            07646535  P  115
              IF NOT GOTOG THEN % GENERATE DUMMY BRANCH TO RESUME  07646540  P  115
                BEGIN                                              07646545  P  115
                  EMIT(LINK);                                      07646550  P  115
                  EMITO(BFW);                                      07646555  P  115
                  LINK := L;                                       07646560  P  115
                END                                                07646565  P  115
              ELSE % SIMPLE GO TO, NO CODE TO GENERATE             07646570  P  115
            ELSE                                                   07646575  P  115
              IF ELCLASS ≠ ENDV THEN                               07646580  P  115
                ERR(71);                                           07646585  P  115
          END;                                                     07646590  P  115
      END OF WHILE LOOP;                                           07646595  P  115
    ENDTOG := TRUE;     % SKIP OVER COMMENT AFTER END               07646601  P  115
    DO STOPDEFINE := TRUE UNTIL                                    07646602  P  115
      STEPI ≤ ENDV AND ELCLASS ≥ UNTILV OR NOT ENDTOG;             07646603  P  115
    ENDTOG := FALSE;                                               07646604  P  115
    EMITB(BFW,ADR,L);     % FIX UP BRANCH TO BRANCH TABLE          07646605  P  115
    EMITO(DUP);           % GENERATE CODE TO MULTIPLY INDEX        07646610  P  115
    EMITO(ADD);           % BY TWO AND BRANCH INTO BRANCH TABLE    07646615  P  115
    EMITO(BFW);                                                    07646620  P  115
```

```
    ENDOFIT := L + 2*N; % CALCULATE WHERE RESUME IS                          07646625  P  115
    WHILE (J:=J+1) ≤ N DO  % GENERATE THE BRANCH TABLE                       07646630  P  115
      EMITB(BBW,L:=L+2,IF (K:=CASEADDRESS[J-1]) = 0 THEN                     07646635  P  115
                          ENDOFIT ELSE K);                                   07646640  P  115
    J := LINK; % TO MAKE THE LOOP WORK                                       07646645  P  115
    WHILE (LINK:=J) ≠ 0 DO                                                   07646650  P  115
      BEGIN  % FIXING UP BRANCHES TO RESUME                                  07646655  P  115
        J := GET(LINK-2);  % LOCATION OF NEXT BRANCH TO FIX                  07646660  P  115
        EMITB(BFW,LINK,L);                                                   07646665  P  115
      END;                                                                   07646670  P  115
XIT:                                                                         07646674  P  115
END OF CASE STATEMENT;                                                       07646675  P  115
    L49:                                                                     07759100  P  115
          CASESTATEMENT; GO TO EXIT;                                         07759200  P  115
 L41:                                                                        07770000  P  115
    OCT0500000250000002, "5FIELD",                          %654             09128700  P  115
    OCT0610000080000000, "4CASE0",                          %656             09128800  P  115
$: BY JTC - MSC DETROIT                                                      99990000  P  115
$: THIS PATCH ADDS A -CASE- STATEMENT TO ESPOL. SEE                          99990100  P  115
$: DOCUMENTATION AT BEGINNING OF PROCEDURE TO SEE HOW                        99990200  P  115
$: IT IS USED.                                                               99990300  P  115
$:*************************************************************************99990400  P  115


$#PATCH NUMBER 116 FOR ESPOL CONTAINS 1 CARD                                 00000000  P  116
 SAVE ARRAY DEFINEARRAY[0:34];                                               01491000  P  116
$: DATE 6/29/76                                                              99990000  P  116
$: BY JTC - MSA CENTRAL                                                      99990100  P  116
$:                                                                           99990200  P  116
$: THIS PATCH FIXES A PROBLEM WITH ESPOL THAT COULD CAUSE THE                99990300  P  116
$: COMPILER TO ABORT WITH AN INVALID INDEX AT END OF FILE ON                 99990400  P  116
$: EITHER THE CARD OR TAPE FILE IF NO 99999999 CARD WAS PROVIDED.            99990500  P  116


$#PATCH NUMBER 117 FOR ESPOL CONTAINS 3 CARDS.                               00000000  P  117

%                       MARK XV1.0.116                                       00001030  P  117
%                        MAY  9, 1977                                        00001040  P  117
"XV1.0.116"                                                                  01831000  P  117
$: BY DJZ - MSC DETROIT                                                      99990000  P  117
$: DATE 05/09/77                                                             99990100  P  117
$: THIS PATCH UPDATES THE MARK LEVELS OF THE SYMBOL FILE.                    99990200  P  117
$:*************************************************************************99990300  P  117


$# PATCH NUMBER 901 FOR ESPOL CONTAINS 1 CARD                                C  901 0000000/PMCARD

$: THIS PATCH CHANGES INPUT TAPE BLOCKING TO ALLOW INCREASED                 C  901
$: EFFICIENCY IN STORING SOURCE FILES ON TAPE                                C  901
$: BY TJP. 2/16/77.  USE UTILITY COMPTP/UTIL TO CREATE SOURCE TAPES.         C  901
```

```
BEGIN RR8←50×10←50←1; RR9←10 END;                                  00538000  C  901

$#PATCH NUMBER 902 FOR FSPOL CONTAINS 1 CARD.   PAGE SKIP BETWEEN PROCEDURES      C  902
$; THIS PATCH CHANGES THE EFFECT OF THE FORMAT OPTION TO CAUSE A                   C  902
$; PAGE SKIP BETWEEN PROCEDURES INSTEAD OF A FOUR-LINE SKIP                       C  902
$;  BY J.H., UCSC, 4 MAY 77                                                       C  902
$;                                                                             C  902
      SPACEITDOWN = WRITE(LINE,PAGE))#;                             14023100  C  902
```

```
**********    CONFLICTS    ************************************************************************************************

$ VOIDT 01001771                                                                01001750 P  110 CONFLICTED WITH:
DEFINE XREFINFO[XREFINFO1,XREFINFO2]=                                           01001750 P  108    DISCARDED
          XINFO[XREFINFO1,(XREFINFO2)DIV 2]#,                                   01001760 P  108    VOIDED
       XMARK= IF XREF THEN XREFAY2[XREFPT-1],[1:1]+1#;                          01001770 P  108    VOIDED


                        %                                                       01007100 P  110 CONFLICTED WITH:
     ARRAY XREFAY2[0:29],XREFAY1[0:10],XINFO[0:31,0:127];                       01007100 P  108    DISCARDED


                        %                                                       01007200 P  110 CONFLICTED WITH:
INTEGER XREFPT,XLUN;                                                           %DFB01007200 P  108    DISCARDED


        DEFINE LASTSEQUENCE = 149#,                                             01569000 P  115 CONFLICTED WITH:
DEFINE LASTSEQUENCE = 147#,                                                     01569000 P  112    DISCARDED


DEFINE LASTSEQUENCE = 147#,                                                     01569000 P  110 CONFLICTED WITH:
         IF XREF THEN                                                           02882100 P  108    VOIDED
         IF GT1#1 THEN                                                          02882200 P  108    VOIDED
         BEGIN                                                                  02882300 P  108    VOIDED
              IF XREFPT=30 THEN                                                 02882400 P  108    VOIDED
              BEGIN                                                             02882500 P  108    VOIDED
                   WRITE (DSK2,30,XREFAY2[*]);                                  02882600 P  108    VOIDED
                   XREFPT ← 0;                                                  02882700 P  108    VOIDED
              END;                                                              02882800 P  108    VOIDED
              XREFAY2[XREFPT]←CARDNUMBER&XREFINFO[GT1,GT2][09:36:12];           02882900 P  108    VOIDED
              XREFPT ← XREFPT+1;                                                02882910 P  108    VOIDED
         END;                                                                   02882920 P  108    VOIDED


    OCT0500000250000002, "5FIELD",                  %654        09128700 P  115 CONFLICTED WITH:
    OCT0500000250000000, "5FIELD",                  %654        09128700 P  112    DISCARDED


    OCT0500000250000000, "5FIELD",                  %654        09128700 P  110 CONFLICTED WITH:
STREAM PROCEDURE XREFMOVE(S,C,SN,SQ,L,D);                      %DFB13301100 P  108    VOIDED
    VALUE C,L;                                                 %DFB13301130 P  108    VOIDED
    BEGIN                                                      %DFB13301160 P  108    VOIDED
         DI:=D;                                                %DFB13301190 P  108    VOIDED
         DS:= 8 LIT" ";                                        %DFB13301220 P  108    VOIDED
         SI:=D;                                                %DFB13301250 P  108    VOIDED
         DS:=8 WDS;                                            %DFB13301280 P  108    VOIDED
         DI:=D;                                                %DFB13301310 P  108    VOIDED
         SI:=S;                                                %DFB13301340 P  108    VOIDED
         SI:=SI+3;                                             %DFB13301370 P  108    VOIDED
         DS:=C CHR;                                            %DFB13301400 P  108    VOIDED
         DI:=D;                                                %DFB13301430 P  108    VOIDED
          DI←DI+60;                                            %DFB13301460 P  108    VOIDED
         SI:=SN;                                               %DFB13301490 P  108    VOIDED
```

```
        DS:=4 DEC;                                                    %DFB13301520 P  108    VOIDED
        S1:=SQ;                                                       %DFB13301610 P  108    VOIDED
        DS:=8 DEC;                                                    %DFB13301640 P  108    VOIDED
        S1:=LOC L;                                                    %DFB13301670 P  108    VOIDED
        DS:=WDS;                                                      %DFB13301700 P  108    VOIDED
    END;                                                              %DFB13301730 P  108    VOIDED


    END;                                                              %DFB13301730 P  110 CONFLICTED WITH:
        IF XREF AND (NOT PTOG OR STREAMTOG) THEN                        13310100 P  108    VOIDED
    BEGIN                                                             %DFB13310200 P  108    VOIDED
    XREFMOVE(ACCUM[1],COUNT,SGNO ,CARDNUMBER,                         %DFB13310300 P  108    VOIDED
             XREFINFO[NEXTINFO.LINKR,NEXTINFO.LINKC]:=XLUN:=XLUN+1,   %DFB13310400 P  108    VOIDED
               XREFAY1);                                              13310450 P  108    VOIDED
    WRITE(DSK1,10,XREFAY1[*]);                                        %DFB13310500 P  108    VOIDED
    END;                                                              %DFB13310600 P  108    VOIDED

          IF XREF AND NOT SPECTOG THEN % ERASE PREVIOUS XREF ENTRY.   13341200 P  110 CONFLICTED WITH:
          IF XREF AND PTOG THEN                                       13341200 P  108    DISCARDED


          IF XREF AND PTOG THEN                                       13341200 P  114 CONFLICTED WITH:
      XMARK(ASSIGNREF); % ASSIGNMENT TO SIMPLE VARIABLE.              15091100 P  110    VOIDED
$VOIDT                                                                15092010 P  110    VOIDED
                        BEGIN                                         15113100 P  110    VOIDED
                          BEGIN                                       15115000 P  110    VOIDED
                            ERR(201); % PARTIAL WORD NOT LEFT-MOST    15115100 P  110    VOIDED
                            GO TO EXIT;                               15115200 P  110    VOIDED
                          END;                                        15115300 P  110    VOIDED
                          XREFIT(TALL,REMEMBERSEQNO,ASSIGNREF);       15116000 P  110    VOIDED
                          GO TO L1;                                   15116100 P  110    VOIDED
                        END;                                          15116200 P  110    VOIDED
L1;                                                                   15092000 P  108    VOIDED
       XMARK;                                                         15092010 P  108    VOIDED
       IF  TALL.FORMALNAME THEN                                       15092020 P  108    VOIDED


       IF  TALL.FORMALNAME THEN                                       15092020 P  114 CONFLICTED WITH:
    JAZZ:          XREFIT(TALL,REMEMBERSEQNO,ASSIGNREF); STEPIT; AEXP; 15233012 P  110    VOIDED


    JAZZ:          XREFIT(TALL,REMEMBERSEQNO,ASSIGNREF); STEPIT; AEXP; 15233012 P  114 CONFLICTED WITH:
                   XREFIT(TALL,REMEMBERSEQNO,ASSIGNREF); % ASSIGNMENT TO15301100 P  110    VOIDED
                                % SUBSCRIPTED VARIABLE.                15301200 P  110    VOIDED
                   IF STEPI = ASSIGNOP THEN                           15342000 P  110    VOIDED
                     IF P1 = FS THEN % PARTIAL WORD IS LEFT-MOST      15342100 P  110    VOIDED
                       BEGIN                                          15342200 P  110    VOIDED
                         XREFIT(TALL,REMEMBERSEQNO,ASSIGNREF); % PARTIAL15342300 P  110    VOIDED
                                % WORD ASSIGNMENT TO SUBSCR. VAR.     15342400 P  110    VOIDED
                         GO TO LAST;                                  15342500 P  110    VOIDED
                       END                                            15342600 P  110    VOIDED


IF (XREF OR DEFINING.[1:1]) AND XLUN > 0 THEN                         16495300 P  110 CONFLICTED WITH:
IF XREF OR  DEFINING.[1:1] THEN                                       16495300 P  108    DISCARDED
```

```
DEFINE XREFINFO[INDEX] = INFO[((INDEX).CF DIV 2),[33:7],                17002005 P  110 CONFLICTED WITH:
        DEFINE XREFINFO=INFO#;                                          17002005 P  108    DISCARDED

TOTALNO := XLUN; % REMEMBER NUMBER OF IDENTIFIERS.                      17004500 P  110 CONFLICTED WITH:
             TOTALNO←REAL(XLUN >500)×1000+3000+XLUN;                    17004500 P  108    DISCARDED


        BEGIN                                                           17022000 P  110 CONFLICTED WITH:
        REWIND(DSK1)                                                  %DFB17022000 P  108    DISCARDED

        IF BOOLEAN(A[8]) THEN                                           17025000 P  110 CONFLICTED WITH:
        A[9]:=XREFINFO[A[9].LINKR,A[9].LINKC]:=XLUN:=XLUN+1;%DFB17025000 P  108    DISCARDED


    IF 63 SC < DC THEN                                                  17033000 P  110 CONFLICTED WITH:
    IF 63 SC LSS DC THEN TALLY:=1;                                    %DFB17033000 P  108    DISCARDED

        IF REAL(COMP1:=COMPS1(A,B)) = 2 THEN % IDS EQUAL                17042300 P  110 CONFLICTED WITH:
        COMP1:=COMPS1(A,B);                                           %DFB17042300 P  108    DISCARDED


SORT(OUTPUT1,INPUT1,0,HV1,COMP1,10,IF TOTALNO < 1000 THEN               17045000 P  110 CONFLICTED WITH:
SORT(OUTPUT1,INPUT1,0,HV1,COMP1,10,TOTALNO    );                      %DFB17045000 P  110 CONFLICTED WITH:
        SORT(OUTPUT1,INPUT1,0,HV1,COMP1,10,TOTALNO    );                %DFB17045000 P  108    VOIDED
        STREAM PROCEDURE PUP(S,D);                                      %DFB17048000 P  108    VOIDED
          BEGIN                                                         %DFB17049000 P  108    VOIDED
          SI:=S;                                                        %DFB17050000 P  108    VOIDED
          DI:=D;                                                        %DFB17051000 P  108    VOIDED
        DS:=7 WDS; 8(DS:=8 LIT " ");                                      17052000 P  108    VOIDED
$VOIDT 17054551                                                           17053000 P  108    VOIDED
          END;                                                          %DFB17055000 P  108    VOIDED
        STREAM PROCEDURE PUP1(S,D);                                       17055100 P  108    VOIDED
        BEGIN                                                             17055200 P  108    VOIDED
        SI:=S; SI:=SI+32; SI:=SI+28;                                      17055300 P  108    VOIDED
        DI:=D; DS:=2 LIT " "; DS:=4 CHR; DS:=2 LIT " *";                  17055400 P  108    VOIDED
        DS:=8 CHR; DS:=LIT "*"; DS:=38 LIT " ";                           17055500 P  108    VOIDED
        END;                                                              17055550 P  108    VOIDED
         STREAM PROCEDURE PUP2(STAR,S,C,D);                               17056000 P  108    VOIDED
         VALUE STAR,S,C;                                                  17056500 P  108    VOIDED
          BEGIN                                                         %DFB17057000 P  108    VOIDED
          DI:=D;                                                        %DFB17058000 P  108    VOIDED
        C(DI:=DI+10);                                                     17059000 P  108    VOIDED
                                 STAR(DI←DI"1; DS←LIT"*");                17059100 P  108    VOIDED
          SI:=LOC S;                                                    %DFB17060000 P  108    VOIDED
          DS:= 8 DEC;                                                   %DFB17061000 P  108    VOIDED
        DS:=LIT " ";          STAR(DI:=DI"1; DS:=LIT "*");                17061100 P  108    VOIDED
          END;                                                          %DFB17062000 P  108    VOIDED
        STREAM PROCEDURE BLANKET(A);                                    %DFB17063000 P  108    VOIDED
             BEGIN                                                      %DFB17064000 P  108    VOIDED
            DI:=A;                                                      %DFB17065000 P  108    VOIDED
             DS:= 8 LIT " ";                                            %DFB17066000 P  108    VOIDED
        S:=A;                                                           %DFB17067000 P  108    VOIDED
```

```
    DS:= 14 WDS;                                                                %DFB17068000 P  108     VOIDED
    END;                                                                        %DFB17069000 P  108     VOIDED


ARRAY !DTYPE[0:(IDMAX+4)*4-1];                                                      17047100 P  112 CONFLICTED WITH:
ARRAY !DTYPE[0:(IDMAX+3)*4-1];                                                      17047100 P  110     DISCARDED


      DEFINE I = LASTADDRESS#;                                                      17073100 P  110 CONFLICTED WITH:
       DEFINE I=XLUN#;                                                          %DFB17073100 P  108     DISCARDED


    A[0] := I & NEWID[I,REFIDNOF] REFIDNOF;                                         17080000 P  110 CONFLICTED WITH:
    A[0]:= !&XREFINFO[I,[ 9:4],I,[13:8]][ 9:36:12];                             %DFB17080000 P  108     DISCARDED


    XREFAY1[8] := XREFPT := LASTADDRESS := 0;                                       17084000 P  110 CONFLICTED WITH:
    XREFAY1[9]:=-1;       XREFPT+0;                                             %DFB17084000 P  108     DISCARDED


      "LABEL.                          ", % 32                                      17084300 P  112 CONFLICTED WITH:
      "LABEL.                          "; % 32                                      17084300 P  110     DISCARDED


    LABEL EOF2, SKIP;                                                               17091100 P  110 CONFLICTED WITH:
         LABEL LOOP,EOF2;                                                           17091100 P  108     DISCARDED


    OWN BOOLEAN B2, FWDTOG, LBLTOG, WAITINGFORFWDREF;                               17091110 P  110 CONFLICTED WITH:
    OWN BOOLEAN B2;                                                                 17091110 P  108     DISCARDED


      END;                                                                          17091200 P  110 CONFLICTED WITH:
      LABEL DUN;                                                                    17091200 P  108     DISCARDED


      LABEL DUN;                                                                    17091200 P  110 CONFLICTED WITH:
     IF NOT B2 THEN                                                                 17091210 P  108     VOIDED
       IF B THEN                                                            %DFB17092000 P  108     VOIDED
          WRITE(PRINTER,15,PAY[*])                                          %DFB17093000 P  108     VOIDED
       ELSE                                                                 %DFB17094000 P  108     VOIDED
IF LASTADDRESS # LASTADDRESS+A[0] AND LASTADDRESS.[9 :12]# 0 THEN               17094100 P  108     VOIDED
          BEGIN                                                             %DFB17095000 P  108     VOIDED
 LOOP:                                                                      %DFB17095050 P  108     VOIDED
          IF  A[0].[ 9:12] GTR XREFAY1[9] THEN                              %DFB17095100 P  108     VOIDED
          BEGIN                                                             %DFB17095200 P  108     VOIDED
$                                                                               17096000 P  108     VOIDED
      WRITE(PRINTER,15,PAY[*]);                                                 17097000 P  108     VOIDED
$                                                                               17098000 P  108     VOIDED
          BLANKET(PAY ); XREFPT:=0;                                             17099000 P  108     VOIDED
          READ(DSK1,10,XREFAY1[*])[EOF2];                                       17099100 P  108     VOIDED
          PUP(XREFAY1,PAY);                                                     17099200 P  108     VOIDED
          WRITE(PRINTER[DBL]);                                                  17099900 P  108     VOIDED
          WRITE(PRINTER,10,PAY[*]);                                         %DFB17100000 P  108     VOIDED
     PUP1(XREFAY1,PAY);                                                         17100100 P  108     VOIDED
          WRITE(PRINTER[DBL],10,PAY[*]);                                        17100200 P  108     VOIDED
          BLANKET(PAY);                                                     %DFB17101000 P  108     VOIDED
       GO LOOP;                                                             %DFB17101100 P  108     VOIDED
```

```
            END;                                                                %DFB17102000 P  108     VOIDED
DUN:                                                                                17102100 P  108     VOIDED
                        PUP2(LASTADDRESS < 0,LASTADDRESS,[21:27],                   17103000 P  108     VOIDED
                             XREFPT,PAY[1]);                                        17103010 P  108     VOIDED
            IF XREFPT:=XREFPT+1 = 11 THEN                                           17103100 P  108     VOIDED
              BEGIN                                                                 %DFB17103200 P  108     VOIDED
                XREFPT:=0;                                                          %DFB17103300 P  108     VOIDED
                WRITE(PRINTER,15,PAY[*]);                                           %DFB17103400 P  108     VOIDED
                BLANKET(PAY);                                                       %DFB17103500 P  108     VOIDED
              END;                                                                  %DFB17103600 P  108     VOIDED
            END;                                                                    %DFB17110000 P  108     VOIDED
          IF FALSE THEN EOF2: B2:=TRUE;                                             17110500 P  108     VOIDED
            END;                                                                    %DFB17111000 P  108     VOIDED


                        XREFAY1[9].CLASS) > IDMAX THEN IF I =                       17095000 P  112 CONFLICTED WITH:
                        XREFAY1[9].CLASS) > IDMAX THEN                              17095000 P  110     DISCARDED


                        FIELDID THEN 33 ELSE 0 ELSE I) * 4],                        17095100 P  112 CONFLICTED WITH:
                        0 ELSE I) * 4],                                             17095100 P  110     DISCARDED


          A[0] := 3"777777777777777"; % BIGGEST FLOATING PT. NO.                    17114000 P  110 CONFLICTED WITH:
          A[0]:=5497558138871                                                       %DFB17114000 P  108     DISCARDED


           COMP2 := IF A[0].REFIDNOF < B[0].REFIDNOF THEN % DIF IDS                 17117000 P  110 CONFLICTED WITH:
           COMP2:= ABS(A[0]) LEQ ABS(B[0]);                                          17117000 P  108     DISCARDED


          SORT(OUTPUT2,INPUT2,0,HV2,COMP2,1,6000);                                  17119000 P  110 CONFLICTED WITH:
          SORT(OUTPUT2,INPUT2,0,HV2,COMP2,1,TOTALNO    );                           %DFB17119000 P  108     DISCARDED
```

```
NUMBER OF ERRORS DETECTED =  0.
PROCESSOR TIME =   65 SECONDS.
I/O TIME =   40 SECONDS.

 LABEL  000000000LINE   001772347 EXECUTE PATCH/MERGE                                  PATCH  /MERGE
```

```
? COMPILE ESPOL/DISK ALGOL LIBRARY

          PACKET     7
          INPUT   1143 CARDS FROM ZIP
          TIME    1601
          DATE   77234 MONDAY, 08/22/77



       *** BURROUGHS B5700 DCMCP MARK XVI.0.73  AND INTRINSICS MARK XVI.0.00 ***

           #NO MESSAGES TODAY

        ? COMPILE ESPOL/DISK ALGOL LIBRARY
            ? ALGOL STACK= 1000
            ? ALGOL FILE TAPE= SYMBOL/ESPOL DISK SERIAL
            ? FILE LINE= LINE PRINT OR BACK UP
        ? DATA CARD
                4:ALGOL/ESPOL=01 SCHEDULED 1601, NEEDS 27328
                4:ALGOL/ESPOL= 1 BOJ 1602 05/12/77
                CDB IN CARD DB:ALGOL/ESPOL= 1
                DKA IN SER SYMBOL ESPOL:ALGOL/ESPOL= 1
                PBD0008 OUT 011 LINE:ALGOL/ESPOL= 1
                DKA OUT SER DSK1 U000000:ALGOL/ESPOL= 1
                DKA OUT SER DSK2 U000000:ALGOL/ESPOL= 1
                DKA OUT RDM ESPOL DISK:ALGOL/ESPOL= 1
               #DUP LIBRARY ESPOL/DISK:ALGOL ESPOL= 1
                ESPOL/DISK REMOVED
                DKA LOK ESPOL DISK:ALGOL/ESPOL= 1
                CDB REL CARD DB:ALGOL/ESPOL= 1
       ? END.
                DKA REL SYMBOL ESPOL:ALGOL/ESPOL= 1
                DKA OUT SER DSRT1 U000000:ALGOL/ESPOL= 1
                DKA OUT SER DSRT2 U000000:ALGOL/ESPOL= 1
                DKA REL DSRT1 U000000:ALGOL/ESPOL= 1
                DKA REL DSRT2 U000000:ALGOL/ESPOL= 1
                DKA OUT SER DSRT1 U000000:ALGOL/ESPOL= 1
                DKA OUT SER DSRT2 U000000:ALGOL/ESPOL= 1
                PBD0008 OUT 012:ALGOL/ESPOL= 1
                DKA REL DSRT1 U000000:ALGOL/ESPOL= 1
                DKA REL DSRT2 U000000:ALGOL/ESPOL= 1
                DKA REL DSK2 U000000:ALGOL/ESPOL= 1
                DKA REL DSK1 U000000:ALGOL/ESPOL= 1
                PBD0008 REL 012 LINE 12687:ALGOL/ESPOL= 1
                ALGOL/ESPOL= 1 EOJ 1613
                FOR ALGOL/ESPOL= 1: PROCESS= 474 SECS, IO= 269 SECS, OLAY= 9
                PKT#0007 REMOVED
```

FSPOL /DISK
==========

SOURCE FILE: SYMBOL /ESPOL

```
                $SET OMIT LISTA = LIST                                          %113-     00000999 C  0001:000:0
               BEGIN COMMENT OUTERMOST BLOCK;                                             00500000 T  0001:000:0
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  0002:000:0
PRT(25) = ERRORCOUNT
                                    SYNTX ERR AT EOJ IF THIS IS NON-ZERO, MUST BE @R+25;    00502000 T  0002:000:0
                INTEGER SAVETIME; COMMENT SAVE-FACTOR FOR CODE FILE, GIVEN BY MCP.          00503000 T  0002:000:0
PRT(26) = SAVETIME
                                    IF COMPILE & GO =0, FOR SYNTAX, =-1. MUST BE AT R+26;   00504000 T  0002:000:0
               INTEGER CARDNUMBER; % SEQ # OF CARD BEING PROCESSED.                         00504100 T  0002:000:0
PRT(27) = CARDNUMBER
               INTEGER CARDCOUNT;    % NUMBER OF CARDS PROCESSED.                        00504150 T  0002:000:0
PRT(30) = CARDCOUNT
                BOOLEAN BUILDLINE;                                                      00504700 T  0002:000:0
PRT(31) = BUILDLINE
                   COMMENT RR1-RR11 ARE USED IN SOME PROCEDURES IN                      00505000 T  0002:000:0
                   PLACE OF LOCALS TO SAVE STACK SPACE;                                 00506000 T  0002:000:0
               REAL RR1,RR2,RR3,RR4,RR5,RR6,RR7,RR8,RR9,RR10,RR11;                      00507000 T  0002:000:0
PRT(32) = RR1
PRT(33) = RR2
PRT(34) = RR3
PRT(35) = RR4
PRT(36) = RR5
PRT(37) = RR6
PRT(40) = RR7
PRT(41) = RR8
PRT(42) = RR9
PRT(43) = RR10
PRT(44) = RR11
                   COMMENT SOME OF THE RR1 ARE USED TO PASS FILE INFORMATION            00508000 T  0002:000:0
                       TO THE MAIN BLOCK;                                               00509000 T  0002:000:0
                   COMMENT EXAMIN RETURNS THE CHARACTER AT ABSOLUTE ADDRESS NCR;        00510000 T  0002:000:0
               REAL STREAM PROCEDURE EXAMIN(NCR); VALUE NCR;                            00511000 T  0002:000:0
PRT(45) = EXAMIN
                        BEGIN SI←NCR;DI←LOC EXAMIN;DI←DI+7; DS←CHR END;                 00512000 T  0002:000:0


               INTEGER STREAM PROCEDURE GETF(Q);VALUE Q;                                00523000 T  0002:002:0
PRT(46) = GETF
```

```
                    BEGIN SI←LOC GETF; SI←SI-7;DI←LOC Q;DI←DI+5;                                    00524000 T  0002↑0002↑0
                         SKIP 3 DB; 9(IF SB THEN DS←SET ELSE DS←RESET; SKIP SB);                    00525000 T  0002↑0003↑2
                         DI←LOC Q;SI←Q;DS←WDS;SI←Q;GETF←SI                                          00526000 T  0002↑0005↑2
                    END GETF;                                                                       00527000 T  0002↑0006↑1



               COMMENT START SETTING UP FILE PARAMETERS;                                            00528000 T  0002↑0007↑3
                         IF EXAMIN(RR11←GETF(3)+"Y08") ≠12 THEN RR1←5 ELSE                          00529000 T  0002↑0007↑3
                         BEGIN RR1←2;RR2←150 END;                                                   00530000 T  0002↑0013↑2
                         IF EXAMIN(RR11+5) ≠12 THEN RR3←4 ELSE                                      00531000 T  0002↑0016↑1
                         BEGIN RR3←2; RR4←150 END;                                                  00532000 T  0002↑0020↑0
                         IF EXAMIN(RR11+10)=12 THEN                                                 00533000 T  0002↑0022↑0
                              BEGIN RR5←2;RR6←10;RR7←150 END ELSE                                   00534000 T  0002↑0024↑0
                         BEGIN RR5←1;RR6←56;RR7←10 END;                                             00535000 T  0002↑0027↑2
                         IF EXAMIN(RR11+15)=12 THEN                                                 00536000 T  0002↑0029↑3
                              BEGIN RR8←10;RR9←150 END ELSE                                         00537000 T  0002↑0032↑0
               BEGIN RR8←50×10←50+1; RR9←10 END;                                    %901←          00538000 P  0002↑0034↑0
               BEGIN COMMENT MAIN BLOCK;                                                            01000000 T  0002↑0037↑3
               INTEGER OPINX; % USED FOR INDEXING INTO OPTIONS ARRAY.                               01000800 T  0002↑0037↑3
PRT(47) = *SEGMENT DESCRIPTOR*

                                                                          START OF SEGMENT **********    3
PRT(50) = OPINX
               BOOLEAN SETTING; % USED BY DOLLARCARD FOR AN OPTION"S SETTING.                        01000802 T  0003↑0000↑0
PRT(51) = SETTING
                INTEGER NEWINX, ADDVALUE, BASENUM, TOTALNO;                                         01000860 T  0003↑0000↑0
PRT(52) = NEWINX
PRT(53) = ADDVALUE
PRT(54) = BASENUM
PRT(55) = TOTALNO
               DEFINE OPARSIZE = 200 #;                                                             01000902 T  0003↑0000↑0
               ARRAY OPTIONS[0:OPARSIZE];                                                           01000904 T  0003↑0000↑0
PRT(56) = OPTIONS
               BOOLEAN OPTIONWORD;                                                                  01000910 T  0003↑0002↑1
PRT(57) = 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#,                                          %108-   01001170 P  0003:0002:1
                BENDBIT         = 27#,                                          %108-   01001171 C  0003:0002:1
                USEROPINX       = 28#;                                          %108-   01001172 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] #,             %108-           01001461 C  0003:0002:1
                BEND            = OPTIONWORD.[BENDBIT:1]#,              %108-           01001462 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(60) = NOHEADING
        BOOLEAN NEWBASE;        % NEW BASENUM FOUND ON A NEW $-CARD.                    01001490 T  0003:0002:1
PRT(61) = NEWBASE
        BOOLEAN LASTCRDPATCH;   % NORMALLY FALSE, SET TO TRUE WHEN THE                  01001500 T  0003:0002:1
PRT(62) = 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(63) = XMODE
        BOOLEAN DOLLARTOG;      % TRUE IF SCANNING A DOLLAR CARD.                       01001540 T  0003:0002:1
PRT(64) = DOLLARTOG
        INTEGER ERRMAX;         % COMPILATION STOPS IF EXCEEDED.                        01001550 T  0003:0002:1
PRT(65) = ERRMAX
        BOOLEAN SEQXEQTOG;      % GIVE SEQ. NO. WHEN DS-ING OBJ.                        01001560 T  0003:0002:1
PRT(66) = SEQXEQTOG
        BOOLEAN LISTER;         % LISTOG OR LISTATOG OR DEBUGTOG.                       01001570 T  0003:0002:1
PRT(67) = LISTER
```

```
                ALPHA MEDIUM;           % INPUT IS: T,C,P,CA,CB,CC.                                  01001580 T  0003:0002:1
PRT(70) = MEDIUM
                INTEGER MYCLASS;        % USED IN DOLLARCARD EVALUATION.                              01001590 T  0003:0002:1
PRT(71) = MYCLASS
                REAL BATMAN;            % USED IN DOLLARCARD EVALUATION.                              01001600 T  0003:0002:1
PRT(72) = BATMAN
                 ARRAY SPECIAL[0:31];                                                                 01003000 T  0003:0002:1
PRT(73) = 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
                                                                                                      01006000 T  0003:0005:2
                ARRAY INFO [0:127,0:255];                                                             01007000 T  0003:0005:2
PRT(74) = INFO
                %*************************************************************************           01007005 C  0003:0007:3
                %                 X R E F   S T U F F                                 %110-           01007010 C  0003:0007:3
                %*************************************************************************           01007015 C  0003:0007:3
                %                                                                     %110-           01007020 C  0003:0007:3
                ARRAY                                                                 %110-           01007025 C  0003:0007:3
                  XREFAY2[0:29],        % ARRAY OF ONE WORD REFERENCE RECORDS,       %110-           01007030 C  0003:0007:3
PRT(75) = XREFAY2
                                        % THE LAYOUT OF EACH WORD IS                 %110-           01007035 C  0003:0010:1
                                        %                                           %110-           01007040 C  0003:0010:1
                                        %         .[1:5]  TYPE OF REFERENCE         %110-           01007045 C  0003:0010:1
                                        %                   = 0 FOR FORWARD DECL    %110-           01007050 C  0003:0010:1
                                        %                   = 1 FOR LABEL OCCURENCE %110-           01007051 C  0003:0010:1
                                        %                   = 2 FOR NORMAL DECL     %110-           01007055 C  0003:0010:1
                                        %                   = 4 FOR NORMAL REFERENCE%110-           01007060 C  0003:0010:1
                                        %                   = 5 FOR ASSIGNMENT      %110-           01007065 C  0003:0010:1
                                        %                                           %110-           01007070 C  0003:0010:1
                                        %                     NOTE: THE LOWER ORDER BIT             01007075 C  0003:0010:1
                                        %                           OF THIS FIELD IS ON             01007080 C  0003:0010:1
                                        %                           IF YOU WANT STARS%110-          01007085 C  0003:0010:1
                                        %                           AROUND THIS REFERENCE           01007090 C  0003:0010:1
                                        %                           IN THE XREF     %110-           01007095 C  0003:0010:1
                                        %                                           %110-           01007100 C  0003:0010:1
                                        %         .[6:15] IDENTIFIER ID. NO.        %110-           01007105 C  0003:0010:1
                                        %                   THIS IS A UNIQUE NUMBER THAT            01007110 C  0003:0010:1
                                        %                   IS ASSIGNED WHEN THE    %110-           01007115 C  0003:0010:1
                                        %                   IDENTIFIER IS ENCOUNTERE%110-           01007120 C  0003:0010:1
                                        %                   FOR THE FIRST TIME.     %110-           01007125 C  0003:0010:1
                                        %                                           %110-           01007130 C  0003:0010:1
                                        %         .[21:27] SEQUENCE NUMBER          %110-           01007135 C  0003:0010:1
                                        %                                           %110-           01007140 C  0003:0010:1
                    XREFAY1[0:9],       % RECORD BUFFER AREA FOR WRITING OUT THE    %110-           01007145 C  0003:0010:1
PRT(76) = XREFAY1
                                        % NAME INFORMATION RECORDS, ONE RECORD      %110-           01007150 C  0003:0013:2
                                        % IS WRITTEN FOR EACH IDENTIFIER IN THE SYMBOL               01007155 C  0003:0013:2
                                        % TABLE WHEN THE IDENTIFIER IS PURGED FROM THE               01007160 C  0003:0013:2
                                        % SYMBOL TABLE, I.E., WHEN LEAVING THE BLOCK%110-           01007165 C  0003:0013:2
                                        % IN WHICH THE IDENTIFIER IS DECLARED.      %110-           01007170 C  0003:0013:2
                                        %                                           %110-           01007175 C  0003:0013:2
                                        % THE LAYOUT OF EACH RECORD IS:             %110-           01007180 C  0003:0013:2
                                        %                                           %110-           01007185 C  0003:0013:2
                                        %      WORDS 0-7   THE IDENTIFIER WITH BLANK%110-           01007190 C  0003:0013:2
                                        %                    FILL ON THE RIGHT      %110-           01007195 C  0003:0013:2
                                        %                                           %110-           01007200 C  0003:0013:2
                                        %      WORD 8                               %110-           01007205 C  0003:0013:2
                                        %        .[21:12]  SEGMENT NUMBER IN WHICH  %110-           01007210 C  0003:0013:2
```

```
                                        %                    THIS IDENTIFIER WAS DECLARED          01007215 C  0003:0013:2
                                        %                                                %110-     01007220 C  0003:0013:2
                                        %          ,[33:15]  IDENTIFIER ID. NO.          %110-     01007225 C  0003:0013:2
                                        %                                                %110-     01007230 C  0003:0013:2
                                        %       WORD 9       ELBAT WORD                  %110-     01007235 C  0003:0013:2
                                        %                                                %110-     01007240 C  0003:0013:2
                    XINFO[0:31,0:127];  % THIS ARRAY CONTAINS ONE ENTRY FOR EACH ENTRY               01007245 C  0003:0013:2
PRT(77) = XINFO
                                        % IN THE INFO TABLE. IF YOU HAVE THE INDEX  %110-          01007250 C  0003:0015:3
                                        % OF THE ELBAT WORD FOR AN IDENTIFIER IN    %110-          01007255 C  0003:0015:3
                                        % THE INFO TABLE YOU CAN FIND THE XINFO WORD%110-          01007260 C  0003:0015:3
                                        % FOR THE IDENTIFIER  BY REFERRING TO:      %110-          01007265 C  0003:0015:3
                                        %                                          %110-          01007270 C  0003:0015:3
                                        %      XINFO[INDEX,LINKR,INDEX,LINKC DIV 2] %110-          01007275 C  0003:0015:3
                                        %                                          %110-          01007280 C  0003:0015:3
                                        % EACH ENTRY CONTAINS:                      %110-          01007285 C  0003:0015:3
                                        %                                          %110-          01007290 C  0003:0015:3
                                        %            .[21:12]  SEGMENT NUMBER IN WHICH             01007295 C  0003:0015:3
                                        %                        THIS IDENTIFIER WAS DECL          01007300 C  0003:0015:3
                                        %                                          %110-          01007305 C  0003:0015:3
                                        %            .[33:15]  IDENTIFIER ID. NO.  %110-          01007310 C  0003:0015:3
                                        %                        IF THIS ID. NO. IS ZERO           01007315 C  0003:0015:3
                                        %                        THEN XREF WAS NOT ON              01007320 C  0003:0015:3
                                        %                        AT THE TIME THE IDENT             01007325 C  0003:0015:3
                                        %                        WAS DECLARED AND ALL              01007330 C  0003:0015:3
                                        %                        FUTURE REFERENCES WILL            01007335 C  0003:0015:3
                                        %                        BE DISCARDED.     %110-           01007340 C  0003:0015:3
                                        %                                          %110-          01007345 C  0003:0015:3
                  INTEGER               %                                          %110-          01007350 C  0003:0015:3
                    XREFPT,             % CONTAINS INDEX OF NEXT AVAILABLE SLOT IN  %110-          01007355 C  0003:0015:3
PRT(100) = XREFPT
                                        % XREFAY2. WHEN THIS BECOMES GREATER        %110-          01007360 C  0003:0015:3
                                        % THAN 30 THE CURRENT ARRAY IS DUMPED TO DISK               01007365 C  0003:0015:3
                                        % AND XREFPT IS RESET TO ZERO.              %110-          01007370 C  0003:0015:3
                                        %                                          %110-          01007375 C  0003:0015:3
                    XLUN;               % THIS VARIABLE CONTROLS THE ASSIGNING OF   %110-          01007380 C  0003:0015:3
PRT(101) = XLUN
                                        % ID. NO. TO IDENTIFIERS. IT IS INCREMENTED %110-          01007385 C  0003:0015:3
                                        % EACH TIME A NEW IDENTIFIER IS ENCOUNTERED.%110-          01007390 C  0003:0015:3
                                        %                                          %110-          01007395 C  0003:0015:3
                  DEFINE                %                                          %110-          01007400 C  0003:0015:3
                    SEGNOF = [21:12]#,  % FIELDS IN XINFO ENTRIES AND WORD 8 OF    %110-          01007405 C  0003:0015:3
                    IDNOF  = [33:15]#,  % IDENTIFIER RECORDS.                      %110-          01007410 C  0003:0015:3
                                        %                                          %110-          01007415 C  0003:0015:3
                    TYPEREF = [1:5]#,   % FIELDS OF REFERENCE WORDS               %110-          01007420 C  0003:0015:3
                    REFIDNOF =[6:15]#,  %                                          %110-          01007425 C  0003:0015:3
                    SEQNOF = [21:27]#,  %                                          %110-          01007430 C  0003:0015:3
                                        %                                          %110-          01007435 C  0003:0015:3
                    XREFIT(INDEX,SEQNO,REFTYPE) = % DEFINE TO ADD INFO TO REF TABLE %110-          01007440 C  0003:0015:3
                      BEGIN IF XREF THEN CROSSREFIT(INDEX,SEQNO,REFTYPE); END#,     %110-          01007445 C  0003:0015:3
                                        %                                          %110-          01007450 C  0003:0015:3
                    XMARK(REFTYPE) =    % DEFINE TO CHANGE LAST ENTRY IN REF TABLE TO A          01007455 C  0003:0015:3
                      BEGIN IF XREF THEN XREFAY2[XREFPT-1].TYPEREF := REFTYPE END#,  %110-          01007460 C  0003:0015:3
                                        %                                          %110-          01007465 C  0003:0015:3
                    XREFDUMP(INDEX) =   % DEFINE TO DUMP SYMBOL TABLE INFO FOR IDENTIFIER 01007470 C  0003:0015:3
                      BEGIN IF DEFINING.[1:1] THEN CROSSREFDUMP(INDEX); END#,       %110-          01007475 C  0003:0015:3
                                        %                                          %110-          01007480 C  0003:0015:3
```

```
                    XREFINFO[INDEX] =     % DEFINE TO TRANSLATE INFO ROW AND COLUMN TO%110-        01007481 C  0003:0015:3
                      XINFO[(INDEX).LINKR,(INDEX).LINKC DIV 2]#, % XINFO ROW AND COL %110-        01007482 C  0003:0015:3
                                          %                                          %110-        01007483 C  0003:0015:3
                    FORWARDREF = 0#,      % DEFINES FOR DIFFERENT REFERENCE TYPES     %110-        01007485 C  0003:0015:3
                    LBLREF     = 1#,      %                                          %110-        01007486 C  0003:0015:3
                    DECLREF    = 2#,      %                                          %110-        01007490 C  0003:0015:3
                    NORMALREF  = 4#,      %                                          %110-        01007495 C  0003:0015:3
                    ASSIGNREF  = 5#;      %                                          %110-        01007500 C  0003:0015:3
                      ARRAY BEGINSTACK[0:255]; INTEGER BSPOINT;                      %108-        01007600 C  0003:0015:3
PRT(102) = BEGINSTACK
PRT(103) = BSPOINT
                    BOOLEAN DEFINING;                                                %108-        01007650 C  0003:0018:0
PRT(104) = DEFINING
                    COMMENT INFO CONTAINS ALL THE INFORMATION ABOUT A GIVEN IDENTIFIER                01008000 T  0003:0018:0
                            OR RESERVED WORD.  THE FIRST WORD OF A GIVEN ENTRY IS                     01009000 T  0003:0018:0
                            THE INTERNAL CODE ( OR ELBAT WORD AS IT IS USUALLY                        01010000 T  0003:0018:0
                            CALLED).  THE SECOND WORD CONTAINS THE FORWARD BIT (IN                    01011000 T  0003:0018:0
                            [1:1]) FOR PROCEDURES, THE LINK TO PREVIOUS ENTRY (IN                     01012000 T  0003:0018:0
                            [4:8]). THE NUMBER OF CHARACTORS IN THE ALPHA REPRESENTA-                  01013000 T  0003:0018:0
                            TION (IN [12:6]), AND THE FIRST 5 CHARACTORS OF ALPHA.                    01014000 T  0003:0018:0
                            SUCCEDING WORDS CONTAIN THE REMAINING CHARACTORS OF ALPHA,                01015000 T  0003:0018:0
                            FOLLOWED BY ANY ADDITIONAL INFORMATION. THE ELBAT WORD                    01016000 T  0003:0018:0
                            AND THE ALPHA FOR ANY QUANTITY ARE NOT SPLIT ACROSS A ROW                 01017000 T  0003:0018:0
                            OF INFO.  FOR PURPOSES OF FINDING AN IDENTIFIER OR                        01018000 T  0003:0018:0
                            RESERVED WORD THE QUANTITIES ARE SCATTERED INTO 125                       01019000 T  0003:0018:0
                            DIFERENT LISTS OR STACKES, WHICH STACK CONTAINS A QUANTITY                01020000 T  0003:0018:0
                            IS GIVEN BY TAKING NAAAAA MOD 125 WHERE N IS THE NUMBER                   01021000 T  0003:0018:0
                            OF CHARACTORS AND AAAAA IS THE FIRST 5 CHARACTORS OF                      01022000 T  0003:0018:0
                            ALPHA, FILLED IN WITH ZEROS FROM THE RIGHT IF NEEDED.                     01023000 T  0003:0018:0
                            THIS NUMBER IS CALLED THE SCRAMBLE NUMBER OR INDEX.                       01024000 T  0003:0018:0
                            THE FIRST ROW OF INFO IS USED FOR OTHER PURPOSES.  THE                    01025000 T  0003:0018:0
                            RESERVED WORDS OCCUPY THE SECOND ROW. IT IS FILLED DURING                 01026000 T  0003:0018:0
                            INITIALIZATION;                                                           01027000 T  0003:0018:0
                    COMMENT                 INFO FORMAT                                               01028000 T  0003:0018:0
                      FOLLOWING IS A DESCRIPTION OF THE FORMAT OF ALL TYPES OF ENTRIES                01029000 T  0003:0018:0
                      ENTERED IN INFO:                                                                01030000 T  0003:0018:0
                         THE FIRST WORD OF ALL ENTRIES IS THE ELBAT WORD.                             01031000 T  0003:0018:0
                         THE INCR FIELD ([27:8]) CONTAINS AN INCREMENT WHICH WHEN                     01032000 T  0003:0018:0
                         ADDED TO THE CURRENT INDEX INTO INFO YELDSAN INDEX TO ANY                    01033000 T  0003:0018:0
                         ADDITIONAL INFO (IF ANY) FOR THIS ENTRY.                                     01034000 T  0003:0018:0
                          E.G. IF THE INDEX IS  IX  THEN  INFO[(IX+INCR),LINKR,(IX+INCR),             01035000 T  0003:0018:0
                              LINKC] WILL CONTAIN THE FIRST WORD OF ADDITIONAL INFO.                   01036000 T  0003:0018:0
                         THE LINK FIELD OF THE ELBAT WORD IN INFO IS DIFFERENT FROM                   01037000 T  0003:0018:0
                         THAT OF THE ENTRY IN ELBAT PUT IN BY TABLE.THE ENTRY IN ELBAT                01038000 T  0003:0018:0
                         POINTS TO ITS OWN LOCATION (RELATIVE) IN INFO.                               01039000 T  0003:0018:0
                         THE LINK IN INFO POINTS TO THE PREVIOUS ENTRY E.G.,THE                       01040000 T  0003:0018:0
                         LINK FROM STACKHEAD WHICH THE CURRENT ENTRY REPLACED.                        01041000 T  0003:0018:0
                         FOR SIMPLICITY,I WILL CONSIDER INFO TO BE A ONE DIMENSIONAL                  01042000 T  0003:0018:0
                         ARRAY,SO THAT THE BREAKING UP OF THE LINKS INTO ROW AND COLUMN               01043000 T  0003:0018:0
                         WILL NOT DETRACT FROM THE DISCUSSION.                                        01044000 T  0003:0018:0
                            ASSUME THAT THREE IDENTIFIERS A,B,AND C "SCRAMBLE" INTO                   01045000 T  0003:0018:0
                         THE SAME STACKHEAD LOCATION IN THE ORDER OF APPEARANCE.                      01046000 T  0003:0018:0
                           FURTHER ASSUME THERE ARE NO OTHER ENTRIES CONNECTED TO                     01047000 T  0003:0018:0
                         THIS STACKHEAD INDEX, LET THIS STACKHEAD LOCATION BE                         01048000 T  0003:0018:0
                                        S[L]                                                          01049000 T  0003:0018:0
                         NOW THE DECLARATION                                                          01050000 T  0003:0018:0
                              BEGIN REAL A,B,C  IS ENCOUNTERED                                        01051000 T  0003:0018:0
```

```
   IF THE NEXT AVAILABLE INFO SPACE IS CALLED NEXTINFO
 THEN  A IS ENTERED AS FOLLOWS:(ASSUME AN ELBAT WORD T HAS BEEN
                                CONSTRUCTED FOR A)
                        T.LINK← S[L]. (WHICH IS ZERO AT FIRST).
INFO[NEXTINFO]←T.       S[L]←NEXTINFO.
                        NEXTINFO←NEXTINFO+NUMBER OF WORDS IN THIS
                                 ENTRY.
  NOW  S[L] POINTS TO THE ENTRY FOR A IN INFO AND THE ENTRY
          ITSELF CONTAINS THE STOP FLAG ZERO.
 B IS ENTERED SIMILARLY TO A.
  NOW  S[L] POINTS TO THE ENTRY FOR B AND IT POINTS TO THE
                        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
```

```
01052000 T  0003:0018:0
01053000 T  0003:0018:0
01054000 T  0003:0018:0
01055000 T  0003:0018:0
01056000 T  0003:0018:0
01057000 T  0003:0018:0
01058000 T  0003:0018:0
01059000 T  0003:0018:0
01060000 T  0003:0018:0
01061000 T  0003:0018:0
01062000 T  0003:0018:0
01063000 T  0003:0018:0
01064000 T  0003:0018:0
01065000 T  0003:0018:0
01066000 T  0003:0018:0
01067000 T  0003:0018:0
01068000 T  0003:0018:0
01069000 T  0003:0018:0
01070000 T  0003:0018:0
01071000 T  0003:0018:0
01072000 T  0003:0018:0
01073000 T  0003:0018:0
01074000 T  0003:0018:0
01075000 T  0003:0018:0
01076000 T  0003:0018:0
01077000 T  0003:0018:0
01078000 T  0003:0018:0
01079000 T  0003:0018:0
01080000 T  0003:0018:0
01081000 T  0003:0018:0
01082000 T  0003:0018:0
01083000 T  0003:0018:0
01084000 T  0003:0018:0
01085000 T  0003:0018:0
01086000 T  0003:0018:0
01087000 T  0003:0018:0
01088000 T  0003:0018:0
01089000 T  0003:0018:0
01090000 T  0003:0018:0
01091000 T  0003:0018:0
01092000 T  0003:0018:0
01093000 T  0003:0018:0
01094000 T  0003:0018:0
01095000 T  0003:0018:0
01096000 T  0003:0018:0
01097000 T  0003:0018:0
01098000 T  0003:0018:0
01099000 T  0003:0018:0
01100000 T  0003:0018:0
01101000 T  0003:0018:0
01102000 T  0003:0018:0
01103000 T  0003:0018:0
01104000 T  0003:0018:0
01105000 T  0003:0018:0
01106000 T  0003:0018:0
01107000 T  0003:0018:0
01108000 T  0003:0018:0
```

```
                    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]
     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.
     SUCCEDING 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]#,
                 SBITF  =[21:6]#, % STARTING BIT FOR FIELD ID.  %112-
                 NBITF  =[27:6]#, % NUMBER OF BITS FOR FIELD ID.%112-
                 LINKR  =[35: 5]#,
                 LINKC  =[40: 8]#;
COMMENT THESE DEFINES ARE USED TO PICK APART THE ELBAT WORD.
          MON IS THE BIT WHICH IS ON IF THE QUANTITY IS MONITORED.
          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
```

```
01109000 T  0003:00_18:0
01110000 T  0003:00_18:0
01111000 T  0003:00_18:0
01112000 T  0003:00_18:0
01113000 T  0003:00_18:0
01114000 T  0003:00_18:0
01115000 T  0003:00_18:0
01116000 T  0003:00_18:0
01117000 T  0003:00_18:0
01118000 T  0003:00_18:0
01119000 T  0003:00_18:0
01120000 T  0003:00_18:0
01121000 T  0003:00_18:0
01122000 T  0003:00_18:0
01123000 T  0003:00_18:0
01124000 T  0003:00_18:0
01125000 T  0003:00_18:0
01126000 T  0003:00_18:0
01127000 T  0003:00_18:0
01128000 T  0003:00_18:0
01129000 T  0003:00_18:0
01130000 T  0003:00_18:0
01131000 T  0003:00_18:0
01132000 T  0003:00_18:0
01133000 T  0003:00_18:0
01134000 T  0003:00_18:0
01135000 T  0003:00_18:0
01136000 T  0003:00_18:0
01137000 T  0003:00_18:0
01138000 T  0003:00_18:0
01139000 T  0003:00_18:0
01140000 T  0003:00_18:0
01141000 T  0003:00_18:0
01142000 T  0003:00_18:0
01143000 T  0003:00_18:0
01144000 T  0003:00_18:0
01145000 T  0003:00_18:0
01146000 T  0003:00_18:0
01147000 T  0003:00_18:0
01148000 T  0003:00_18:0
01149000 T  0003:00_18:0
01150000 T  0003:00_18:0
01151000 T  0003:00_18:0
01152000 T  0003:00_18:0
01153000 T  0003:00_18:0
01154000 T  0003:00_18:0
01154200 C  0003:00_18:0
01154300 C  0003:00_18:0
01155000 T  0003:00_18:0
01156000 T  0003:00_18:0
01157000 T  0003:00_18:0
01158000 T  0003:00_18:0
01159000 T  0003:00_18:0
01160000 T  0003:00_18:0
01161000 T  0003:00_18:0
01162000 T  0003:00_18:0
01163000 T  0003:00_18:0
01164000 T  0003:00_18:0
```

```
                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.
            INCR GIVES A RELATIVE LINK TO ANY ADDITIONAL INFORMATION
                NEEDED, RELATIVE TO THE LOCATION IN INFO.
            LINK CONTAINS A LINK TO THE LOCATION IN INFO IF THE
                QUANTITY LIES IN ELBAT, OTHERWISE IT LINKS TO THE
                NEXT ITEM IN THE STACK, ZERO IS AN END FLAG.
            LINKR AND LINKC ARE SUBDIVISIONS OF LINK.;
COMMENT  CLASSES FOR ALL QUANTITIES  -  OCTAL CLASS IS IN COMMENT;
   COMMENT   CLASSES FOR IDENTIFIERS;
DEFINE  UNKNOWNID       =00#,    COMMENT 000;
        STLABID         =01#,    COMMENT 001;
        LOCLID          =02#,    COMMENT 002;
        DEFINEDID       =03#,    COMMENT 003;
        LISTID          =04#,    COMMENT 004;
        FRMTID          =05#,    COMMENT 005;
        SUPERFRMTID     =06#,    COMMENT 006;
        REALSUBID       =07#,    COMMENT 007;
        SUBID           =08#,    COMMENT 010;
        SWITCHID        =09#,    COMMENT 011;
        PROCID          =10#,    COMMENT 012;
        INTRNSICPROCID  =11#,    COMMENT 013;
        STRPROCID       =12#,    COMMENT 014;
        BOOSTRPROCID    =13#,    COMMENT 015;
        REALSTRPROCID   =14#,    COMMENT 016;
        ALFASTRPROCID   =15#,    COMMENT 017;
        INTSTRPROCID    =15#,    COMMENT 017;
        BOOPROCID       =17#,    COMMENT 021;
        REALPROCID      =18#,    COMMENT 022;
        ALFAPROCID      =19#,    COMMENT 023;
        INTPROCID       =19#,    COMMENT 023;
        BOOID           =21#,    COMMENT 025;
        REALID          =22#,    COMMENT 026;
        ALFAID          =23#,    COMMENT 027;
        INTID           =23#,    COMMENT 027;
        BOOARRAYID      =25#,    COMMENT 031;
        REALARRAYID     =26#,    COMMENT 032;
        ALFAARRAYID     =27#,    COMMENT 033;
        INTARRAYID      =27#,    COMMENT 033;
        NAMEID          =30#,    COMMENT 036;
        INTNAMEID       =31#,    COMMENT 037;
        LABELID         =32#,    COMMENT 040;
COMMENT   CLASSES FOR PRIMARY BEGINNERS;
        TRUTHV          =33#,    COMMENT 041;
        NONLITNO        =34#,    COMMENT 042;
        LITNO           =35#,    COMMENT 043;
        STRNGCON        =36#,    COMMENT 044;
        LEFTPAREN       =37#,    COMMENT 045;
        POLISHV         =38#,    COMMENT 046;
        ASTRISK         =39#,    COMMENT 047;
COMMENT   CLASS FOR ALL DECLARATORS;
        DECLARATORS     =40#,    COMMENT 050;
COMMENT   CLASSES FOR STATEMENT BEGINNERS
        DOUBLEV         =42#,    COMMENT 052;
        FORV            =43#,    COMMENT 053;
```

```
01165000 T  0003:0018:0
01166000 T  0003:0018:0
01167000 T  0003:0018:0
01168000 T  0003:0018:0
01169000 T  0003:0018:0
01170000 T  0003:0018:0
01171000 T  0003:0018:0
01172000 T  0003:0018:0
01173000 T  0003:0018:0
01174000 T  0003:0018:0
01175000 T  0003:0018:0
01176000 T  0003:0018:0
01177000 T  0003:0018:0
01178000 T  0003:0018:0
01179000 T  0003:0018:0
01180000 T  0003:0018:0
01181000 T  0003:0018:0
01182000 T  0003:0018:0
01183000 T  0003:0018:0
01184000 T  0003:0018:0
01185000 T  0003:0018:0
01186000 T  0003:0018:0
01187000 T  0003:0018:0
01188000 T  0003:0018:0
01189000 T  0003:0018:0
01190000 T  0003:0018:0
01191000 T  0003:0018:0
01192000 T  0003:0018:0
01193000 T  0003:0018:0
01194000 T  0003:0018:0
01195000 T  0003:0018:0
01196000 T  0003:0018:0
01197000 T  0003:0018:0
01198000 T  0003:0018:0
01199000 T  0003:0018:0
01200000 T  0003:0018:0
01201000 T  0003:0018:0
01202000 T  0003:0018:0
01203000 T  0003:0018:0
01204000 T  0003:0018:0
01205000 T  0003:0018:0
01205200 T  0003:0018:0
01205400 T  0003:0018:0
01206000 T  0003:0018:0
01207000 T  0003:0018:0
01208000 T  0003:0018:0
01209000 T  0003:0018:0
01210000 T  0003:0018:0
01211000 T  0003:0018:0
01212000 T  0003:0018:0
01212100 T  0003:0018:0
01212200 T  0003:0018:0
01213000 T  0003:0018:0
01214000 T  0003:0018:0
01215000 T  0003:0018:0
01222000 T  0003:0018:0
01223000 T  0003:0018:0
```

```
      WHILEV          =44#,    COMMENT 054;
      DOV             =45#,    COMMENT 055;
      UNTILV          =46#,    COMMENT 056;
      ELSEV           =47#,    COMMENT 057;
      ENDV            =48#,    COMMENT 060;
      CASEV           =49#,    COMMENT 061;
      SEMICOLON       =50#,    COMMENT 062;
      IFV             =51#,    COMMENT 063;
      GOV             =52#,    COMMENT 064;
      IOCLASS         =53#,    COMMENT 065;
      BEGINV          =54#,    COMMENT 066;
COMMENT   CLASSES FOR STREAM RESERVED WORDS;
      SIV             =55#,    COMMENT 067;
      DIO             =56#,    COMMENT 070;
      CIV             =57#,    COMMENT 071;
      TALLYV          =58#,    COMMENT 072;
      DSV             =59#,    COMMENT 073;
      SKIPV           =60#,    COMMENT 074;
      JUMPV           =61#,    COMMENT 075;
      DBV             =62#,    COMMENT 076;
      SBV             =63#,    COMMENT 077;
      TOGGLEV         =64#,    COMMENT 100;
      SCV             =65#,    COMMENT 101;
      LOCV            =66#,    COMMENT 102;
      DCV             =67#,    COMMENT 103;
      LOCALV          =68#,    COMMENT 104;
      LITV            =69#,    COMMENT 105;
      TRNSFER         =70#,    COMMENT 106;
COMMENT   CLASSES FOR VARIOUS MISCELLANEOUS QUANTITIES;
      COMMENTV        =71#,    COMMENT 107;
      FORWARDV        =72#,    COMMENT 110;
      STEPV           =73#,    COMMENT 111;
      THENV           =74#,    COMMENT 112;
      TOV             =75#,    COMMENT 113;
      VALUEV          =76#,    COMMENT 114;
      WITHV           =77#,    COMMENT 115;
      COLON           =78#,    COMMENT 116;
      COMMA           =79#,    COMMENT 117;
      CROSSHATCH      =80#,    COMMENT 120;
      LFTBRKET        =81#,    COMMENT 121;
      PERIOD          =82#,    COMMENT 122;
      RTBRKET         =83#,    COMMENT 123;
      RTPAREN         =84#,    COMMENT 124;
      AMPERSAND       =85#,    COMMENT 125;
COMMENT   CLASSES FOR OPERATORS;
      HEXOP           =86#,    COMMENT 126;
      BITOP           =87#,    COMMENT 127;
      ISOLATE         =88#,    COMMENT 130;
      OPERATOR        =89#,    COMMENT 131;
      NOTOP           =90#,    COMMENT 132;
      ASSIGNOP        =91#,    COMMENT 133;
      EQVOP           =92#,    COMMENT 134;
      OROP            =93#,    COMMENT 135;
      ANDOP           =94#,    COMMENT 136;
      RELOP           =95#,    COMMENT 137;
      ADOP            =96#,    COMMENT 140;
      MULOP           =97#,    COMMENT 141;
```

```
                                01224000 T  0003;001810
                                01225000 T  0003;001810
                                01226000 T  0003;001810
                                01227000 T  0003;001810
                                01228000 T  0003;001810
%115-                           01229000 C  0003;001810
                                01230000 T  0003;001810
                                01231000 T  0003;001810
                                01232000 T  0003;001810
                                01233000 T  0003;001810
                                01234000 T  0003;001810
                                01235000 T  0003;001810
                                01236000 T  0003;001810
                                01237000 T  0003;001810
                                01238000 T  0003;001810
                                01239000 T  0003;001810
                                01240000 T  0003;001810
                                01241000 T  0003;001810
                                01242000 T  0003;001810
                                01243000 T  0003;001810
                                01244000 T  0003;001810
                                01245000 T  0003;001810
                                01246000 T  0003;001810
                                01247000 T  0003;001810
                                01248000 T  0003;001810
                                01249000 T  0003;001810
                                01250000 T  0003;001810
                                01251000 T  0003;001810
                                01252000 T  0003;001810
                                01253000 T  0003;001810
                                01254000 T  0003;001810
                                01255000 T  0003;001810
                                01256000 T  0003;001810
                                01257000 T  0003;001810
                                01258000 T  0003;001810
                                01259000 T  0003;001810
                                01260000 T  0003;001810
                                01261000 T  0003;001810
                                01262000 T  0003;001810
                                01263000 T  0003;001810
                                01264000 T  0003;001810
                                01265000 T  0003;001810
                                01266000 T  0003;001810
                                01266500 T  0003;001810
                                01267000 T  0003;001810
                                01268000 T  0003;001810
                                01269000 T  0003;001810
                                01270000 T  0003;001810
                                01271000 T  0003;001810
                                01272000 T  0003;001810
                                01273000 T  0003;001810
                                01274000 T  0003;001810
                                01275000 T  0003;001810
                                01276000 T  0003;001810
                                01277000 T  0003;001810
                                01278000 T  0003;001810
                                01278500 T  0003;001810
```

```
%         STRING          =99#,    COMMENT 143;                                  01278600 T  0003:001B:0
          FIELDID         =125#,   COMMENT 175;                  %112-          01278700 C  0003:001B:0
     COMMENT   SUBCLASSES FOR DECLARATORS (KEPT IN ADDRESS);                      01279000 T  0003:001B:0
          OWNV            =01#,    COMMENT  01;                                  01280000 T  0003:001B:0
          SAVEV           =02#,    COMMENT  02;                                  01281000 T  0003:001B:0
          BOOV            =03#,    COMMENT  03;                                  01282000 T  0003:001B:0
          REALV           =04#,    COMMENT  04;                                  01283000 T  0003:001B:0
          ALFAV           =05#,    COMMENT  05;                                  01284000 T  0003:001B:0
          INTV            =05#,    COMMENT  05;                                  01285000 T  0003:001B:0
          LABELV          =07#,    COMMENT  07;                                  01286000 T  0003:001B:0
          DUMPV           =08#,    COMMENT  10;                                  01287000 T  0003:001B:0
          SUBV            =09#,    COMMENT  11;                                  01288000 T  0003:001B:0
          OUTV            =10#,    COMMENT  12;                                  01289000 T  0003:001B:0
          INV             =11#,    COMMENT  13;                                  01290000 T  0003:001B:0
          MONITORV        =12#,    COMMENT  14;                                  01291000 T  0003:001B:0
          SWITCHV         =13#,    COMMENT  15;                                  01292000 T  0003:001B:0
          PROCV           =14#,    COMMENT  16;                                  01293000 T  0003:001B:0
          ARRAYV          =15#,    COMMENT  17;                                  01294000 T  0003:001B:0
          NAMEV           =16#,    COMMENT  20;                                  01295000 T  0003:001B:0
          FILEV           =17#,    COMMENT  21;                                  01296000 T  0003:001B:0
          STREAMV         =18#,    COMMENT  22;                                  01297000 T  0003:001B:0
          DEFINEV         =19#,    COMMENT  23;                  %112-           01298000 P  0003:001B:0
          AUXMEMV         =20#,    COMMENT 24;                   %112-           01298500 C  0003:001B:0
          FIELDV          =21#;    COMMENT 25;                   %112-           01298600 C  0003:001B:0
DEFINE DDES          =  8#,                                                      01299000 T  0003:001B:0
       ADES          = 28#,                                                      01299010 T  0003:001B:0
       PDES          = 29#,                                                      01299020 T  0003:001B:0
       LDES          = 30#,                                                      01299030 T  0003:001B:0
       CHAR          = 31#,                                                      01299040 T  0003:001B:0
       FACTOP        = ASTRISK#,                                                 01299100 T  0003:001B:0
       OPERATORS     = HEXOP#,                                                   01299200 T  0003:001B:0
       FILEID        = 0#,                                                       01299300 T  0003:001B:0
       MAXINTRINSIC = 150#, % USED IN BUILDING INTABLE @ 09414120                01299400 T  0003:001B:0
       INTRINSICADR = (MAXINTRINSIC DIV 30)#; % RESERVES SEG FOR INTABLE         01299500 T  0003:001B:0
                REAL TIME1;                                                      01300000 T  0003:001B:0
PRT(105) = TIME1
                BOOLEAN ASTOG;                                                   01300100 T  0003:001B:0
PRT(106) = ASTOG
                BOOLEAN SAF;                                                     01300200 T  0003:001B:0
PRT(107) = SAF
                INTEGER SCRAM;                                                   01301000 T  0003:001B:0
PRT(110) = SCRAM
                   COMMENT SCRAM CONTAINS THE SCRAMBLE INDEX FOR THE LAST IDENTIFIER  01302000 T  0003:001B:0
                           OR RESERVED WORD SCANNED;                             01303000 T  0003:001B:0
                ALPHA ARRAY ACCUM[0:10];                                         01304000 T  0003:001B:0
PRT(111) = ACCUM
                   COMMENT ACCUM HOLDS THE ALPHA AND CHARACTER COUNT OF THE LAST  01305000 T  0003:0020:1
                           SCANNED ITEM IN A FORM COMPATIBLE WITH ITS APPEARANCE  01306000 T  0003:0020:1
                           IN INFO, THAT IS ACCUM[1] = 00NAAAAA, ACCUM[I] , I> 1,  01307000 T  0003:0020:1
                           HAS ANY ADDITIONAL CHARACTERS, ACCUM[0] IS USED FOR    01308000 T  0003:0020:1
                           THE ELBAT WORD BY THE ENTER ROUTINES;                  01309000 T  0003:0020:1
              ARRAY STACKHEAD[0:125];                                            01310000 T  0003:0020:1
PRT(112) = STACKHEAD
                   COMMENT STACKHEAD[N] CONTAINS AN INDEX INTO INFO GIVING THE TOP  01311000 T  0003:0023:2
                           ITEM IN THE N-TH STACK;                               01312000 T  0003:0023:2
                INTEGER COUNT;                                                   01313000 T  0003:0023:2
PRT(113) = COUNT
```

```
                    COMMENT COUNT CONTAINS THE NUMBER OF CHARACTORS OF THE LAST ITEM          01314000 T  0003:002:2
                            SCANNED;                                                          01315000 T  0003:002:2
                ALPHA Q;                                                                      01316000 T  0003:002:2
PRT(114) = Q
                    COMMENT Q CONTAINS ACCUM[1] FOR THE LAST IDENTIFIER OR RESERVED           01317000 T  0003:002:2
                            WORD SCANNED;                                                     01318000 T  0003:002:2
                ARRAY ELBAT[0:75]; INTEGER I, NXTELBT;                                        01319000 T  0003:002:2
PRT(115) = ELBAT
PRT(116) = I
PRT(117) = NXTELBT
                    COMMENT ELBAT IS AN ARRAY HOLDING ELBAT WORDS FOR RECENTLY SCANNED        01320000 T  0003:002:3
                            QUANTITIES.  THE TABLE ROUTINE MAINTAINS THIS ARRAY.              01321000 T  0003:002:3
                            (ELBAT IS TABLE SPELLED BACKWARDS.)  THE TABLE ROUTINE            01322000 T  0003:002:3
                            GUARANTIES THAT ELBAT ALWAYS CONTAINS THE ELBAT WORDS             01323000 T  0003:002:3
                            FOR THE LAST 10 QUANTITIES SCANNED.  NXTELBT IS AN INDEX          01324000 T  0003:002:3
                            POINTING TO THE NEXT AVAILABLE WORD IN ELBAT.  I IS AN            01325000 T  0003:002:3
                            INDEX USED BY THE REST OF THE COMPILER TO FETCH THINGS            01326000 T  0003:002:3
                            FROM ELBAT.  I IS ALSO MAINTAINED BY THE TABLE ROUTINE;           01327000 T  0003:002:3
                INTEGER ELCLASS;                                                              01328000 T  0003:002:3
PRT(120) = ELCLASS
                    COMMENT ELCLASS USUALLY CONTAINS ELBAT[I].CLASS;                          01329000 T  0003:002:3
                INTEGER FCR, NCR, LCR,TLCR,CLCR;                                              01330000 T  0003:002:3
PRT(121) = FCR
PRT(122) = NCR
PRT(123) = LCR
PRT(124) = TLCR
PRT(125) = CLCR
                INTEGER MAXTLCR;                                                              01331000 T  0003:002:3
PRT(126) = MAXTLCR
                    COMMENT FCR CONTAINS ABSOLUTE ADDRESS OF THE FIRST CHARACTOR OF           01332000 T  0003:002:3
                            THE CARD IMAGE CURRENTLY BEING SCANNED, NCR THE ADDRESS           01333000 T  0003:002:3
                            OF THE NEXT CHARACTOR TO BE SCANNED, AND LCR THE LAST             01334000 T  0003:002:3
                            CHARACTOR (COLUMN 73).  TLCR AND CLCR CONTAIN ADDRESS OF          01335000 T  0003:002:3
                            THE LAST CHARACTER IN THE TAPE AND CARD BUFFERS.  MAXTLCR          01336000 T  0003:002:3
                            IS THE MAXIMUM OF TLCR WHEN THE INPUT IS BLOCKED;                 01337000 T  0003:002:3
                ARRAY TEN[-46:69];                                                            01340000 T  0003:002:3
PRT(127) = TEN

                                                                                              01341000 T  0003:002:0
                DEFINE PRTBASE=129#,PRTOP=896#;  COMMENT BASE AND TOP OF PRT;                 01342000 T  0003:002:0
                ARRAY PRT[PRTBASE:PRTOP];                                                     01343000 T  0003:002:0
PRT(130) = PRT
                INTEGER DISKADR,CORADR; COMMENT GLOBALS FOR PROGDESCBLDR;                     01344000 T  0003:003:1
PRT(131) = DISKADR
PRT(132) = CORADR
                INTEGER SGAVL;COMMENT NEXT AVAILABLE SEGMENT NUMBER;                          01369000 T  0003:003:1
PRT(133) = SGAVL
                INTEGER SGNO;COMMENT THIS IS THE CURRENT SEGMENT NUMBER;                      01370000 T  0003:003:1
PRT(134) = SGNO
                    ARRAY  COP,WOP[0:127];                                                    01371000 T  0003:003:1
PRT(135) = COP
PRT(136) = WOP
                    COMMENT  THE EMIT ROUTINES PLACE EACH SYLLABLE INTO THE EDOC ARRAY        01372000 T  0003:003:0
                            AS SPECIFIED BY "L".                                              01373000 T  0003:003:0
                            IF THE DEBUGTOG IS TRUE COP AND WOP ARE FILLED WITH               01374000 T  0003:003:0
                            THE BCD FOR THE OPERATORS,OTHERWISE THEY ARE NOT USED;            01375000 T  0003:003:0
                REAL  LASTENTRY ;                                                             01376000 T  0003:003:0
PRT(137) = LASTENTRY
                    COMMENT LASTENTRY IS USED BY EMITNUM AND CONSTANTCLEAN, IT POINTS         01377000 T  0003:003:0
```

```
                        INTO INFO[0,*] AT THE NEXT AVAILABLE CELL FOR CONSTANTS;          01378000 T  0003!0034!0
                BOOLEAN MRCLEAN ;                                                         01379000 T  0003!0034!0
PRT(140) = MRCLEAN
                   COMMENT  NO  CONSTANTCLEAN ACTION TAKES PLACE WHILE MRCLEAN IS         01380000 T  0003!0034!0
                            FALSE. THIS FEATURE IS USED BY BLOCK BECAUSE OF THE           01381000 T  0003!0034!0
                            POSSIBILITY THAT CONSTANTCLEAN WILL USE INFO[NEXTINFO]        01382000 T  0003!0034!0
                            DURING AN ARRAY DECLARATION ;                                 01383000 T  0003!0034!0
                REAL GT1,GT2,GT3,GT4,GT5;                                                 01384000 T  0003!0034!0
PRT(141) = GT1
PRT(142) = GT2
PRT(143) = GT3
PRT(144) = GT4
PRT(145) = GT5
                INTEGER GTI1;                                                             01384500 T  0003!0034!0
PRT(146) = GTI1
                   COMMENT THESE VARIABLES ARE USED FOR TEMPORARY STORAGE;                01385000 T  0003!0034!0
                INTEGER RESULT;                                                           01386000 T  0003!0034!0
PRT(147) = RESULT
                   COMMENT THIS VARIABLE IS USED FOR A DUAL PURPOSE BY THE TABLE          01387000 T  0003!0034!0
                            ROUTINE AND THE SCANNER.  THE TABLE ROUTINE USES THIS         01388000 T  0003!0034!0
                            VARIABLE TO SPECIFY SCANNER OPERATIONS AND THE SCANNER        01389000 T  0003!0034!0
                            USES IT TO INFORM THE TABLE ROUTINE OF THE ACTION TAKEN;      01390000 T  0003!0034!0
                INTEGER LASTUSED;                                                         01391000 T  0003!0034!0
PRT(150) = LASTUSED
                   COMMENT LASTUSED IS A VARIABLE THAT CONTROLS THE ACTION OF             01392000 T  0003!0034!0
                            READACARD, THE ROUTINE WHICH READS CARDS AND INITIALIZES      01393000 T  0003!0034!0
                            OR PREPARES THE CARD FOR THE SCANNER.                         01394000 T  0003!0034!0
                         LASTUSED  LAST CARD READ FROM                                    01394500 T  0003!0034!0
                         --------  ------------------                                     01394600 T  0003!0034!0
                            1      CARD READER ONLY, NO TAPE.                             01395000 T  0003!0034!0
                            2      CARD READER, TAPE AND CARD MERGE.                      01396000 T  0003!0034!0
                            3      TAPE, TAPE AND CARD MERGE.                             01397000 T  0003!0034!0
                            4      INITIALIZATION ONLY, CARD ONLY.                        01398000 T  0003!0034!0
                   ;                                                                      01398300 T  0003!0034!0
                   BOOLEAN LINKTOG;                                                       01399000 T  0003!0034!0
PRT(151) = LINKTOG
                      COMMENT LINKTOG IS FALSE IF THE LAST THING EMITTED IS A LINK,       01400000 T  0003!0034!0
                               OTHERWISE IT IS TRUE;                                      01401000 T  0003!0034!0
                   INTEGER LEVEL,FRSTLEVEL,SUBLEVEL,MODE;                                 01402000 T  0003!0034!0
PRT(152) = LEVEL
PRT(153) = FRSTLEVEL
PRT(154) = SUBLEVEL
PRT(155) = MODE
                      COMMENT THESE VARIABLES ARE MAINTAINED BY THE BLOCK ROUTINE TO KEEP 01403000 T  0003!0034!0
                               TRACK OF LEVELS OF DEFINITION.  LEVEL GIVES THE DEPTH OF   01404000 T  0003!0034!0
                               NESTING IN DEFINITION, WHERE EACH BLOCK AND EACH PROCEDURE 01405000 T  0003!0034!0
                               GIVES RISE TO A NEW LEVEL.  SUBLEVEL GIVES THE LEVEL OF    01406000 T  0003!0034!0
                               THE PARAMETERS OF THE PROCEDURE CURRENTLY BEING COMPILED.   01407000 T  0003!0034!0
                               FRSTLEVEL IS THE LEVEL OF THE PARAMETERS OF THE MOST       01408000 T  0003!0034!0
                               GLOBAL OF THE PROCEDURES CURRENTLY BEING COMPILED.  MODE   01409000 T  0003!0034!0
                               IS THE CURRENT DEPTH OF THE PROCEDURE IN WHICH WE ARE      01410000 T  0003!0034!0
                               NESTED (AT COMPILE TIME);                                  01411000 T  0003!0034!0
                   BOOLEAN ERRORTOG;                                                      01412000 T  0003!0034!0
PRT(156) = ERRORTOG
                      COMMENT ERRORTOG IS TRUE IF MESSAGES ARE CURRENTLY ACCEPTABLE TO THE 01413000 T  0003!0034!0
                               ERROR ROUTINES.  ERRORCOUNT IS THE COUNT OF ERROR MSSGS;   01414000 T  0003!0034!0
                   BOOLEAN ENDTOG;    COMMENT ENDTOG TELLS THE TABLE TO ALLOW             01415000 T  0003!0034!0
```

```
PRT(157) = ENDTOG
                                COMMENT TO BE PASSED BACK TO COMPOUNDTAIL;                    01416000 T  0003:0034:0
                BOOLEAN STREAMTOG;                                                            01417000 T  0003:0034:0
PRT(160) = STREAMTOG
                   COMMENT STREAMTOG IS TRUE IF WE ARE COMPILING STREAM STATEMENT.  IT        01418000 T  0003:0034:0
                           IS USED TO CONTROL COUMPOUNDTAIL;                                  01419000 T  0003:0034:0
                   DEFINE FS = 1#, FP = 2#, FL = 3#, FR =4#;                                  01420000 T  0003:0034:0
                      COMMENT THESE DEFINES ARE USED WHEN CALLING THE VARIABLE ROUTINE.       01421000 T  0003:0034:0
                              THEIR PURPOSES IS TO TELL VARIABLE WHO IS CALLING.              01422000 T  0003:0034:0
                              THEIR MEANING IS:                                               01423000 T  0003:0034:0
                              FS MEANS FROM STATEMENT,                                        01424000 T  0003:0034:0
                              FP MEANS FROM PRIMARY,                                          01425000 T  0003:0034:0
                              FL MEANS FROM LIST,                                             01426000 T  0003:0034:0
                              FR MEANS FROM FOR;                                              01427000 T  0003:0034:0
                   INTEGER L;                                                                 01428000 T  0003:0034:0
PRT(161) = L
                      COMMENT   L IS THE LOCATION OF THE NEXT SYLLABLE TO BE EMITTED;         01429000 T  0003:0034:0
                   DEFINE BLOCKCTR = 16#, JUNK = 17 #, XITR = 18 #, LSTRTN = 19#;             01430000 T  0003:0034:0
                   DEFINE ATYPE =3#, BTYPE=ATYPE#,DTYPE=ATYPE#;                               01452000 T  0003:0034:0
                   BOOLEAN TB1;                                                               01457000 T  0003:0034:0
PRT(162) = TB1
                      COMMENT TB1 IS A TEMPORARY BOOLEAN VARIABLE;                            01458000 T  0003:0034:0
                   INTEGER JUMPCTR;                                                           01459000 T  0003:0034:0
PRT(163) = JUMPCTR
                      COMMENT   JUMPCTR IS A VARIABLE USED FOR COMMUNICATION BETWEEN BLOCK     01460000 T  0003:0034:0
                                AND GENGO.  IT GIVES HIGHEST LEVEL TO WHICH A JUMP HAS        01461000 T  0003:0034:0
                                BEEN MADE FROM WITHIN A THE PRESENTLY BEING COMPILED          01462000 T  0003:0034:0
                                SEGMENT.  THE BLOCK COMPILES CODE TO INCREMENT AND DECRE-      01463000 T  0003:0034:0
                                MENT THE BLOCKCTR ON THE BASIS OF JUMPCTR AT COMPLETION       01464000 T  0003:0034:0
                                OF COMPILATION OF A SEGMENT = I.E. THE BLOCKCTR IS TALLIED     01465000 T  0003:0034:0
                                IF LEVEL = JUMPCTR;                                           01466000 T  0003:0034:0
                                                                                              01467000 T  0003:0034:0
                                                                                              01468000 T  0003:0034:0
                                                                                              01469000 T  0003:0034:0
                                                                                              01470000 T  0003:0034:0
                   DEFINE BUMPL = L+L+2#;                                                     01477000 T  0003:0034:0
                      COMMENT BUMPL IS USED MOSTLY TO PREPARE A FORWARD JUMP;                 01478000 T  0003:0034:0
                   DEFINE IDMAX =   LABELID#;                                                 01479000 T  0003:0034:0
                      COMMENT IDMAX IS THE MAXIMUM CLASS NUMBER FOR IDENTIFIERS;              01480000 T  0003:0034:0
                   INTEGER DEFINECTR,DEFINEINDEX;                                             01481000 T  0003:0034:0
PRT(164) = DEFINECTR
PRT(165) = DEFINEINDEX
                   REAL  JOINFO,        COMMENT POINTS TO PSEUDO LABEL FOR JUMP OUTS;         01482000 T  0003:0034:0
PRT(166) = JOINFO
                        LPRT, COMMENT SHOWS LOCATION OF THE LAST LABEL IN THE PRT ;            01483000 T  0003:0034:0
PRT(167) = LPRT
                     NESTLEVEL,         COMMENT  COUNTS NESTING FOR GO TO AND JUMP OUTS;      01484000 T  0003:0034:0
PRT(170) = NESTLEVEL
                      JUMPLEVEL;     COMMENT NUMBER OF LEVELS TO BE JUMPED OUT;              01485000 T  0003:0034:0
PRT(171) = JUMPLEVEL
                   COMMENT THE REALS ABOVE ARE FOR STREAM STATEMENT;                          01486000 T  0003:0034:0
                   ARRAY  MACRO[0:35];                                                        01487000 T  0003:0034:0
PRT(172) = MACRO
                   COMMENT MACRO IS FILLED WITH SYLLABLES FOR STREAM STATEMENT;               01488000 T  0003:0036:1
                REAL  P,     COMMENT CONTAINS NUMBER OF FORMALS FOR STREAM PROCS;           01489000 T  0003:0036:1
PRT(173) = P
                      Z;     COMMENT CONTAINS 1ST WORD OF INFO FOR STREAM FUNCTIONS;        01490000 T  0003:0036:1
PRT(174) = Z
```

```
                    ARRAY NEWTAPBUF[0:9];                                                    01490510 T  0003:0036:1
PRT(175) = NEWTAPBUF
               SAVE ARRAY DEFINEARRAY[0:34];                                  %116-       01491000 P  0003:0039:2
PRT(176) = DEFINEARRAY
                  COMMENT   THESE VARIABLES ARE USED TO CONTROL ACTION OF THE DEFINE.     01492000 T  0003:0041:3
                            DEFINECTR COUNTS DEPTH OF NESTING OF DEFINE-# PAIRS.         01493000 T  0003:0041:3
                            THE CROSSHATCH PART OF THE TABLE ROUTINE USES DEFINECTR       01494000 T  0003:0041:3
                            TO DETERMINE THE MEANING OF A CROSSHATCH.  DEFINEINDEX IS    01495000 T  0003:0041:3
                            THE NEXT AVAILABLE CELL IN THE DEFINEARRAY.  THE DEFINE-     01496000 T  0003:0041:3
                            ARRAY HOLDS THE ALPHA OF THE DEFINE BEING RECREATED AND      01497000 T  0003:0041:3
                            THE PREVIOUS VALUES OF LASTUSED, LCR, AND NCR;               01498000 T  0003:0041:3
                  INTEGER BEGINCTR;                                                      01499000 T  0003:0041:3
PRT(177) = BEGINCTR
                  COMMENT   BEGINCTR GIVES THE NUMBER OF UNMATCHED BEGINS.  IT IS USED  01500000 T  0003:0041:3
                            FOR ERROR CONTROL ONLY;                                      01501000 T  0003:0041:3
                  INTEGER DIALA,DIALB;                                                   01502000 T  0003:0041:3
PRT(200) = DIALA
PRT(201) = DIALB
                  COMMENT THESE VARIABLES GIVE THE LAST VALUE TO WHICH A AND B WERE      01503000 T  0003:0041:3
                            DIALED.  THIS GIVES SOME LOCAL OPTIMIZATION.  EMITD          01504000 T  0003:0041:3
                            WORRIES ABOUT THIS.  OTHER ROUTINES CAUSE A LOSS OF MEMORY   01505000 T  0003:0041:3
                            BY SETTING DIALA AND DIALB TO ZERO;                          01506000 T  0003:0041:3
               BOOLEAN RRB1;  COMMENT RRB1---RRBN ARE BOOLEAN VARIABLES THAT SERVE THE  01522000 T  0003:0041:3
PRT(202) = RRB1
                            SAME FUNCTION AS RR1---RRN FOR REAL VARIABLES.  SEE          01523000 T  0003:0041:3
                            COMMENT AT RR1;                                              01524000 T  0003:0041:3
                     BOOLEAN RRB2;  COMMENT SEE COMMENT AT RRB1 DECLARATION;             01525000 T  0003:0041:3
PRT(203) = RRB2

DEFINE ARRAYMONFILE = [27:11]#;  COMMENT ARRAYMONFILE IS THE DEFINE FOR                  01526000 T  0003:0041:3
                                 THE ADDRESS OF THE FILE DESCRIPTOR IN                   01527000 T  0003:0041:3
                                 THE FIRST WORD OF ADDITIONAL INFO;                      01528000 T  0003:0041:3
DEFINE SVARMONFILE = [37:11]#;  COMMENT MONITORFILE IS THE DEFINE FOR                    01529000 T  0003:0041:3
                                THE ADDRESS OF THE FILE DESCRIPTOR IN                    01530000 T  0003:0041:3
                                INFO FOR MONITORED SIMPLE VARIABLES;                     01531000 T  0003:0041:3
DEFINE NODIMPART = [40:8]#;  COMMENT THE FIRST ADDITIONAL WORD OF INFO                   01532000 T  0003:0041:3
                             FOR ARRAYS CONTAINS THE NUMBER OF DIMENSIONS                01533000 T  0003:0041:3
                             IN NODIMPART;                                               01534000 T  0003:0041:3
DEFINE LABLMONFILE = [13:11]#;  COMMENT LABLMONFILE DESIGNATES THE BIT                   01535000 T  0003:0041:3
                                POSITION IN THE FIRST WORD OF ADDITIONAL                 01536000 T  0003:0041:3
                                INFO THAT CONTAINS THE MONITOR FILE                      01537000 T  0003:0041:3
                                ADDRESS FOR LABELS;                                      01538000 T  0003:0041:3
DEFINE SWITMONFILE = [13:11]#;  COMMENT SWITMONFILE DESIGNATES THE BIT                   01539000 T  0003:0041:3
                                POSITION IN THE FIRST WORD OF ADDITIONAL                 01540000 T  0003:0041:3
                                INFO THAT CONTAINS THE MONITOR FILE                      01541000 T  0003:0041:3
                                ADDRESS FOR LABELS;                                      01542000 T  0003:0041:3
DEFINE FUNCMONFILE = [27:11]#;  COMMENT FUNCMONFILE DESIGNATES THE BIT                   01543000 T  0003:0041:3
                                POSITION IN THE FIRST WORD OF ADDITIONAL                 01544000 T  0003:0041:3
                                INFO THAT CONTAINS THE MONITOR FILE                      01545000 T  0003:0041:3
                                ADDRESS FOR LABELS;                                      01546000 T  0003:0041:3
DEFINE DUMPEE = [2:11]#;  COMMENT THE DUMPEE FIELD IN THE FIRST                          01547000 T  0003:0041:3
                          ADDITIONAL WORD OF INFO FOR LABELS CONTAINS                    01548000 T  0003:0041:3
                          THE ADDRESS OF THE COUNTER THAT IS INCREMENTED                 01549000 T  0003:0041:3
                          EACH TIME THE LABEL IS PASSED IF THAT LABEL                    01550000 T  0003:0041:3
                          APPEARS IN A DUMP DECLARATION;                                 01551000 T  0003:0041:3
DEFINE DUMPOR = [24:11]#;  COMMENT THE DUMPOR FIELD IN THE FIRST                         01552000 T  0003:0041:3
                           ADDITIONAL WORD OF INFO FOR LABELS CONTAINS                   01553000 T  0003:0041:3
                           THE ADDRESS OF THE ROUTINE THAT IS GENERATED                  01554000 T  0003:0041:3
```

```
                                        FROM THE DUMP DECLARATION THAT IN TURN CALLS                    01555000 T  0003:004:3
                                        THE PRINTI ROUTINE;                                             01556000 T  0003:004:3
                    DEFINE SUBOP=48#;                                                                   01556500 T  0003:004:3
                     FILE OUT CODE DISK SERIAL[1:1](1,1023);                                            01556900 T  0003:004:3
PRT(204) = CODE
                    FILE IN CARD(RR1,10,RR2);                                                           01557000 T  0003:004:3
PRT(205) = CARD
                    FILE OUT LINE DISK SERIAL[20:2400](RR3,15,RR4,SAVE 10);                             01558000 T  0003:005:0
PRT(206) = LINE
PRT(207) =
PRT(210) = FILE ATTRBUTS
                     ARRAY LIN[0:20]; COMMENT PRINT OUTPUT BUILT IN LIN;                                01559010 T  0003:005:2
PRT(211) = LIN
                    INTEGER DA;                                                                         01559020 T  0003:005:3
PRT(212) = DA
                    SAVE FILE OUT NEWTAPE DISK SERIAL[20:2400](RR5,RR6,RR7,SAVE 1);                     01560000 T  0003:005:3
PRT(213) = NEWTAPE
                    FILE IN TAPE "OCRDIMG"(2,RR8,RR9);                                                  01561000 T  0003:006:1
PRT(214) = TAPE
                    SAVE ARRAY CBUFF,TBUFF[0:9]; % INPUT BUFFERS.                                       01561056 T  0003:007:1
PRT(215) = CBUFF
PRT(216) = TBUFF
                         FILE DSK1 DISK SERIAL [20:816](2,10,30);                   %108-               01561085 C  0003:007:0
PRT(217) = DSK1
                         FILE DSK2 DISK SERIAL [20:450](2,30,30);                   %108-               01561087 C  0003:007:1
PRT(220) = DSK2
                    FILE OUT CODISK DISK SERIAL [20:600] (2,30,300);                                    01561300 T  0003:008:1
PRT(221) = CODISK
                    FILE OUT DISK DISK [1:2100] "MCP""DISK"(3,30,300,SAVE 99);                          01561400 T  0003:008:2
PRT(222) = DISK
                    DEFINE    MCPTYPE               = 63#,                                              01561410 T  0003:009:0
                              DCINTYPE              = 62#,                                              01561420 T  0003:009:0
                              TSSINTYPE             = 61#;                                              01561430 T  0003:009:0
                    COMMENT   ESPOL CODE FILES ARE UNIQUELY TYPED IN THEIR FILE                         01561440 T  0003:009:0
                              HEADERS.   HEADER[4].[36:6] IS THE FIELD USED TO CONTAIN                  01561450 T  0003:009:0
                              THE TYPE;                                                                 01561460 T  0003:009:0
                    FILE OUT DECK 0 (2,10);                                                             01561500 T  0003:009:0
PRT(223) = DECK
                    FILE STUFF DISK SERIAL[20:150](2,10,30,SAVE 15);                                    01561600 T  0003:009:0
PRT(224) = STUFF
                    ARRAY TWXA[0:16];                                                                   01561700 T  0003:010:2
PRT(225) = TWXA
                     REAL C;                                                                            01562000 T  0003:010:3
PRT(226) = C
                        COMMENT  C CONTAINS ACTUAL VALUE OF LAST CONSTANT SCANNED;                      01563000 T  0003:010:3
                     REAL T;                                                                            01564000 T  0003:010:3
PRT(227) = T
                        COMMENT  T IS A TEMPORARY CELL;                                                 01565000 T  0003:010:3
                     INTEGER TCOUNT;                                                                    01566000 T  0003:010:3
PRT(230) = TCOUNT
                        REAL STACKCT;                                                                   01566010 T  0003:010:3
PRT(231) = STACKCT
                        COMMENT  TCOUNT IS A VARIABLE WHICH HOLDS A PREVIOUS VALUE OF COUNT             01567000 T  0003:010:3
                                  FOR THE USE OF CONVERT;                                               01568000 T  0003:010:3
                            DEFINE LASTSEQUENCE = 149#,                             %115-               01569000 P  0003:010:3
                            LASTSEQROW = 2#;                                                            01570000 T  0003:010:3
                                                                                                        01571000 T  0003:010:3
```

```
                    REAL FOULED;                                                                    01572000 T  0003:010713
PRT(232) = FOULED
                    BOOLEAN                                                                         01573000 T  0003:010713
                         FUNCTOG,    COMMENT TELLS WHETHER PROCEDURE BEING DECLARED IS A         01574000 T  0003:010713
PRT(233) = FUNCTOG
                                             FUNCTION;                                              01575000 T  0003:010713
                         P2,         COMMENT GENERALY TELLS WHETHER OWN WAS SEEN;                   01576000 T  0003:010713
PRT(234) = P2
                         P3,         COMMENT TELLS WHETHER SAVE WAS SEEN;                           01577000 T  0003:010713
PRT(235) = P3
                         VONF,       COMMENT VALUE OR OWN FIELD OF ELBAT WORD;                      01578000 T  0003:010713
PRT(236) = VONF
                         FORMALF,    COMMENT FORMAL FIELD OF ELBAT WORD;                            01579000 T  0003:010713
PRT(237) = FORMALF
                         PTOG,       COMMENT TELLS THAT FORMAL PARAPART IS BEING PROCESSD;          01580000 T  0003:010713
PRT(240) = PTOG
                    SPECTOG,                                                                        01581000 T  0003:010713
PRT(241) = SPECTOG
                         STOPENTRY,  COMMENT THIS MAKES THE ENTRY PROCEDURE ENTER ONLY              01582000 T  0003:010713
PRT(242) = STOPENTRY
                                             ONE ID AND THEN EIXT;                                  01583000 T  0003:010713
                         AJUMP;      COMMENT TELLS WHETHER A JUMP IS HANGING;                       01583100 T  0003:010713
PRT(243) = AJUMP
                    BOOLEAN STOPDEFINE;                                                             01584000 T  0003:010713
PRT(244) = STOPDEFINE
                    INTEGER MAXSAVE;                                                                01585000 T  0003:010713
PRT(245) = MAXSAVE
                         COMMENT THIS CONTAINS THE SIZE OF THE MAXIMUM SAVE ARRAY                   01586000 T  0003:010713
                         DECLARED. IT IS USED TO HELP DETERMINE STORAGE REQUIREMENTS                01587000 T  0003:010713
                         FOR THE PROGRAM PARAMETER BLOCK FOR THE OBJECT PROGRAM;                    01588000 T  0003:010713
                    REAL                                                                            01589000 T  0003:010713
                         KLASSF,     COMMENT CLASS IN LOW ORDER 7 BITS;                             01590000 T  0003:010713
PRT(246) = KLASSF
                         ADDRSF,     COMMENT ADDRESS IN LOW ORDER 11 BITS;                          01591000 T  0003:010713
PRT(247) = ADDRSF
                         LEVELF,     COMMENT LVL IN LOW ORDER 5 BITS;                               01592000 T  0003:010713
PRT(250) = LEVELF
                         LINKF,      COMMENT LINK IN LOW ORDER 13 BITS;                             01593000 T  0003:010713
PRT(251) = LINKF
                         INCRF,      COMMENT INCR CN LOW ORDER 8 BITS;                              01594000 T  0003:010713
PRT(252) = INCRF
                         PROINFO,    COMMENT CONTAINS ELBAT WORD FOR PROCEDURE BEING                01595000 T  0003:010713
PRT(253) = PROINFO
                                             DECLARED;                                              01596000 T  0003:010713
                                                                                                    01597000 T  0003:010713
                                                                                                    01598000 T  0003:010713
                                                                                                    01599000 T  0003:010713
                                                                                                    01600000 T  0003:010713
                                                                                                    01601000 T  0003:010713
                                                                                                    01602000 T  0003:010713
                                                                                                    01603000 T  0003:010713
                                                                                                    01604000 T  0003:010713
                                                                                                    01605000 T  0003:010713
                                                                                                    01606000 T  0003:010713
                                                                                                    01607000 T  0003:010713
                                                                                                    01608000 T  0003:010713
                                                                                                    01609000 T  0003:010713
```

```
                    G,          COMMENT GLOBAL TEMPORARY FOR BLOCK;                    01610000 T  0003:010:3
PRT(254) = G
                    TYPEV,      COMMENT USED TO CARRY CLASS OF IDENTIFIER              01611000 T  0003:010:3
PRT(255) = TYPEV
                                        BEING DECLARED;                                01612000 T  0003:010:3
                  PROADO,         COMMENT CONTAINS ADDRESS OF PROCEDURE BEING          01613000 T  0003:010:3
PRT(256) = PROADO
                                        DECLARED;                                      01614000 T  0003:010:3
                    MARK ,      COMMENT CONTAINS INDEX INTO INFO WHERE FIRST WORD     01615000 T  0003:010:3
PRT(257) = MARK
                                        OF ADDITIONAL INFO FOR A PROCEDURE ENTRY;      01616000 T  0003:010:3
                    PJ,         COMMENT FORMAL PARAMETER COUNTER;                      01617000 T  0003:010:3
PRT(260) = PJ
                     J,         COMMENT ARRAY COUNTER;                                 01618000 T  0003:010:3
PRT(261) = J
                    LASTINFO,   COMMENT INDEX TO LAST ENTRY IN INFO;                   01619000 T  0003:010:3
PRT(262) = LASTINFO
                    NEXTINFO,   COMMENT INDEX FOR NEXT ENTRYIN INFO;                   01620000 T  0003:010:3
PRT(263) = NEXTINFO
                    FIRSTX,     COMMENT RELATIVE ADD OF FIRST EXECUTABLE CODE          01621000 T  0003:010:3
PRT(264) = FIRSTX
                                        IN BLOCK,INITIALIZED TO 4095 EACH TIME;        01622000 T  0003:010:3
                    SAVEL;      COMMENT SAVE LOCATION FOR FIXUPS IN BLOCK;             01623000 T  0003:010:3
PRT(265) = SAVEL
                INTEGER NCII;   COMMENT THIS CONTAINS THE COUNT OF CONSTANTS           01624000 T  0003:010:3
PRT(266) = NCII
                                        ENTERED IN INFO AT ANY GIVEN TIME;             01625000 T  0003:010:3
                PROCEDURE UNHOOK;FORWARD;                                              01626000 T  0003:010:3
PRT(267) = UNHOOK
                PROCEDURE MAKEUPACCUM;FORWARD;                                         01627000 T  0003:011:0
PRT(270) = MAKEUPACCUM
                DEFINE PURPT=[4:8]#,SECRET=2#;                                         01628000 T  0003:011:0
                 COMMENT THESE DEFINES GIVE THE NAMES OF THE WORD MODE OPERATORS.  THE 01629000 T  0003:011:0
                    NUMBERS REFER TO THE APPROPRIATE SECTION OF THE PRODUCT SPECS.  THE 01630000 T  0003:011:0
                    FULL NAME IS ALSO GIVEN;                                           01631000 T  0003:011:0
                 DEFINE                                                                01632000 T  0003:011:0
                   ADD =  16#,    COMMENT (0101) 7.4.2.1    ADD;                      01633000 T  0003:011:0
                   BBC =  22#,    COMMENT (0131) 7.4.5.4    BRANCH BACKWARD CONDITIONAL; 01634000 T  0003:011:0
                   BBW = 534#,    COMMENT (4131) 7.4.5.2    BRANCH BACKWARD;          01635000 T  0003:011:0
                   BFC =  38#,    COMMENT (0231) 7.4.5.3    BRANCH FORWARD CONDITIONAL; 01636000 T  0003:011:0
                   BFW = 550#,    COMMENT (4231) 7.4.5.1    BRANCH FORWARD;           01637000 T  0003:011:0
                   CDC = 168#,    COMMENT (1241) 7.4.10.4   CONSTRUCT DESCRIPTOR CALL; 01638000 T  0003:011:0
                   CHS = 134#,    COMMENT (1031) 7.4.7.11   CHANGE SIGN;              01639000 T  0003:011:0
                   COC =  40#,    COMMENT (0241) 7.4.10.3   CONSTRUCT OPERAND CALL;    01640000 T  0003:011:0
                   COM = 130#,    COMMENT (1011) 7.4.10.5   COMMUNICATION OPERATOR;    01641000 T  0003:011:0
                   DEL =  10#,    COMMENT (0045) 7.4.9.3    DELETE;                   01642000 T  0003:011:0
                   DUP = 261#,    COMMENT (2025) 7.4.9.2    DUPLICATE;                01643000 T  0003:011:0
                   EQL = 581#,    COMMENT (4425) 7.4.4.3    EQUAL;                    01644000 T  0003:011:0
                   LBC = 278#,    COMMENT(2131) 7.4.5.9     GO BACKWARD CONDITIONAL;  01645000 T  0003:011:0
                   LBU = 790#,    COMMENT(6131) 7.4.5.7     GO BACKWARD (WORD);       01646000 T  0003:011:0
                   GEQ =  21#,    COMMENT (0125) 7.4.4.2    GREATER THAN OR EQUAL TO; 01647000 T  0003:011:0
                   LFC = 294#,    COMMENT(2231) 7.4.5.8     GO FORWARD CONDITIONAL;   01648000 T  0003:011:0
                    LFU = 806#,   COMMENT(6231) 7.4.5.6     GO FORWARD (WORD);        01649000 T  0003:011:0
                   GTR =  37#,    COMMENT (0225) 7.4.4.1    GREATER THAN;             01650000 T  0003:011:0
                   IDV = 384#,    COMMENT (3001) 7.4.2.5    INTEGER DIVIDE;           01651000 T  0003:011:0
                   INX =  24#,    COMMENT (0141) 7.4.10.2   INDEX;                    01652000 T  0003:011:0
                   ISD = 532#,    COMMENT (4121) 7.4.6.3    INTEGER STORE DESTRUCTIVE; 01653000 T  0003:011:0
```

```
  ISN = 548#,     COMMENT (4221) 7.4.6.4    INTEGER STORE NON-DESTRUCT;          01654000 T  0003:0112:0
  LEQ = 533#,     COMMENT (4125) 7.4.4.4    LESS THAN OR EQUAL TO;               01655000 T  0003:0112:0
  LND =  67#,     COMMENT (0415) 7.4.3.1    LOGICAL AND;                         01656000 T  0003:0112:0
  LNG =  19#,     COMMENT (0115) 7.4.3.4    LOGICAL NEGATE;                      01657000 T  0003:0112:0
  LOD = 260#,     COMMENT (2021) 7.4.10.1   LOAD OPERATOR;                       01658000 T  0003:0112:0
  LOR =  35#,     COMMENT (0215) 7.4.3.2    LOGICAL OR;                          01659000 T  0003:0112:0
  LQV = 131#,     COMMENT (1015) 7.4.3.3    LOGICAL EQIVALENCE;                  01660000 T  0003:0112:0
  LSS = 549#,     COMMENT (4225) 7.4.4.5    LESS THAN;                           01661000 T  0003:0112:0
  MKS =  72#,     COMMENT (0441) 7.4.8.1    MARK STACK;                          01662000 T  0003:0112:0
  MUL =  64#,     COMMENT (0401) 7.4.2.3    MULTIPLY;                            01663000 T  0003:0112:0
  NEQ =  69#,     COMMENT (0425) 7.4.4.6    NOT EQUAL TO;                        01664000 T  0003:0112:0
  NOP =  11#,     COMMENT (0055) 7.4.7.1    NO OPERATION;                        01665000 T  0003:0112:0
  PRL =  18#,     COMMENT (0111) 7.4.10.6   PROGRAM RELEASE;                     01666000 T  0003:0112:0
  PRTE=  12#,     COMMENT (0061) 7.4.10.9   EXTEND PRT;                          01667000 T  0003:0112:0
  RDV = 896#,     COMMENT (7001) 7.4.2.6    REMAINDER DIVIDE;                    01668000 T  0003:0112:0
  RTN =  39#,     COMMENT (0235) 7.4.8.3    RETURN NORMAL;                       01669000 T  0003:0112:0
  RTS = 167#,     COMMENT (1235) 7.4.8.4    RETURN SPECIAL;                      01670000 T  0003:0112:0
  SND = 132#,     COMMENT (1021) 7.4.6.2    STORE NON-DESTRUCTIVE;               01671000 T  0003:0112:0
  SSP = 582#,     COMMENT (4431) 7.4.7.10   SET SIGN PLUS;                       01672000 T  0003:0112:0
  STD =  68#,     COMMENT (0421) 7.4.6.1    STORE DESTRUCTIVE;                   01673000 T  0003:0112:0
  SUB =  48#,     COMMENT (0301) 7.4.2.2    SUBTRACT;                            01674000 T  0003:0112:0
  XCH = 133#,     COMMENT (1025) 7.4.9.1    EXCHANGE;                            01675000 T  0003:0112:0
  XIT =  71#,     COMMENT (0435) 7.4.8.2    EXIT;                                01676000 T  0003:0112:0
  ZP1 = 322#,     COMMENT (2411) 7.4.10.8   CONDITIONAL HALT;                    01677000 T  0003:0112:0
  SCI =1003#,     COMMENT (7655)            SCAN OUT INITIALIZE;                 01677050 T  0003:0112:0
  SAN =1004#,     COMMENT (7661)            SYSTEM ATTENTION NEEDED;             01677100 T  0003:0112:0
  SCS =1019#;     COMMENT (7755)            SCAN OUT STOP;                       01677150 T  0003:0112:0
COMMENT  THESE DEFINES ARE USED BY EMITD;                                      01678000 T  0003:0112:0
DEFINE                                                                         01679000 T  0003:0112:0
  DIA =  45#,     COMMENT (XX55) 7.4.7.1    DIAL A;                              01680000 T  0003:0112:0
  DIB =  49#,     COMMENT (XX61) 7.4.7.2    DIAL B;                              01681000 T  0003:0112:0
  TRB =  53#;     COMMENT (XX65) 7.4.7.3    TRANSFER BITS;                       01682000 T  0003:0112:0
REAL MAXSTACK,STACKCTR;                                                        01683000 T  0003:0112:0
PRT(271) = MAXSTACK
PRT(272) = STACKCTR
INTEGER MAXROW;                                                                01684000 T  0003:0112:0
PRT(273) = MAXROW
          COMMENT THIS CONTAINS THE MAXIMUM ROW SIZE OF ALL NON-SAVE           01685000 T  0003:0112:0
          ARRAYS DECLARED. ITS USE IS LIKE THAT OF MAXSAVE;                    01686000 T  0003:0112:0
INTEGER SEGSIZEMAX; COMMENT CONTAINS MAX SEGMENT SIZE;                         01687000 T  0003:0112:0
PRT(274) = SEGSIZEMAX
INTEGER F;                                                                     01688000 T  0003:0112:0
PRT(275) = F
          REAL NLO,NHI,TLO,THI;                                                01689000 T  0003:0112:0
PRT(276) = NLO
PRT(277) = NHI
PRT(300) = TLO
PRT(301) = THI
          BOOLEAN DPTOG;                                                       01690000 T  0003:0112:0
PRT(302) = DPTOG
          COMMENT  THE ABOVE THINGS ARE TEMP STORAGE FOR DOUBLE NOS;           01691000 T  0003:0112:0
INTEGER DOLLAR2TOG;                                                            01691500 T  0003:0112:0
PRT(303) = DOLLAR2TOG
DEFINE FZERO=896#;                                                             01692000 T  0003:0112:0
REAL T1,T2,N,K,AKKUM;                                                          01693000 T  0003:0112:0
PRT(304) = T1
PRT(305) = T2
```

```
PRT(306) = N
PRT(307) = K
PRT(310) = AKKUM
                BOOLEAN STOPGSP;                                                    01694000 T  0003:011210
PRT(311) = STOPGSP
                INTEGER BUP;                                                        01695000 T  0003:011210
PRT(312) = BUP
                BOOLEAN INLINETOG;                                                  01695500 T  0003:011210
PRT(313) = INLINETOG
                        COMMENT UNIQUE GLOBAL TEMP FOR BLOCK;                       01696000 T  0003:011210
                ARRAY GTA1[0:10];                                                   01697000 T  0003:011210
PRT(314) = GTA1
                 BOOLEAN ARRAY SPRT[0:31];                                          01698000 T  0003:011411
PRT(315) = SPRT
                    COMMENT  SPRT IS TO BE CONSIDERED TO BE AN ARRAY OF 32 32 BIT   01699000 T  0003:011712
                             FIELDS.  THE 32 BITS ARE IN THE LOW ORDER PART OF EACH 01700000 T  0003:011712
                             WORD.  THE BIT IS ON IF AND ONLY IF THE CORRESPONDING  01701000 T  0003:011712
                             PRT CELL HAS A PERMANENT ASSIGNMENT;                   01702000 T  0003:011712
                 INTEGER PRTI,PRTIMAX;                                              01703000 T  0003:011712
PRT(316) = PRTI
PRT(317) = PRTIMAX
                    COMMENT  PRTIMAX GIVES NEXT PRT CELL AVAILABLE FOR PERMANENT ASSIGN-
                             MENT.  PRTI GIVES NEXT PRT CELL POSSIBLY AVAILABLE FOR
                             TEMPORARY ASSIGNMENT;
                DEFINE ALPHASIZE = [12:6]#;  COMMENT ALPHASIZE IS THE DEFINE FOR THE BIT
                                             POSITION IN THE SECOND WORD OF INFO WHICH
                                             CONTAINS THE LENGTH OF ALPHA;
                DEFINE EDOCINDEX = L.[36:3],L.[39:7]#;  COMMENT EDOCINDEX IS THE WORD
                                                        PORTION OF L SPLIT INTO A ROW AND
                                                        COLUMN INDEX FOR EDOC;
                    DEFINE CPLUS1 = 769#;  COMMENT SEE COMMENT AT CPLUS2 DEFINE;
                    DEFINE CPLUS2 = 770#;  COMMENT CPLUS1 AND CPLUS2 ARE EXPLICIT CONSTANTS
                                           USED IN THE GENERATION OF C-RELATIVE CODE;
                     PROCEDURE FLAG(ERRNUM); VALUE ERRNUM; INTEGER ERRNUM; FORWARD;
PRT(320) = FLAG
                     ALPHA PROCEDURE B2D(B); VALUE B; REAL B; FORWARD;
PRT(321) = B2D
                     REAL PROCEDURE TAKE(W); VALUE W; INTEGER W; FORWARD;
PRT(322) = TAKE
                     BOOLEAN MACROID;
PRT(323) = MACROID
                     REAL PROCEDURE FIXDEFINEINFO(T); VALUE T; REAL T; FORWARD;
PRT(324) = FIXDEFINEINFO
                     PROCEDURE ERR (ERRNUM); VALUE ERRNUM; INTEGER ERRNUM; FORWARD;
PRT(325) = ERR
                     INTEGER PROCEDURE GIT(L); VALUE L; REAL L; FORWARD;
PRT(326) = GIT
                     ARRAY CALLA[0:31,0:255];
PRT(327) = CALLA
                     DEFINE CALL[CALL1]=CALLA[(GT3+CALL1),LINKR,GT3,LINKC]#;
                     REAL CALLX,CALLINFO,NESTCTR,NESTCUR;
PRT(330) = CALLX
PRT(331) = CALLINFO
PRT(332) = NESTCTR
PRT(333) = NESTCUR
                     ARRAY NESTPRT[PRTBASE:PRTOP];
PRT(334) = NESTPRT
                     ARRAY  SORTPRT[0:PRTOP-PRTBASE];
```

```
PRT(335) = SORTPRT

                COMMENT  "BLANKET" BLANKS OUT N+1 WORDS IN "THERE";                    01726000 T  0003:012:0
                STREAM PROCEDURE BLANKET(N,THERE); VALUE N;                           01737300 T  0003:012:0
PRT(336) = BLANKET                                                                    01737350 T  0003:012:0
                    BEGIN                                                             01737400 T  0003:012:0
                    DI:=THERE; DS:=8 LIT" "; SI:=THERE; DS:=N WDS;                    01737450 T  0003:012:2
                    END BLANKET;                                                      01737500 T  0003:012:2



                STREAM PROCEDURE CHANGESEQ(VAL,OLDSEQ); VALUE OLDSEQ;                 01741200 T  0003:012:3
PRT(337) = CHANGESEQ
                      BEGIN  DI:=OLDSEQ; SI:=VAL; DS:=8 DEC  END CHANGESEQ;           01741300 T  0003:012:3



                STREAM PROCEDURE SEQUENCEERROR(L);                                    01742100 T  0003:013:2
PRT(340) = SEQUENCEERROR
                    BEGIN  DI:=L; DS:=16 LIT"SEQUENCE ERROR  ";  END SEQUENCEERROR;   01742110 T  0003:013:2



                STREAM PROCEDURE GETVOID(VP,NCR,LCR,SEQ);  VALUE NCR,LCR;             01756000 T  0003:013:3
PRT(341) = GETVOID
    BEGIN                                                                             01757000 T  0003:013:3
    LABEL L,EXIT;                                                                     01758000 T  0003:013:0
    LOCAL N;                                                                          01759000 T  0003:013:0
    SI:=NCR; DI:=VP; DS:=8 LIT "0";                                                   01761000 T  0003:013:0
    2(34(IF SC=" " THEN SI:=SI+1 ELSE JUMP OUT 2 TO L));                              01762000 T  0003:013:3
    GO TO EXIT;                       % NO VOID RANGE GIVEN, RETURN ZERO.          01763000 T  0003:013:1
L:                                                                                    01764000 T  0003:013:1
    IF SC="%" THEN GO TO EXIT;     % STILL NO RANGE.                                 01764500 T  0003:013:1
    IF SC="'" THEN                                                                    01765000 T  0003:013:3
         BEGIN                                                                        01766000 T  0003:014:0
         SI:=SI+1; DI:=LCR; DS:=1 LIT"'"; % STOPPER FOR SCAN                          01767000 T  0003:014:0
         NCR:=SI; % TEMP. STORAGE, SINCE NCR IS "LOCAL" TO GETVOID.                   01768000 T  0003:014:3
         8(IF SC="'" THEN JUMP OUT ELSE                                               01769000 T  0003:014:0
              BEGIN  TALLY:=TALLY+1; SI:=SI+1 END);                                   01770000 T  0003:014:3
         END                                                                          01771000 T  0003:014:2
    ELSE BEGIN                                                                        01772000 T  0003:014:2
         NCR:=SI; % TEMP. STORAGE, SINCE NCR IS "LOCAL" TO GETVOID.                   01773000 T  0003:014:2
         DI:=LCR; DS:=1 LIT" "; % STOPPER FOR SCAN                                    01774000 T  0003:014:3
         8(IF SC=" " THEN JUMP OUT ELSE                                               01775000 T  0003:014:0
              BEGIN  TALLY:=TALLY+1; SI:=SI+1 END);                                   01776000 T  0003:014:3
         END;                                                                         01777000 T  0003:014:2
    SI:=NCR; DI:=VP; DI:=DI+8; % RESTORE POINTERS.                                    01780000 T  0003:014:2
    N:=TALLY; DI:=DI-N; DS:=N CHR;                                                    01781000 T  0003:014:3
EXIT:                                                                                 01782000 T  0003:015:2
    END OF GETVOID;                                                                   01784000 T  0003:015:2



REAL VOIDCR,VOIDPLACE,VOIDTCR,VOIDTPLACE;                                             01785000 T  0003:015:2
```

```
PRT(342) = VOIDCR
PRT(343) = VOIDPLACE
PRT(344) = VOIDTCR
PRT(345) = VOIDTPLACE
                FORMAT
                        BUG(X24,4(A4,X2));                                                  01800000 T  0003:015:2
                                                                                            01802000 T  0003:015:2
                                                                                          START OF SEGMENT **********    4
PRT(346) = BUG

                                                                                             4 IS    8 LONG, NEXT SEG    3
                PROCEDURE DATIME;                                                           01820000 T  0003:015:2
PRT(347) = DATIME
                    BEGIN
                    INTEGER H,MIN,Q; ALPHA N1,N2;                                           01821000 T  0003:015:2
                                                                                            01822000 T  0003:015:2
                                                                                          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(350) = DATER
                        BEGIN
                        DI:=LOC DATER; SI:=LOC DATE; SI:=SI+2;                              01824000 T  0005:000:0
                        2(DS:=2 CHR; DS:="/")); DS:=2 CHR;                                  01825000 T  0005:000:0
                        END OF DATER;                                                       01826000 T  0005:000:1
                                                                                            01827000 T  0005:0002:0
```

```
                    H:=TIME1 DIV 216000; MIN:=(TIME1 DIV 3600) MOD 60;                      01828000 T  0005:0003:2
                    N1:=DISK.MFID; N2:=DISK.FID;                                            01828500 T  0005:0007:2
                    WRITE(LINE,                                                             01829000 T  0005:0013:3
                        <X22,"BURROUGHS B-5700 ESPOL COMPILER MARK ",                       01830000 T  0005:0015:2
PRT(351) = *FORMAT DESCRIPTOR*
                "XVI,0,116"                                                         %117-   01831000 P  0006:0015:2
                        ,"  ",A6,"DAY, ",0,",  ",I2,"!",A2,X1,A3,                        01832000 T  0006:0015:2
                        ////X45,A1,A6,"/",A1,A6,/X45,15("=")//>,                          01832500 T  0006:0015:2
                                                                                          6 IS   40 LONG, NEXT SEG    5
                        TIME(6),DATER(TIME(5)),12×REAL(Q:=H MOD 12=0)+Q,                    01833000 T  0005:0016:0
PRT(352) = *LIST, LABEL, OR SEGMENT DESCRIPTOR*
                        Q:=MIN MOD 10+(MIN DIV 10)×64,                                      01834000 T  0005:0032:0
                        IF H≥12 THEN "PM," ELSE "AM,",                                      01835000 T  0005:0036:1
                        N1,[6:6],N1,N2,[6:6],N2);                                           01835500 T  0005:0043:2
PRT(353) = OUTPUT(W)
                    IF MERGETOG THEN % INDICATE NAME OF SOURCE FILE,                %111-   01835600 C  0005:0053:2
                      WRITE(LINE,<X40,"SOURCE FILE: ",A1,A6,"/",A1,A6,//>,          %111-   01835700 C  0005:0054:0
PRT(354) = *FORMAT DESCRIPTOR*
                                                                                          7 IS   14 LONG, NEXT SEG    5
                    (N1:=TAPE.MFID),[6:6],N1,(N2:=TAPE.FID),[6:6],N2);              %111-   01835800 C  0005:0057:2
PRT(355) = *LIST, LABEL, OR SEGMENT DESCRIPTOR*
                    NOHEADING:=FALSE;                                                       01836000 T  0005:0077:2
                    END OF DATIME;                                                          01837000 T  0005:0078:0
                                                                                          5 IS   82 LONG, NEXT SEG    3


        COMMENT  THIS SECTION CONTAINS ALL CODE PERTAINENT TO READING CARDS             02000000 T  0003:015:2
```

```
                    AND SCANNING THEM;
                  %*********************************************************************
                  %                                                                    %110-   02001000 T  0003:0151:2
                  %       MISCELLANEOUS CROSS REFERENCE PROCEDURES                     %110-   02001605 C  0003:0151:2
                  %                                                                    %110-   02001610 C  0003:0151:2
                  %*********************************************************************   02001615 C  0003:0151:2
                  %                                                                    %110-   02001620 C  0003:0151:2
                  PROCEDURE CROSSREFIT(INDEX,SEQNO,REFTYPE);                           %110-   02001630 C  0003:0151:2
                                                                                       %110-   02001635 C  0003:0151:2
PRT(356) = CROSSREFIT                                                                  %110-   02001640 C  0003:0151:2
                    VALUE INDEX,SEQNO,REFTYPE;                                         %110-   02001645 C  0003:0151:2
                    REAL  INDEX,SEQNO,REFTYPE;                                         %110-   02001650 C  0003:0151:2
                  BEGIN                                                                %110-   02001655 C  0003:0151:2
                   IF XREFINFO[INDEX].IDNOF # 0 THEN % SAVE                            %110-   02001660 C  0003:0151:2
                   BEGIN                                                               %110-   02001665 C  0003:0155:3
                    IF XREFPT > 29 THEN % NO SLOTS LEFT IN ARRAY, WRITE IT OUT,        %110-   02001670 C  0003:0156:0
                      BEGIN                                                            %110-   02001675 C  0003:0157:2
                        WRITE(DSK2,30,XREFAY2[*]);                                     %110-   02001680 C  0003:0157:3
                        XREFPT := 0;                                                   %110-   02001685 C  0003:0161:3
                      END;                                                             %110-   02001690 C  0003:0162:1
                    XREFAY2[XREFPT] := SEQNO & REFTYPE TYPEREF & XREFINFO[INDEX]       %110-   02001695 C  0003:0162:1
                                                 REFIDNOF;                             %110-   02001700 C  0003:0166:1
                    XREFPT := XREFPT + 1; % EVEN THOUGH THE ARRAY MAY BE FULL NOW WE   %110-   02001705 C  0003:0168:0
                                          % CANT WRITE IT OUT BECAUSE SOME ROUTINES    %110-   02001710 C  0003:0169:3
                                          % WILL LOOK BACK AT THE ENTRY WE JUST PUT    %110-   02001715 C  0003:0169:3
                                          % IN AND FIX IT UP.                          %110-   02001720 C  0003:0169:3
                   END;                                                                %110-   02001725 C  0003:0169:3
                  END OF CROSSREFIT;                                                   %110-   02001730 C  0003:0169:3
```

```
                  %                                                                    %110-   02001735 C  0003:0169:3
                  PROCEDURE CROSSREFDUMP(INDEX);                                       %110-   02001740 C  0003:0169:3
PRT(357) = CROSSREFDUMP                                                                %110-   02001745 C  0003:0169:3
                    VALUE INDEX;                                                       %110-   02001750 C  0003:0169:3
                    REAL  INDEX;                                                       %110-   02001755 C  0003:0169:3
                  BEGIN                                                                %110-   02001760 C  0003:0169:3
                    STREAM PROCEDURE MOVEXREFINFO(S,D,N);                              %110-        START OF SEGMENT **********    8
PRT(360) = MOVEXREFINFO
                      VALUE N;                                                         %110-   02001765 C  0008:0000:0
                    BEGIN                                                              %110-   02001770 C  0008:0000:0
                      SI := D; DI := D; DS := 8 LIT " "; DS := 7 WDS; % BLANK RECORD   %110-   02001775 C  0008:0000:0
                      SI := S; SI := SI + 3; DI := D; DS := N CHR; % MOVE IDENTIFIER   %110-   02001780 C  0008:0002:0
                    END OF MOVEXREFINFO;                                               %110-   02001785 C  0008:0003:2
```

```
                  %                                                                    %110-   02001790 C  0008:0003:3
                  IF XREFINFO[INDEX].IDNOF # 0 THEN % DUMP IT                          %110-   02001795 C  0008:0003:3
                  BEGIN                                                                %110-   02001800 C  0008:0007:3
                   MOVEXREFINFO(INFO[INDEX.LINKR,INDEX.LINKC+1],XREFAY1[*],            %110-   02001805 C  0008:0008:0
                                TAKE(INDEX+1),[12:6]);                                 %110-   02001810 C  0008:0011:3
                   XREFAY1[8] := XREFINFO[INDEX];                                      %110-   02001815 C  0008:0013:3
                   XREFAY1[9] := TAKE(INDEX); % FLBAT WORD                             %110-   02001820 C  0008:0017:3
                   WRITE(DSK1,10,XREFAY1[*]);                                          %110-   02001821 C  0008:0019:2
```

```
                XREFINFO[INDEX] := 0;                                                %110-      02001822 C  0008:0023:3
              END;                                                                   %110-      02001825 C  0008:0027:2
            END OF CROSSREFDUMP;                                                     %110-      02001830 C  0008:0027:2
                                                                                                  8 IS   28 LONG, NEXT SEG    3



            COMMENT  OCTIZE REFORMATS ACCUM FOR OCTAL CONSTANTS;                                  02001836 T  0003:0169:3
            BOOLEAN STREAM PROCEDURE OCTIZE(S,D,SKP,CNT); VALUE SKP,CNT;                          02001838 T  0003:0169:3
PRT(361) = OCTIZE
                BEGIN
                S1:=S; S1:=SI+4; DI:=D; SKP(DS:=3 RESET); % RIGHT JUSTIFY.                        02001840 T  0003:0169:3
                CNT(IF SC≥"8" THEN TALLY:=1 ELSE IF SC<"0" THEN TALLY:=1; SKIP 3 SB;              02001842 T  0003:0170:0
                    3(IF SB THEN DS:=SET ELSE DS:=RESET; SKIP SB));                               02001844 T  0003:0172:0
                S1:=D; IF SB THEN                                                                 02001846 T  0003:0175:2
                     BEGIN  TALLY:=1; DI:=D; DS:=RESET   END; % PREVENT FLAG BIT.                 02001848 T  0003:0177:3
                OCTIZE:=TALLY; % "1" = NON OCTAL CHARACTER OR FLAG BIT.                           02001850 T  0003:0178:0
                END OCTIZE;                                                                       02001852 T  0003:0179:3
                                                                                                  02001854 T  0003:0179:3



            COMMENT  HEXIZE REFORMATS ACCUM FOR HEXADECIMAL CONSTANTS;                            02001856 T  0003:0180:1
            BOOLEAN STREAM PROCEDURE HEXIZE(S,D,SKP,CNT); VALUE SKP,CNT;                          02001858 T  0003:0180:1
PRT(362) = HEXIZE
                BEGIN  LOCAL T1,T2,TEMP2,TEMP1; LABEL AGIN;                                       02001860 T  0003:0180:1
              COMMENT  LOCAL VARIABLES ARE LOCATED IN REVERSE ORDER FROM THE                      02001862 T  0003:0181:2
                       WAY THEY ARE DECLARED IN STREAM PROCEDURES;                                02001864 T  0003:0181:2
                DI:=LOC TEMP1; CNT(DS:=LIT"1"); % IN CASE A CHAR=A,B,C,D,OR F.                    02001866 T  0003:0181:2
                S1:=S; S1:=SI+3; DI:=LOC TEMP1; % WE MAY OVERFLOW INTO TEMP2.                     02001868 T  0003:0183:2
                CNT(IF SC<"0" THEN IF SC≥"A" THEN IF SC≤"F" THEN % WORK HARD.                     02001870 T  0003:0183:3
                     BEGIN                                                                        02001872 T  0003:0187:2
                     T1:=SI; T2:=DI; DI:=T1; SI:=T2; % FLIP, MAN.                                 02001874 T  0003:0187:2
                     DS:=3 RESET; SI:=T1; DI:=T2; % FLIP BACK.                                    02001876 T  0003:0188:1
                     DS:=1 ADD; DI:=DI-1; SKIP 2 DB; DS:=1 SET; SKIP 3 DB;                        02001878 T  0003:0189:3
                     GO AGIN;                                                                     02001880 T  0003:0190:1
                     END;                                                                         02001882 T  0003:0191:2
                     IF SC<"0" THEN TALLY:=1; DS:=CHR; % < 0 = NON-HEX CHARACTER.                 02001884 T  0003:0191:2
              AGIN;                                                                               02001886 T  0003:0192:0
                 );                                                                               02001888 T  0003:0192:0
                S1:=LOC TEMP1; DI:=D; SKP(DS:=4 RESET); % RIGHT ADJUST CONSTANT.                   02001890 T  0003:0192:0
                CNT(SKIP 2 SB;                                                                    02001892 T  0003:0194:0
                  4(IF SB THEN DS:=SET ELSE DS:=RESET; SKIP SB));% FINAL CONVERT.                  02001894 T  0003:0195:3
                S1:=D; IF SB THEN                                                                 02001895 T  0003:0197:3
                     BEGIN  TALLY:=1; DI:=D; DS:=RESET   END; % PREVENT FLAG BIT.                 02001896 T  0003:0198:1
                HEXIZE:=TALLY; % "1" IF PROGRAMMER GOOFED.                                        02001897 T  0003:0199:3
                END HEXIZE;                                                                       02001898 T  0003:0200:0



            COMMENT  PUTSEQNO PUTS THE SEQUENCE NUMBER OF THE CARD-IMAGE                          02002000 T  0003:0201:2
                     CURRENTLY BEING SCANNED INTO THE INFO TABLE IN CASE                          02003000 T  0003:0201:2
                     IT IS NEEDED FOR FUTURE REFERENCE;                                           02004000 T  0003:0201:2
            STREAM PROCEDURE PUTSEQNO(INFO,LCR); VALUE LCR;                                       02005000 T  0003:0201:2
PRT(363) = PUTSEQNO
```

```
                BEGIN  DI:=INFO; SI:=LCR; DS:=WDS;  END PUTSEQNO;                                02006000 T  0003:020:2


            COMMENT  TURNONSTOPLIGHT TURNS THE LIGHT "RED" ON THE "CORNER",                     02007000 T  0003:020:0
                     I.E., THE PURPOSE OF THIS ROUTINE IS TO INSERT A PER-                      02008000 T  0003:020:0
                     CENT SIGN IN COLUMN 73 AS AN END OF CARD SENTINEL FOR                      02009000 T  0003:020:0
                     THE SCANNER;                                                               02010000 T  0003:020:0
            STREAM PROCEDURE TURNONSTOPLIGHT(RED,CORNER); VALUE RED,CORNER;                     02011000 T  0003:020:0
PRT(364) = TURNONSTOPLIGHT
                BEGIN  DI:=CORNER; SI:=LOC CORNER; SI:=SI-1; DS:=CHR  END;                      02012000 T  0003:020:0


               COMMENT WRITNEW TRANSFERS THE CARD IMAGE TO THE NEWTAPE BUFFER                   02014000 T  0003:020:2
                       AND REPORTS IF THE CARD MIGHT BE CONTROL CARD;                           02015000 T  0003:020:2
             BOOLEAN STREAM PROCEDURE WRITNEW(NEW,FCR); VALUE FCR;                              02016000 T  0003:020:2
PRT(365) = WRITNEW
                    BEGIN SI ← FCR; IF SC ≠ "$" THEN TALLY ← 1;                                 02017000 T  0003:020:2
                    DI←NEW;DS←10 WDS;                                                           02018000 T  0003:020:2
                          WRITNEW ← TALLY END WRITNEW;                                          02020000 T  0003:020:3


                  COMMENT MKABS CONVERTS A DESCRIPTOR  TO AN ABSOLUTE ADDRESS;                  02021000 T  0003:020:1
                REAL STREAM PROCEDURE MKABS(A);                                                 02022000 T  0003:020:1
PRT(366) = MKABS
                        BEGIN DI ← A; MKABS ← DI END MKABS;                                     02023000 T  0003:020:1


            REAL STREAM PROCEDURE CONV(ACCUM,SKP,N);VALUE SKP,N;                                02041000 T  0003:020:1
PRT(367) = CONV
                        BEGIN                                                                   02042000 T  0003:020:1
                             SI← ACCUM; SI←SI+SKP;SI←SI+3;DI←LOC CONV;DS←N OCT                   02043000 T  0003:020:2
                        END;                                                                    02044000 T  0003:021:1


             STREAM PROCEDURE  MOVECHARACTERS(N,SORCE,SSKIP,DEST,DSKIP);                        02045000 T  0003:021:3
PRT(370) = MOVECHARACTERS
                             VALUE  N,SSKIP,DSKIP;                                              02046000 T  0003:021:3
                             BEGIN                                                              02047000 T  0003:021:3
                             SI←SORCE ; DI←DEST;                                                02048000 T  0003:021:0
                             SI←SI+SSKIP; DI← DI+DSKIP ;                                        02049000 T  0003:021:1
                             DS ← N CHR ;                                                       02050000 T  0003:021:3
                             END ;                                                              02051000 T  0003:021:0


            COMMENT  MOVECHARACTERS MOVES  N CHARACTERS FROM THE SSKIP-TH CHAR IN               02052000 T  0003:021:0
```

```
                        "SORCE" TO THE DSKIP-TH CHAR IN "DEST".          )                02053000 T  0003:021:0
                STREAM PROCEDURE MOVE(W)"WORDS FROM"(A)"TO"(B); VALUE W;                 02054000 T  0003:021:0
PRT(371) = MOVE
                        BEGIN SI ← A; DI ← B; DS ← W WDS END;                            02055000 T  0003:021:0



                STREAM PROCEDURE RESIZE(FIEL);                                           02056000 T  0003:021:0
PRT(372) = RESIZE
                BEGIN LOCAL T;                                                           02057000 T  0003:021:0
                 SI←FIFL; DI←LOC T; DS←WDS;                                              02058000 T  0003:021:2
                SI←T;DI←FIEL;DI←DI+1; SKIP 2 DB; DS←10 SET                               02059000 T  0003:021:3
                END;                                                                     02060000 T  0003:021:2



              COMMENT  EQUAL COMPARES COUNT CHARACTERS LOCATED AT A AND B FOR            02061000 T  0003:021:2
                       EQUALITY.  THIS ROUTINE IS USED IN THE LOOK-UP OF ALPHA           02061500 T  0003:021:2
                       QUANTITIES IN THE DIRECTORY;                                      02062000 T  0003:021:2
              BOOLEAN STREAM PROCEDURE EQUAL(COUNT,A,B); VALUE COUNT;                    02062500 T  0003:021:2
PRT(373) = EQUAL
                  BEGIN                                                                  02063000 T  0003:021:2
                  TALLY:=1; SI:=A; DI:=B;                                                02063500 T  0003:022:0
                  IF COUNT SC=DC THEN EQUAL:=TALLY                                       02064000 T  0003:022:1
                  END EQUAL;                                                             02064500 T  0003:022:3
```

```
                  PROCEDURE READACARD;  FORWARD;                                         02065000 T  0003:022:1
PRT(374) = READACARD
                  PROCEDURE DOLLARCARD; FORWARD;                                         02065500 T  0003:022:1
PRT(375) = DOLLARCARD
                  BOOLEAN PROCEDURE BOOLEXP; FORWARD;                                    02065600 T  0003:022:1
PRT(376) = BOOLEXP
                  PROCEDURE SCANNER;                                                     02066000 T  0003:022:1
PRT(377) = SCANNER
                      BEGIN                                                              02066500 T  0003:022:1
                  COMMENT  "SCAN" IS THE STREAM PROCEDURE WHICH DOES THE ACTUAL SCANNING.  02067000 T  0003:022:1
                           IT IS DRIVEN BY A SMALL WORD MODE PROCEDURE CALLED "SCANNER",   02067500 T  0003:022:1
                           WHICH CHECKS FOR A QUANTITY BEING BROKEN ACROSS A CARD. "SCAN"  02068000 T  0003:022:1
                           IS CONTROLLED BY A VARIABLE CALLED "RESULT". "SCAN" ALSO        02068500 T  0003:022:1
                           INFORMS THE WORLD OF ITS ACTION BY MEANS OF THE SAME VARIABLE.  02069000 T  0003:022:1
                           HENCE THE VARIABLE "RESULT" IS PASSED BY BOTH NAME AND VALUE.   02069500 T  0003:022:1
                           THE MEANING OF "RESULT" AS INPUT IS:                            02070000 T  0003:022:1
                              VALUE     MEANING                                          02070500 T  0003:022:1
                              =====     ========================================         02071000 T  0003:022:1
                                0       INITIAL CODE - DEBLANK AND START TO FETCH THE    02071500 T  0003:022:1
                                        NEXT QUANTITY.                                   02072000 T  0003:022:1
                                1       CONTINUE BUILDING AN IDENTIFIER (INTERRUPTED BY  02072500 T  0003:022:1
                                        END-OF-CARD BREAK).                              02073000 T  0003:022:1
                                2       LAST QUANTITY BUILT WAS SPECIAL CHARACTER. HENCE, 02073500 T  0003:022:1
                                        EXIT (INTERRUPTION BY END-OF-CARD BREAK IS NOT   02074000 T  0003:022:1
                                        IMPORTANT).                                      02074500 T  0003:022:1
                                3       CONTINUE BUILDING A NUMBER (INTERRUPTED BY END-OF-02075000 T  0003:022:1
```

```
                CARD BREAK).                                                    02075500 T  0003:022:1
      4         LAST THING WAS AN ERROR (COUNT EXCEEDED 63). HENCE,             02076000 T  0003:022:1
                EXIT (INTERRUPTION BY END-OF-CARD BREAK NOT                     02076500 T  0003:022:1
                IMPORTANT).                                                     02077000 T  0003:022:1
      5         GET NEXT CHARACTER AND EXIT.                                    02077500 T  0003:022:1
      6         SCAN A COMMENT.                                                 02078000 T  0003:022:1
      7         DEBLANK ONLY.                                                   02078500 T  0003:022:1
THE MEANING OF "RESULT" AS OUTPUT IS:                                           02079000 T  0003:022:1
   VALUE    MEANING                                                             02079500 T  0003:022:1
   =====    ========================================                            02080000 T  0003:022:1
      1     AN IDENTIFIER WAS BUILT.                                            02080500 T  0003:022:1
      2     A SPECIAL CHARACTER WAS OBTAINED.                                   02081000 T  0003:022:1
      3     A NUMBER (INTEGER) WAS BUILT.                                       02081500 T  0003:022:1
"SCAN" PUTS ALL STUFF SCANNED (EXCEPT FOR COMMENTS AND                          02082000 T  0003:022:1
DISCARDED BLANKS) INTO "ACCUM" (CALLED "ACCUMULATOR"                            02082500 T  0003:022:1
FOR THE REST OF THIS DISCUSSION).                                               02083000 T  0003:022:1
"COUNT" IS THE VARIABLE THAT GIVES THE NUMBER OF CHARACTERS                     02083500 T  0003:022:1
"SCAN" HAS PUT INTO THE "ACCUMULATOR".  SINCE "SCAN" NEEDS                      02084000 T  0003:022:1
THE VALUE SO THAT IT CAN PUT MORE CHARACTERS INTO THE "ACCUM-                   02084500 T  0003:022:1
ULATOR" AND NEEDS TO UPDATE "COUNT" FOR THE OUTSIDE WORLD.                      02085000 T  0003:022:1
"COUNT" IS PASSED BY BOTH NAME AND VALUE.  IT IS ALSO                           02085500 T  0003:022:1
CONVENIENT TO HAVE (63-COUNT). THIS IS CALLED "COMCOUNT".                       02086000 T  0003:022:1
"NCR" (NEXT CHARACTER TO BE SCANNED) IS ALSO PASSED BY                          02086500 T  0003:022:1
NAME AND VALUE SO THAT IT MAY BE UPDATED.                                       02087000 T  0003:022:1
"ST1" AND "ST2" ARE TEMPORARY STORAGES WHICH ARE EXPLICITLY                     02087500 T  0003:022:1
PASSED TO "SCAN" IN ORDER TO OBTAIN THE MOST USEFULL STACK                      02088000 T  0003:022:1
ARRANGEMENT.                                                                    02088500 T  0003:022:1
*/                                                                              02089000 T  0003:022:1
STREAM PROCEDURE SCAN(NCR,COUNTV,ACCUM,COMCOUNT,RESULT,RESULTV,                 02089500 T  0003:022:1
                                                    START OF SEGMENT **********     9
PRT(400) = SCAN
                        COUNT,ST2,NCRV,ST1);                                    02090000 T  0009:000:0
                    VALUE COUNTV, COMCOUNT,RESULTV,ST2,NCRV,ST1;                02090500 T  0009:000:0
            BEGIN                                                               02091000 T  0009:000:0
            LABEL DEBLANK,NUMBERS,IDBLDR,GNC,K,EXIT,FINIS,L,ERROR,              02091500 T  0009:000:0
                  COMMENTS,COMMANTS;                                            02092000 T  0009:000:0
            DI:=RESULT; DI:=DI+7; SI:=NCRV;                                     02092500 T  0009:000:0
COMMENT     SETUP "DI" FOR A CHANGE IN "RESULT" AND "SI" FOR A LOOK AT          02093000 T  0009:000:1
            THE BUFFER;                                                         02093500 T  0009:000:1
            CI:=CI+RESULTV;   % SWITCH ON VALUE OF RESULT;                      02094000 T  0009:000:1
            GO DEBLANK;  % 0 IS INITIAL CODE.                                   02094500 T  0009:000:2
            GO IDBLDR;   % 1 IS ID CODE.                                        02095000 T  0009:000:3
            GO FINIS;    % 2 IS SPECIAL CHARACTER CODE.                         02095500 T  0009:000:3
            GO NUMBERS;  % 3 IS NUMBER CODE.                                    02096000 T  0009:000:0
            GO FINIS;    % 4 IS ERROR CODE.                                     02096500 T  0009:000:0
            GO GNC;      % 5 IS GET NEXT CHARACTER CODE.                        02097000 T  0009:000:1
            GO COMMANTS; % 6 IS COMMENT CODE.                                   02097500 T  0009:000:1
                         % 7 IS DEBLANK ONLY CODE.                             02098000 T  0009:000:2
            IF SC=" " THEN                                                      02098500 T  0009:000:2
K:              BEGIN  SI:=SI+1; IF SC=" " THEN GO K  END;                      02099000 T  0009:000:3
            GO FINIS;                                                           02099500 T  0009:000:2
DEBLANK:                                                                        02100000 T  0009:000:2
            IF SC=" " THEN                                                      02100500 T  0009:000:2
L:              BEGIN  SI:=SI+1; IF SC=" " THEN GO L  END;                      02101000 T  0009:000:1
COMMENT     IF WE ARRIVE HERE WE HAVE A NON-BLANK CHARACTER;                    02101500 T  0009:000:0
            NCRV:=SI;                                                           02102000 T  0009:000:0
            IF SC ≥ "0" THEN GO NUMBERS;                                        02102500 T  0009:000:0
```

```
          IF SC=ALPHA THEN GO TDBLDR;                                          02103000 T  0009:0009:2
COMMENT   IF WE ARRIVE HERE WE HAVE A SPECIAL CHARACTER (OR GNC);              02103500 T  0009:0009:3
GNC:                                                                           02104000 T  0009:0009:3
          DS:=LIT"2"; TALLY:=1; SI:=SI+1; GO EXIT;                             02104500 T  0009:0009:3
COMMANTS:                                                                      02105000 T  0009:0011:2
          IF SC#";" THEN                                                       02105500 T  0009:0011:2
               BEGIN                                                           02106000 T  0009:0012:1
COMMENTS:                                                                      02106500 T  0009:0012:1
               SI:=SI+1;                                                       02107000 T  0009:0013:2
               IF SC > "%" THEN GO COMMENTS;                                   02107500 T  0009:0013:2
               IF SC < ";" THEN GO COMMENTS;                                   02108000 T  0009:0014:0
COMMENT   CHARACTERS BETWEEN % AND SEMICOLON ARE HANDLED BY WORD-              02108500 T  0009:0014:1
          MODE PART OF COMMENT ROUTINE;                                        02109000 T  0009:0014:1
               END;                                                            02109500 T  0009:0014:1
          GO FINIS;                                                            02110000 T  0009:0014:1
IDBLDR:                                                                        02110500 T  0009:0015:2
          TALLY:=63; DS:=LIT "1";                                              02111000 T  0009:0015:2
          COMCOUNT(TALLY:=TALLY+1;                                             02111500 T  0009:0015:3
                    IF SC=ALPHA THEN SI:=SI+1 ELSE JUMP OUT TO EXIT);          02112000 T  0009:0017:2
          TALLY:=TALLY+1;                                                      02112500 T  0009:0018:1
          IF SC=ALPHA THEN                                                     02113000 T  0009:0019:2
               BEGIN                                                           02113500 T  0009:0019:3
ERROR:                                                                         02114000 T  0009:0019:3
               DI:=DI-1; DS:=LIT "4"; GO EXIT;                                 02114500 T  0009:0020:0
               END                                                             02115000 T  0009:0021:2
          ELSE GO EXIT;                                                        02115500 T  0009:0021:2
COMMENT   IF WE ARRIVE AT ERROR WE HAVE MORE THAN 63 CHARACTERS                02116000 T  0009:0021:3
          IN AN IDENTIFIER OR NUMBER;                                          02116500 T  0009:0021:3
               NUMBERS:                                                        02117000 T  0009:0021:3
               TALLY:=63; DS:=LIT "3";                                         02117500 T  0009:0021:3
PRT(401) = *LIST, LABEL, OR SEGMENT DESCRIPTOR*
               COMCOUNT(TALLY:=TALLY+1;                                        02118000 T  0009:0022:1
                         IF SC <"0"THEN JUMP OUT TO EXIT; SI:=SI+1);           02118500 T  0009:0024:0
               GO ERROR;                                                       02119000 T  0009:0025:3
               EXIT:                                                           02119500 T  0009:0025:3
               ST1:=TALLY; % "ST1" CONTAINS NUMBER OF CHARACTERS WE ARE        02120000 T  0009:0025:3
                           % GOING TO MOVE INTO THE "ACCUMULATOR",             02120500 T  0009:0026:0
               TALLY:=TALLY+COUNTV; ST2:=TALLY;                                02121000 T  0009:0026:0
               DI:=COUNT; SI:=LOC ST2; DS:=WDS;                                02121500 T  0009:0027:2
               COMMENT  THIS CODE UPDATED "COUNT";                             02122000 T  0009:0027:3
               DI:=ACCUM; SI:=SI-3; DS:=3 CHR;                                 02122500 T  0009:0027:3
               COMMENT  THIS CODE PLACES "COUNT" IN "ACCUM" AS WELL;           02123000 T  0009:0028:1
               DI:=DI+COUNTV; % POSITION "DI" PAST CHARACTERS ALREADY          02123500 T  0009:0028:1
                              % IN THE "ACCUMULATOR", IF ANY.                  02124000 T  0009:0029:2
               SI:=NCRV; DS:=ST1 CHR;                                          02124500 T  0009:0029:2
               COMMENT  MOVE CHARACTERS INTO "ACCUM";                          02125000 T  0009:0029:3
               FINIS:                                                          02125500 T  0009:0029:3
               DI:=NCR; ST1:=SI; SI:=LOC ST1; DS:=WDS;                         02126000 T  0009:0029:3
PRT(402) = *LIST, LABEL, OR SEGMENT DESCRIPTOR*
               COMMENT  RESET "NCR" TO LOCATION OF NEXT CHARACTER TO BE SCANNED; 02126500 T  0009:0031:2
               END OF SCAN;                                                    02127000 T  0009:0031:2

               LABEL L;%                                                       02127500 T  0009:0031:2
               L:                                                              02128000 T  0009:0031:2
```

```
                    SCAN(NCR,COUNT,ACCUM[1],63-COUNT,RESULT,                                02128500 T  0009:003:0
                         RESULT,COUNT,0,NCR,0);                                             02129000 T  0009:003:4:1
                    IF NCR=LCR THEN                                                         02129500 T  0009:003:6:0
                         BEGIN                                                              02130000 T  0009:003:7:2
                         READACARD;                                                         02130500 T  0009:003:7:3
                         GO TO L; % GO DIRECTLY TO L, DO NOT PASS GO,                       02135500 T  0009:003:8:0
                                  % DO NOT COLLECT $200,                                    02136000 T  0009:003:8:1
                         END;                                                               02136500 T  0009:003:8:1
                    END SCANNER;                                                            02137000 T  0009:003:8:1
                                                                                  9 IS   39 LONG, NEXT SEG    3


               DEFINE WRITELINE = IF SINGLTOG THEN WRITE(LINE,15,LIN[*])                    02181000 T  0003:022:1:1
                                    ELSE WRITE(LINE[DBL],15,LIN[*])#,                       02181250 T  0003:022:1:1
                      PRINTCARD = BEGIN                                                     02182500 T  0003:022:1:1
                                  EDITLINE(LIN,FCR,L DIV 4,L,[46:2],MEDIUM,OMITTING);       02182750 T  0003:022:1:1
                                  IF NOHEADING THEN DATIME; WRITELINE;                      02183000 T  0003:022:1:1
                                  END #;                                                    02183250 T  0003:022:1:1
               STREAM PROCEDURE EDITLINE(LINE,NCR,R,L,SYMBOL,OMIT);                         02183500 T  0003:022:1:1
PRT(403) = EDITLINE
                                       VALUE  NCR,R,L,SYMBOL,OMIT;                          02183750 T  0003:022:1:1
                    BEGIN                                                                   02184000 T  0003:022:1:1
                    DI := LINE; DS := 16 LIT " ";                                           02184250 T  0003:022:3:2
                    SI := NCR; DS := 9 WDS;                                                 02184500 T  0003:022:5:3
                    DS := 8 LIT " ";                                                        02184750 T  0003:022:6:0
                    DS := WDS; % SEQUENCE NUMBER,                                           02185000 T  0003:022:7:2
                    DS:=LIT" "; SI:=LOC SYMBOL; SI:=SI+6;                                   02185250 T  0003:022:7:3
                    DS:=2 CHR; DS:=LIT" ";                                                  02185500 T  0003:022:8:1
                    SI←LOC R;  DS←4 DEC;  DS←LIT ":";                                       02185750 T  0003:022:9:2
                    SI←LOC L;  DS←1 DEC;                                                    02186000 T  0003:023:0:0
                    DS←6 LIT " ";                                                           02186250 T  0003:023:0:1
                    OMIT(DI:=DI-12; DS:=8 LIT"    OMIT");                                   02186750 T  0003:023:1:3
                    END EDITLINE;                                                           02187000 T  0003:023:4:1


               COMMENT  COMPARE COMPARES SEQUENCE NUMBERS OF TAPE AND CARD. IF              02187250 T  0003:023:4:1
                        TAPE IS SMALLER THEN RESULT = 0 ELSE IF CARD IS SMALLER             02187500 T  0003:023:4:1
                        RESULT = 1 ELSE RESULT = 2;                                         02187750 T  0003:023:4:1
               REAL STREAM PROCEDURE COMPARE(TAPE,CARD); VALUE TAPE,CARD;                   02188000 T  0003:023:4:1
PRT(404) = COMPARE
                    BEGIN                                                                   02188250 T  0003:023:4:1
                    SI := TAPE; DI := CARD;                                                 02188500 T  0003:023:5:2
                    IF 8 SC ≥ DC THEN                                                       02188750 T  0003:023:5:3
                         BEGIN                                                              02189000 T  0003:023:6:0
                         SI := SI-8; DI := DI-8; TALLY := 1;                                02189250 T  0003:023:6:0
                         IF 8 SC = DC THEN TALLY := 2                                       02189500 T  0003:023:7:2
                         END;                                                               02189750 T  0003:023:7:3
                    COMPARE := TALLY                                                        02190000 T  0003:023:8:0
                    END COMPARE;                                                            02190250 T  0003:023:8:0


               PROCEDURE OUTPUTSOURCE;                                                      02190500 T  0003:023:9:2
```

```
PRT(405) = OUTPUTSOURCE
                    BEGIN
                    LABEL LCARD,LTAPE,AWAY;

                    SWITCH SW:=LCARD,LCARD,LTAPE,AWAY,LCARD,LTAPE;
                    IF SEQTOG THEN % RESEQUENCING.
                         BEGIN
                         IF TOTALNO = -10 OR NEWBASE THEN
                              BEGIN
                              NEWBASE := FALSE; GTI1:= TOTALNO:=BASENUM
                              END
                         ELSE GTI1:= TOTALNO:= TOTALNO + ADDVALUE;
                         CHANGESEQ(GTI1,LCR);
                         END;
                    IF NEWTOG THEN
                      IF WRITNEW(LIN,FCR) THEN WRITE(NEWTAPE,10,LIN[*]);
                    IF OMITTING THEN IF NOT LISTATOG THEN GO AWAY;
                    GO SW[LASTUSED];
               LCARD:
                    IF LISTER OR LISTPTOG THEN PRINTCARD;
                    GO AWAY;
               LTAPE:
                    IF LISTER THEN PRINTCARD;
               %    GO AWAY;
               AWAY:
                    END OUTPUTSOURCE;
```

```
                    02190750 T  0003:023912
                    02191000 T  0003:023912
               START OF SEGMENT *********   10
                    02191250 T  0010:000010
                    02191500 T  0010:000610
                    02191750 T  0010:000713
                    02192000 T  0010:000810
                    02192250 T  0010:000913
                    02192500 T  0010:001010
                    02192750 T  0010:001112
                    02193000 T  0010:001210
                    02193250 T  0010:001411
                    02193500 T  0010:001513
                    02193750 T  0010:001513
                    02194000 T  0010:001610
                    02194250 T  0010:002312
                    02194500 T  0010:002513
                    02194750 T  0010:002713
                    02195000 T  0010:002810
                    02195250 T  0010:004512
                    02195500 T  0010:004513
                    02195750 T  0010:004610
                    02196000 T  0010:006210
                    02196250 T  0010:006210
                    02196500 T  0010:006210
               10 IS   63 LONG, NEXT SEG    3
```

```
                PROCEDURE BEGINPRINT;                                %108-       02196510 C  0003:023912
PRT(406) = BEGINPRINT
                BEGIN                                                %108-       02196520 C  0003:023912
                     STREAM PROCEDURE STUFF(N,L); VALUE N;           %108-       02196530 C  0003:023912
                                                                          START OF SEGMENT *********   11
PRT(407) = STUFF
                     BEGIN                                           %108-       02196540 C  0011:000010
                          DI:=L; DS:=8 LIT " "; SI:=L; DS:=13 WDS;   %108-       02196550 C  0011:000010
                          SI:=LOC N; DS:=8 DEC;                      %108-       02196560 C  0011:000210
                     END;                                            %108-       02196570 C  0011:000211



                     STUFF(BEGINSTACK[BSPOINT],LIN);                 %108-       02196580 C  0011:000211
                     IF NOHEADING THEN DATIME; WRITELINE;            %108-       02196590 C  0011:000411
                END BEGINPRINT;                                      %108-       02196610 C  0011:001610
                                                                         11 IS   17 LONG, NEXT SEG    3



          PROCEDURE READACARD;                                                   02196750 T  0003:023912
          COMMENT  READACARD READS CARDS FROM EITHER THE CARD READER OR THE      02197000 T  0003:023912
                   TAPE MERGING AS REQUESTED AND CREATING A NEW TAPE AND         02197250 T  0003:023912
                   LISTING IF REQUESTED.  READACARD ALSO INSERTS A PERCENT       02197500 T  0003:023912
```

```
                    SIGN AS AN END OF CARD SENTINEL IN COLUMN 73 AND SETS                    02197750 T  0003!023!2
                    FCR,NCR,LCR,TLCR, AND CLCR;                                             02198000 T  0003!023!2
                BEGIN                                                                       02198250 T  0003!023!2
                PROCEDURE READTAPE;                                                         02198500 T  0003!023!2
                                                                                      START OF SEGMENT **********   12
PRT(410) = READTAPE
                        BEGIN                                                               02201500 T  0012!000!0
            LABEL ENDREADTAPE, EOFT;                                              %105-     02201510 C  0012!000!0
                                                                                      START OF SEGMENT **********   13
            READ (TAPE, 10, TBUFF[*])[EOFT];                                      %105-     02201750 P  0013!000!0
PRT(411) = EOFT
PRT(412) = GO TO SOLVER
                        LCR!=MKABS(TBUFF[9]);                                     %105-     02202000 P  0013!000!2
            GO TO ENDREADTAPE;                                                    %105-     02202010 C  0013!000!2
            EOFT:                                                                 %105-     02202020 C  0013!000!3
            DEFINEARRAY[25]!="ND!END."& "E"[1!43!5];                              %105-     02202030 C  0013!000!0
            DEFINEARRAY[34]!="9999" & "9999"[1!25!23];                            %105-     02202040 C  0013!001!0
            TLCR!= MKABS(DEFINEARRAY[34]);                                        %105-     02202050 C  0013!001!1
            PUTSEQNO (DEFINEARRAY[33],TLCR-8);                                    %105-     02202060 C  0013!001!1
            TURNONSTOPLIGHT("%", TLCR-8);                                         %105-     02202070 C  0013!001!0
            ENDREADTAPE:                                                          %105-     02202080 C  0013!001!3
                        END READTAPE;                                                       02202250 T  0013!001!0
                                                                                    13 IS   24 LONG, NEXT SEG   12



        PROCEDURE SEQCOMPARE(TLCR,CLCR, LIB); VALUE LIB; BOOLEAN LIB;                       02202500 T  0012!000!0
PRT(413) = SEQCOMPARE
                                        REAL TLCR, CLCR ;                                   02202750 T  0012!000!0
            BEGIN                                                                           02203000 T  0012!000!0
            MEDIUM!="C "; % CARD READER,                                                    02203250 T  0012!000!0
            IF GT1!=COMPARE(TLCR,CLCR)=0 THEN  % TAPE HAS LOW SEQUENCE NUMB                 02203500 T  0012!000!1
                BEGIN                                                                       02203750 T  0012!000!2
                LCR!=TLCR; LASTUSED!=3;                                                     02204000 T  0012!000!3
                MEDIUM!="T "; % TAPE INPUT,                                                 02204250 T  0012!000!2
                END                                                                         02204500 T  0012!000!0
            ELSE BEGIN                                                                      02204750 T  0012!000!0
                IF GT1 ≠ 1 THEN % TAPE AND CARD HAVE SAME SEQ                               02205000 T  0012!000!2
                    BEGIN                                                                   02205250 T  0012!000!3
                    MEDIUM!="P "; % CARD PATCHES TAPE,                                      02205500 T  0012!001!0
                    READTAPE;                                                               02208500 T  0012!001!2
                    END;                                                                    02208750 T  0012!001!3
                LCR!=CLCR;                                                                  02209000 T  0012!001!3
                LASTUSED!=2;                                                                02209250 T  0012!001!0
                END;                                                                        02209500 T  0012!001!2
            END OF SEQCOMPARE;                                                              02209750 T  0012!001!2



        LABEL CARDONLY, CARDLAST, TAPELAST,  EXIT, FIRSTTIME,                               02210000 T  0012!001!2
              EOF, USETHESWITCH,                                                            02210250 T  0012!001!2
              COMPAR, TESTVOID, X,T;                                                        02210500 T  0012!001!2
            SWITCH USESWITCH!=CARDONLY,CARDLAST,TAPELAST,FIRSTTIME;                         02210750 T  0012!001!2
            IF ERRORCOUNT≥ERRMAX THEN ERR(611); % ERR LIMIT EXCEEDED - STOP.                02211500 T  0012!002!0
        USETHESWITCH:                                                                       02211750 T  0012!002!2
```

```
                    DOLLAR2TOG:=FALSE;                                                          02211800 T  00121002:2
                    GO TO USESWITCH[LASTUSED];                                                  02212000 T  00121002:3
                    MOVE(1,INFO[LASTUSED.LINKR,LASTUSED.LINKC],                                 02212250 T  00121002:3
                         DEFINEARRAY[DEFINEINDEX-2]);                                           02212500 T  00121002:1
                    LASTUSED := LASTUSED + 1;                                                   02212750 T  00121002:3
                    NCR := LCR-1;                                                               02213000 T  00121003:2
                    GO TO XIT;                                                                  02213250 T  00121003:0
               FIRSTTIME:                                                                       02213500 T  00121003:1
                    READ(CARD,10,CBUFF[*]);                                                     02213750 T  00121003:2
                    FCR:=NCR:=(LCR:=MKABS(CBUFF[9]))-9;                                         02214000 T  00121003:2
                    MEDIUM:="C ";                                                               02214100 T  00121004:1
                    IF EXAMIN(FCR)#"$" AND LISTER THEN PRINTCARD;                               02214200 T  00121004:3
                    PUTSEQNO(INFO[LASTSEQROW,LASTSEQUENCE],LCR);                                02214250 T  00121006:2
                    CARDNUMBER:=CONV(INFO[LASTSEQROW,LASTSEQUENCE-1],5,8);          %108-       02214260 C  00121006:2
                    TURNONSTOPLIGHT("%",LCR);                                                   02214500 T  00121006:2
                    GO XIT;                                                                     02214750 T  00121006:0
               COMMENT  WE HAVE JUST INITIALIZED CARD INPUT;                                    02215000 T  00121006:1
               CARDONLY:                                                                        02215250 T  00121006:1
                    READ(CARD,10,CBUFF[*]);                                                     02215500 T  00121006:2
                    LCR := MKABS(CBUFF[9]); GO EXIT;                                            02215750 T  00121007:2
               CARDLAST:                                                                        02216000 T  00121007:3
                    READ(CARD,10,CBUFF[*])[EOF];                                                02216250 T  00121007:0
PRT(414) = EOF
                    CLCR := MKABS(CBUFF[9]);                                                    02216500 T  00121008:2
                    GO COMPAR;                                                                  02216750 T  00121008:2
               EOF:                                                                             02217000 T  00121008:3
                    DEFINEARRAY[25]:="ND'END,"&"E"[1:43:5];                                     02217250 T  00121008:0
                    DEFINEARRAY[34]:="9999"&"9999"[1:25:23];                                    02217500 T  00121008:0
                    CLCR:=MKABS(DEFINEARRAY[34]);                                               02217750 T  00121008:1
                    PUTSEQNO(DEFINEARRAY[33],CLCR-8);                                           02218000 T  00121009:1
                    TURNONSTOPLIGHT("%",CLCR-8);                                                02218250 T  00121009:0
               %                                                                                02218400 T  00121009:3
                    GO COMPAR;                                                                  02218500 T  00121009:3
               COMMENT  THIS RELEASES THE PREVIOUS CARD FROM CARD READER AND                    02218750 T  00121009:2
                        SETS UP CLCR;                                                           02219000 T  00121009:2
               TAPELAST:                                                                        02219250 T  00121009:2
                    READTAPE;                                                                   02219500 T  00121009:2
               COMMENT  THIS RELEASES THE PREVIOUS CARD FROM TAPE AND SETS UP TLCR;              02219750 T  00121009:3
               COMPAR:                                                                          02224250 T  00121009:3
                    SEQCOMPARE(TLCR,CLCR,FALSE);                                                02224500 T  00121009:0
               EXIT:                                                                            02225000 T  00121009:2
                    NCR := FCR:= LCR - 9;                                                       02225250 T  00121010:0
               COMMENT  SETS UP NCR AND FCR;                                                    02225500 T  00121010:3
                    IF EXAMIN(FCR)#"$" THEN % $-CARDS DON"T COUNT.                              02225750 T  00121010:3
                      IF COMPARE(MKABS(INFO[LASTSEQROW,LASTSEQUENCE]),LCR)=1 THEN               02226000 T  00121010:3
                        BEGIN                                                                   02226250 T  00121010:0
                        FLAG(610); % SEQUENCE ERROR.                                            02226500 T  00121010:1
                        SEQUENCEERROR(LIN);                                                     02226750 T  00121010:2
                        END;                                                                    02227000 T  00121011:0
                    IF LASTUSED=3 THEN                                                          02228050 T  00121011:0
                         BEGIN                                                                  02228075 T  00121011:2
                         IF VOIDTAPE THEN GO USETHESWITCH;                                      02228100 T  00121011:3
                         IF VOIDTCR#0 THEN                                                      02228125 T  00121011:2
                           IF COMPARE(LCR,VOIDTCR)=0 THEN GO USETHESWITCH;                      02228150 T  00121011:3
                         END;                                                                   02228175 T  00121011:1
                    IF EXAMIN(FCR)="$" THEN                                                     02228250 T  00121011:1
                         BEGIN                                                                  02228500 T  00121011:1
```

```
          IF LISTPTOG OR PRINTDOLLARTOG THEN PRINTCARD;                                  02228750 T  0012:011:2
          NCR:=NCR+32768; DOLLARCARD;                                                    02229000 T  0012:013:1
COMMENT   DONT FORGET THAT NCR IS NOT WORD MODE, BUT CHAR. MODE POINTER;                 02229250 T  0012:013:0
          GO USETHESWITCH;                                                               02229500 T  0012:013:0
          END;                                                                           02229750 T  0012:014:0
     IF EXAMIN(FCR)=" " THEN                                                             02230000 T  0012:014:0
       IF DOLLAR2TOG:=EXAMIN(FCR+32768)="$" THEN                                         02230100 T  0012:014:3
          BEGIN                                                                          02230250 T  0012:014:2
          OUTPUTSOURCE;                                                                  02230500 T  0012:014:3
          NCR:=NCR+65536; % SCAN PAST " $" (CHARACTER MODE).                             02230750 T  0012:014:0
          DOLLARCARD;                                                                    02231000 T  0012:014:2
          END;                                                                           02231250 T  0012:014:3
     IF VOIDING THEN GO USETHESWITCH;                                                    02231500 T  0012:014:3
     IF VOIDCR#0 THEN                                                                    02231750 T  0012:014:2
          IF COMPARE(LCR,VOIDCR)>0 THEN VOIDCR:=VOIDPLACE:=0                             02232000 T  0012:015:0
            ELSE GO USETHESWITCH;                                                        02232250 T  0012:015:2
     IF VOIDTAPE THEN GO TESTVOID;                                                       02232500 T  0012:015:0
     IF VOIDTCR#0 THEN                                                                   02233000 T  0012:015:3
          IF COMPARE(LCR,VOIDTCR)>0 THEN VOIDTCR:=VOIDTPLACE:=0 ELSE                     02233500 T  0012:015:1
TESTVOID:  IF LASTUSED=3 THEN GO USETHESWITCH;                                           02234000 T  0012:016:1
     CARDCOUNT:=CARDCOUNT+1;                                                             02234500 T  0012:016:2
     IF DOLLAR2TOG THEN GO USETHESWITCH;                                                 02234600 T  0012:016:1
     PUTSEQNO(INFO[LASTSEQROW,LASTSEQUENCE],LCR);                                        02234750 T  0012:016:3
          CARDNUMBER:=IF SEQTOG THEN TOTALNO+ADDVALUE ELSE          %108-               02234800 C  0012:016:3
                      CONV(INFO[LASTSEQROW,LASTSEQUENCE-1],5,8);    %108-               02234900 C  0012:017:0
     OUTPUTSOURCE;                                                                       02235000 T  0012:017:3
     IF OMITTING THEN GO USETHESWITCH;                                                   02235250 T  0012:017:0
     %                                                                                   02235500 T  0012:017:3
          TURNONSTOPLIGHT("%",LCR);                                                      02235750 T  0012:017:3
     XIT;                                                                                02237750 T  0012:017:1
          END READACARD;                                                                 02238000 T  0012:017:2
                                                                                    12 IS  183 LONG, NEXT SEG    3



          REAL PROCEDURE CONVERT;                                                        02248000 T  0003:023:2
PRT(415) = CONVERT
                    BEGIN   REAL  T;  INTEGER  N;                                        02249000 T  0003:023:2
                                                                                  START OF SEGMENT **********   14
STACK(F+3) = T
STACK(F+4) = N
                    TLO+0;   THI+                                                        02250000 T  0014:000:0
                         T+ CONV(ACCUM[1],TCOUNT,N+(COUNT-TCOUNT)MOD 8);                 02251000 T  0014:000:1
                         FOR N+ TCOUNT+N STEP 8 UNTIL COUNT= 1 DO                        02252000 T  0014:000:2
                    IF DPTOG THEN                                                        02253000 T  0014:001:0
                    BEGIN                                                                02254000 T  0014:001:0
                    DOUBLE(THI,TLO,100000000,0,0,*,CONV(ACCUM[1],N,8),0,+,-,             02255000 T  0014:001:1
                                THI,TLO);                                                02256000 T  0014:001:2
                    T+THI;                                                               02257000 T  0014:001:6:0
                    END  ELSE                                                            02258000 T  0014:001:6:1
                         T+  T*100000000+ CONV(ACCUM[1],N,8);                            02259000 T  0014:001:6:1
                         CONVERT+T;                                                      02260000 T  0014:002:4:0
                    END;                                                                 02261000 T  0014:002:4:1
                                                                                    14 IS   28 LONG, NEXT SEG    3
```

```
                REAL STREAM PROCEDURE FETCH(F); VALUE F;                                                    02262000 T  0003:023:2
PRT(416) = FETCH
                    BEGIN  SI:=F; SI:=SI+8; DI:=LOC FETCH; DS:=WDS  END FETCH;                               02263000 T  0003:023:2



                PROCEDURE DUMPINFO;                                                                         02264000 T  0003:024:0
PRT(417) = DUMPINFO
                    BEGIN                                                                                   02264050 T  0003:024:0
                    ARRAY A[0:14]; INTEGER JEDEN,DWA;                                                       02264100 T  0003:024:0
                                                                                                  START OF SEGMENT **********   15
STACK(F+2) = A
STACK(F+3) = JEDEN
STACK(F+4) = DWA
                    STREAM PROCEDURE OCTALWORDS(S,D,N); VALUE N;                                            02264400 T  0015:000:3
PRT(420) = OCTALWORDS
                        BEGIN                                                                               02264450 T  0015:000:3
                        SI:=S; DI:=D;                                                                       02264500 T  0015:000:2
                        N(2(8(DS:=3 RESET; 3(IF SB THEN DS:=1 SET ELSE                                      02264550 T  0015:000:3
                             DS:=1 RESET; SKIP 1 SB)); DS:=1 LIT " ");DS:=2 LIT" ");                        02264600 T  0015:000:1
                        END OF OCTALWORDS;                                                                  02264650 T  0015:000:2



                    STREAM PROCEDURE ALPHAWORDS(S,D,N); VALUE N;                                            02264700 T  0015:000:2
PRT(421) = ALPHAWORDS
                        BEGIN                                                                               02264750 T  0015:000:2
                        SI:=S; DI:=D;                                                                       02264800 T  0015:001:0
                        N(2(4(DS:=1 LIT" "; DS:=1 CHR); DS:=1 LIT" "); DS:=2 LIT" ");                       02264850 T  0015:001:1
                        END OF ALPHAWORDS;                                                                  02264900 T  0015:001:1



                    IF NOHEADING THEN DATIME;WRITE(LINE[DBL],<//"ELBAT">);                                  02264950 T  0015:001:1
PRT(422) = *FORMAT DESCRIPTOR*
                                                                                                  16 IS     6 LONG, NEXT SEG   15
                    FOR JEDEN:=0 STEP 6 UNTIL 71 DO                                                         02265000 T  0015:001:2
                        BEGIN                                                                               02265050 T  0015:002:0
                        BLANKET(14,A); OCTALWORDS(ELBAT[JEDEN],A,6);                                        02265100 T  0015:002:0
                        WRITE(LINE[DBL],15,A[*]);                                                           02265150 T  0015:002:2
                        END;                                                                                02265200 T  0015:002:2
                    BLANKET(14,A); OCTALWORDS(ELBAT[72],A,4);                                               02265250 T  0015:002:3
                    WRITE(LINE[DBL],15,A[*]);                                                               02265300 T  0015:003:1
                    FOR JEDEN:=0 STEP 1 UNTIL NEXTINFO DIV 256 DO                                           02265350 T  0015:003:1
                        BEGIN                                                                               02265400 T  0015:004:2
                        WRITE(LINE[DBL],<//"INFO[",I2,",*]">,JEDEN);                                        02265450 T  0015:004:2
PRT(423) = *FORMAT DESCRIPTOR*
                                                                                                  17 IS     8 LONG, NEXT SEG   15
PRT(424) = *LIST, LABEL, OR SEGMENT DESCRIPTOR*
                    FOR DWA:=0 STEP 6 UNTIL 251 DO                                                          02265500 T  0015:004:0
                        BEGIN                                                                               02265550 T  0015:004:2
                        BLANKET(14,A); ALPHAWORDS(INFO[JEDEN,DWA],A,6);                                     02265600 T  0015:004:2
                        WRITE(LINE,15,A[*]);                                                                02265650 T  0015:005:1
                        BLANKET(14,A); OCTALWORDS(INFO[JEDEN,DWA],A,6);                                     02265700 T  0015:005:2
```

```
                WRITE(LINE[DBL],15,A[*]);                                    02265750 T  0015:0060:1
                END;                                                         02265800 T  0015:0065:2
            BLANKET(14,A); ALPHAWORDS(INFO[JEDEN,252],A,4);                  02265850 T  0015:0067:2
            WRITE(LINE,15,A[*]);                                             02265900 T  0015:0071:2
            BLANKET(14,A); OCTALWORDS(INFO[JEDEN,252],A,4);                  02265950 T  0015:0075:2
            WRITE(LINE[DBL],15,A[*]);                                        02266000 T  0015:0079:2
            END;                                                             02266050 T  0015:0083:2
        END OF DUMPINFO;                                                     02266100 T  0015:0083:3
                                                                    15 IS   88 LONG, NEXT SEG    3


DEFINE SKAN = BEGIN                                                          02277000 T  0003:0242:0
                COUNT:=RESULT:=ACCUM[1]:=0;                                  02278000 T  0003:0242:0
                SCANNER;                                                     02279000 T  0003:0242:0
                Q:=ACCUM[1];                                                 02280000 T  0003:0242:0
                END #;                                                       02281000 T  0003:0242:0
COMMENT  DOLLARCARD HANDLES THE COMPILER CONTROL CARDS.                      02282000 T  0003:0242:0
         ALL COMPILER- AND USER-DEFINED OPTIONS ARE KEPT                     02283000 T  0003:0242:0
         IN THE ARRAY "OPTIONS".                                             02284000 T  0003:0242:0
         EACH OPTION HAS A TWO-WORD ENTRY:                                   02285000 T  0003:0242:0
                                                                             02286000 T  0003:0242:0
           WORD          CONTAINS                                            02287000 T  0003:0242:0
           ----          --------                                            02288000 T  0003:0242:0
             1           ENTRY FROM ACCUM[1]: 00XZZZZ, WHERE                 02289000 T  0003:0242:0
                         X IS THE SIZE OF THE ID AND                         02290000 T  0003:0242:0
                         ZZZZZ IS THE FIRST FIVE CHARS OF THE ID.            02291000 T  0003:0242:0
             2           PUSH-DOWN, 47-BIT STACK CONTAINING THE              02292000 T  0003:0242:0
                         HISTORY OF THE SETTINGS OF THIS OPTION.             02293000 T  0003:0242:0
                                                                             02294000 T  0003:0242:0
         IN "FINDOPTION", ALL COMPILER-DEFINED OPTIONS ARE USUALLY           02295000 T  0003:0242:0
         LOCATED BASED UPON A UNIQUE NUMBER ASSIGNED TO EACH.                02296000 T  0003:0242:0
         FOR ALL USER-DEFINED OPTIONS, A SEQUENTIAL TABLE SEARCH IS          02297000 T  0003:0242:0
         INITIATED USING "USEROPINX" AS THE INITIAL INDEX INTO THE           02298000 T  0003:0242:0
         "OPTIONS" ARRAY. IF THE NUMBER OF COMPILER-DEFINED OPTIONS          02299000 T  0003:0242:0
         IS CHANGED, THEN "USEROPINX" MUST BE ACCORDINGLY CHANGED.           02300000 T  0003:0242:0
         THE NUMBER OF USER DEFINED OPTIONS ALLOWED CAN BE                   02301000 T  0003:0242:0
         CHANGED BY CHANGING THE DEFINE "OPARSIZE".                          02302000 T  0003:0242:0
         THE VARIABLE "OPTIONWORD" CONTAINS THE CURRENT TRUE OR FALSE        02303000 T  0003:0242:0
         SETTING OF ALL OF THE COMPILER-DEFINED OPTIONS, ONE BIT PER         02304000 T  0003:0242:0
         OPTION.                                                             02305000 T  0003:0242:0
                                                                             02306000 T  0003:0242:0
        ;                                                                    02307000 T  0003:0242:0
       BOOLEAN PROCEDURE FINDOPTION(BIT); VALUE BIT; INTEGER BIT;
PRT(425) = FINDOPTION
            BEGIN                                                            02308000 T  0003:0242:0
            LABEL FOUND;                                                     02309000 T  0003:0242:0
                                                                  START OF SEGMENT **********   18
            REAL ID;                                                         02310000 T  0018:0000:0
STACK(F+3) = ID
            OPINX:=2*BIT-4;                                                  02311000 T  0018:0000:0
            WHILE ID:=OPTIONS[OPINX:=OPINX+2]#0 DO                           02312000 T  0018:0001:3
                 IF Q=ID THEN GO FOUND;                                      02313000 T  0018:0005:2
            OPTIONS[OPINX]:=Q; % NEW USER-DEFINED OPTION.                    02314000 T  0018:0006:1
          FOUND;                                                    %103=    02315000 P  0018:0008:0
             IF OPINX +1>OPARSIZE THEN FLAG(602) ELSE % TOO MANY USER OPTIONS 02316000 P  0018:0008:0
            FINDOPTION:=BOOLEAN(OPTIONS[OPINX+1]);                           02317000 T  0018:0010:1
```

```
                    END FINDOPTION;                                                                  02318000 T  0018:0012:1
                                                                                                   18 IS   15 LONG, NEXT SEG    3


                PROCEDURE DOLLARCARD;                                                                02319000 T  0003:0242:0
                    BEGIN                                                                            02320000 T  0003:0242:0
                    STRFAM PROCEDURE RESTORESEQNUM(LCR,INFO); VALUE LCR;                             02320200 T  0003:0242:0
                                                                                                  START OF SEGMENT **********   19
PRT(426) = RESTORESEQNUM
                        BEGIN                                                                        02320400 T  0019:0000:0
                        DI:=LCR; SI:=INFO; DS:=WDS;                                                  02320600 T  0019:0000:0
                        END;                                                                         02320800 T  0019:0000:1


                PROCEDURE SWITCHIT(XBIT); VALUE XBIT; INTEGER XBIT;                                  02321000 T  0019:0001:2
PRT(427) = SWITCHIT
                        BEGIN                                                                        02322000 T  0019:0001:2
                        BOOLEAN B,T;                                                                 02323000 T  0019:0001:2
                                                                                                  START OF SEGMENT **********   20
STACK(F+2) = B
STACK(F+3) = T
                        INTEGER SAVEINX;                                                             02324000 T  0020:0000:0
STACK(F+4) = SAVEINX
                        LABEL XMODE0,XMODE1,XMODE2,XMODE3,XMODE4,ALONG;                               02325000 T  0020:0000:0
                        SWITCH SW:=XMODE0,XMODE1,XMODE2,XMODE3,XMODE4;                               02326000 T  0020:0000:0
                        SETTING:=FINDOPTION(XBIT); SKAN;                                             02327000 T  0020:0005:3
                        GO SW[XMODE+1];                                                              02328000 T  0020:0011:2
                XMODE0: % FIRST OPTION ON CARD, BUT NOT SET, RESET, OR POP,                          02329000 T  0020:0013:3
                        OPTIONWORD:=BOOLEAN(0);                                                      02330000 T  0020:0014:0
                        FOR SAVEINX:=1 STEP 2 UNTIL OPARSIZE DO OPTIONS[SAVEINX]:=0;                 02331000 T  0020:0014:1
                        XMODE:=LASTUSED:=1; % CARD INPUT ONLY.                                       02332000 T  0020:0019:3
                XMODE1: % NOT FIRST OPTION AND NOT BEING SET, RESET, OR POPPED.                      02333000 T  0020:0020:1
                        OPTIONS[OPINX+1]:=REAL(TRUE);                                                02334000 T  0020:0021:2
                        IF XBIT<USEROPINX THEN OPTIONWORD:=OPTIONWORD & TRUE[XBIT:1];                02335000 T  0020:0022:1
PRT(430) = DYNAMIC DIALS
                        GO ALONG;                                                                    02336000 T  0020:0026:1
                XMODE2: % RESET,                                                                     02337000 T  0020:0027:2
                        OPTIONS[OPINX+1]:=REAL(FALSE & SETTING[1:2:46]);                             02338000 T  0020:0028:0
                        IF XBIT<USEROPINX THEN OPTIONWORD:=OPTIONWORD & FALSE[XBIT:1];               02339000 T  0020:0030:1
                        GO ALONG;                                                                    02340000 T  0020:0034:1
                XMODE3: % SET,                                                                       02341000 T  0020:0035:2
                        SAVEINX:=OPINX; % REMEMBER OPTION WE ARE SETTING,                            02342000 T  0020:0036:0
                        B:=IF Q="1=0000" THEN BOOLEXP ELSE TRUE;                                     02343000 T  0020:0036:1
                        OPTIONS[SAVEINX+1]:=REAL(B & SETTING[1:46]);                                 02352000 T  0020:0039:3
                        IF XBIT<USEROPINX THEN OPTIONWORD:=OPTIONWORD & B [XBIT:1];                  02353000 T  0020:0042:1
                        GO ALONG;                                                                    02354000 T  0020:0046:1
                XMODE4: % POP,                                                                       02355000 T  0020:0048:0
                        OPTIONS[OPINX+1]:=REAL(B:=SETTING,[1:46]);                                   02356000 T  0020:0048:0
                        IF XBIT<USEROPINX THEN OPTIONWORD:=OPTIONWORD & B [XBIT:1];                  02357000 T  0020:0051:2
                ALONG:                                                                               02358000 T  0020:0055:2
                        END SWITCHIT;                                                                02359000 T  0020:0056:0
                                                                                                   20 IS   59 LONG, NEXT SEG   19
```

```
                    LABEL EXIT,AGAIN,SKANAGAIN,LENGTH1,LENGTH2,LENGTH3,LENGTH4,          02360000 T  0019:0001:2
                          LENGTH5,LENGTH6,LENGTH7,LENGTH8,LENGTH9,                      02361000 T  0019:0001:2
                          WHATISIT,                                                     02362000 T  0019:0001:2
                          CARDOPTION,MERGEOPTION;                                       02363000 T  0019:0001:2
                    SWITCH OPTIONLENGTH:=LENGTH1,WHATISIT,LENGTH3,LENGTH4,LENGTH5,      02364000 T  0019:0001:2
                                         LENGTH6,LENGTH7,WHATISIT,LENGTH9,WHATISIT;     02365000 T  0019:0003:3
                    INTEGER SRESULT,SCOUNT;                                             02365100 T  0019:0009:2
STACK(F+2) = SRESULT
STACK(F+3) = SCOUNT
                    ALPHA VOIDRANGE;                                                    02365200 T  0019:0009:2
STACK(F+4) = VOIDRANGE
                    DOLLARTOG:=TRUE;                                                    02366000 T  0019:0009:2
                    MOVE(10,ACCUM[0],DEFINEARRAY[0]); % SAVE INFORMATION FOR            02366100 T  0019:0010:1
                    SCOUNT:=COUNT; SRESULT:=RESULT;   % "TABLE" TO RESUME SCAN.       02366200 T  0019:0012:1
                    XMODE:=0;                                                           02367000 T  0019:0014:0
                    PUTSEQNO(INFO[LASTSEQROW,LASTSEQUENCE],LCR);                        02368000 T  0019:0014:1
                    TURNONSTOPLIGHT("%",LCR);                                           02369000 T  0019:0016:1
                SKANAGAIN:                                                              02370000 T  0019:0017:3
                    SKAN;                                                               02371000 T  0019:0018:0
                AGAIN:                                                                  02372000 T  0019:0021:3
                    GO OPTIONLENGTH[MIN(COUNT,10)];                                     02373000 T  0019:0022:0
                LENGTH1:                                                                02374000 T  0019:0026:0
                    IF Q = "1%0000" THEN GO EXIT;                                       02375000 T  0019:0027:2
                    IF Q = "1$0000" THEN                                                02376000 T  0019:0028:0
                        BEGIN SWITCHIT(PRINTDOLLARBIT); GO AGAIN END;                   02377000 T  0019:0029:2
                    IF Q = "1,0000" THEN GO SKANAGAIN;                                  02378000 T  0019:0033:2
                    GO WHATISIT;                                                        02379000 T  0019:0034:0
                LENGTH2: % NO OPTIONS OF THIS LENGTH ARE CURRENTLY IMPLEMENTED.         02380000 T  0019:0036:0
                LENGTH3:                                                                02381000 T  0019:0036:0
                    IF Q = "3SET00" THEN                                                02382000 T  0019:0036:0
                        BEGIN XMODE:=3; GO SKANAGAIN END;                               02383000 T  0019:0036:1
                    IF Q = "3POP00" THEN                                                02384000 T  0019:0040:0
                        BEGIN XMODE:=4; GO SKANAGAIN END;                               02385000 T  0019:0040:1
                    IF Q = "3NEW00" THEN                                                02386000 T  0019:0044:0
                        BEGIN                                                           02387000 T  0019:0044:1
                        SWITCHIT(NEWBIT);                                               02388000 T  0019:0045:2
                        IF Q = "4TAPE0" THEN GO SKANAGAIN;                              02389000 T  0019:0046:0
                        GO AGAIN;                                                       02390000 T  0019:0047:2
                        END;                                                            02391000 T  0019:0050:0
                    IF Q = "3SEQ00" THEN                                                02392000 T  0019:0050:0
                        BEGIN SWITCHIT(SEQBIT); GO AGAIN END;                           02393000 T  0019:0050:1
                    IF Q = "3PRT00" THEN                                                02394000 T  0019:0054:0
                        BEGIN SWITCHIT(PRTBIT); GO AGAIN END;                           02395000 T  0019:0054:1
                    IF Q = "3MCP00" THEN                                                02396000 T  0019:0058:0
                        BEGIN SWITCHIT(MCPBIT); GO AGAIN END;                           02397000 T  0019:0058:1
                    GO WHATISIT;                                                        02398000 T  0019:0062:0
                LENGTH4:                                                                02399000 T  0019:0062:1
                    IF Q = "4LIST0" THEN                                                02400000 T  0019:0063:2
                        BEGIN                                                           02401000 T  0019:0063:3
                        SWITCHIT(LISTBIT);                                              02402000 T  0019:0064:0
                        GO AGAIN;                                                       02404000 T  0019:0065:2
                        END;                                                            02405000 T  0019:0067:2
                    IF Q = "4VOID0" THEN                                                02406000 T  0019:0067:2
                        BEGIN                                                           02407000 T  0019:0067:3
                        IF XMODE=0 THEN                                                 02408000 T  0019:0068:0
                            BEGIN                                                       02409000 T  0019:0069:2
                            GETVOID(VOIDRANGE,NCR,LCR,INFO[LASTSEQROW,LASTSEQUENCE]);   02410000 T  0019:0069:3
```

```
                IF VOIDCR=0 THEN VOIDCR:=MKABS(VOIDPLACE) ELSE                  02410500 T  0019:0072:0
                  IF COMPARE(MKABS(VOIDRANGE),VOIDCR)≠1 THEN GO TO EXIT;        02411000 T  0019:0075:2
                MOVE(1,VOIDRANGE,VOIDPLACE);                                    02412000 T  0019:0080:1
                GO EXIT;                                                        02413000 T  0019:0081:3
                END;                                                            02414000 T  0019:0082:0
             SWITCHIT(VOIDBIT);                                                 02415000 T  0019:0082:0
             GO AGAIN;                                                          02418000 T  0019:0083:2
             END;                                                               02419000 T  0019:0083:3
        IF Q = "4XREFO" THEN BEGIN SWITCHIT(XREFBIT); IF BOOLEAN(XMODE) THEN    02419100 C  0019:0083:3
             DEFINING := BOOLEAN(REAL(DEFINING)&1[1:47:1]);              %108-  02419110 C  0019:0085:3
                              GO AGAIN  END;                             %108-  02419120 C  0019:0088:0
        IF Q = "4BENDO" THEN BEGIN SWITCHIT(BENDBIT); GO AGAIN END;     %108-  02419200 C  0019:0090:0
        IF Q = "4OMITO" THEN                                                    02420000 T  0019:0094:0
             BEGIN IF NOT DOLLAR2TOG THEN BEGIN PRINTCARD; FLAG(605); END;      02421000 T  0019:0094:1
                SWITCHIT(OMITBIT); GO AGAIN END;                                02421100 T  0019:0114:0
        IF Q = "4CARDO" THEN                                                    02422000 T  0019:0115:2
             BEGIN                                                              02423000 T  0019:0116:0
             Q:="4TAPEO"; % FAKE OUT SWITCHIT.                                  02424000 T  0019:0116:1
             SWITCHIT(MERGEBIT);                                                02425000 T  0019:0117:2
             IF XMODE≠2 THEN MERGETOG:=NOT MERGETOG;                            02425500 T  0019:0118:0
             OPTIONS[2×MERGEBIT-1]:=                  % CARD IS                 02426000 T  0019:0121:3
               REAL(SETTING & (MERGETOG)[47:1]);      % INVERSE OF MERGE.        02427000 T  0019:0123:2
             IF MERGETOG THEN GO MERGEOPTION;                                   02428000 T  0019:0125:3
CARDOPTION:                                                                     02429000 T  0019:0127:2
             LASTUSED:=1;                                                       02430000 T  0019:0127:2
             GO AGAIN;                                                          02431000 T  0019:0127:3
             END;                                                               02432000 T  0019:0131:2
        IF Q = "4TAPEO" THEN                                                    02433000 T  0019:0131:2
             BEGIN                                                              02434000 T  0019:0131:3
             SWITCHIT(MERGEBIT);                                                02435000 T  0019:0132:0
             IF NOT MERGETOG THEN GO CARDOPTION;                                02436000 T  0019:0133:2
MERGEOPTION:                                                                    02437000 T  0019:0134:0
             LASTUSED:=2; % NEXT CARD IS READ FROM READER.                      02438000 T  0019:0135:2
             IF MAXTLCR=0 THEN                                                  02439000 T  0019:0135:3
                  BEGIN                                                         02440000 T  0019:0136:1
                  INTEGER STREAM PROCEDURE FEJ(F,T); VALUE T;                   02441000 T  0019:0137:2
PRT(431) = *SEGMENT DESCRIPTOR*

                                                                  START OF SEGMENT **********   21
PRT(432) = FEJ
                                BEGIN                                           02442000 T  0021:0000:0
                                SI:=F; DI:=LOC T; DS:=WDS;                      02443000 T  0021:0000:0
                                SI:=T; SI:=SI-16; DI:=LOC FEJ; DS:=WDS;         02444000 T  0021:0000:1
                                END FEJ;                                        02445000 T  0021:0001:3



                           STREAM PROCEDURE FIX(F,T); VALUE T;                  02446000 T  0021:0002:1
PRT(433) = FIX
                                BEGIN                                           02447000 T  0021:0002:1
                                SI:=F; SI:=SI-24; DI:=LOC T; DS:=WDS;           02448000 T  0021:0003:2
                                DI:=T; DI:=DI+47; SKIP 4 DB; DS:=2 RESET;       02449000 T  0021:0004:0
                                2(DI:=DI+48); DS:=8 LIT"00#01+0#";              02450000 T  0021:0005:2
                                END FIX;                                        02451000 T  0021:0007:2
```

```
                    IF GT1:=FEJ(TAPE,0)=10 THEN                                                 02452000 T  0021:0007:2
                         BEGIN                                                                  02453000 T  0021:0010:1
                         REWIND(TAPE); FIX(TAPE,0);                                             02454000 T  0021:0011:2
                         END;                                                                   02455000 T  0021:0014:0
                    MAXTLCR:=GT1+TLCR:=9+MKABS(TBUFF[0]);                                       02456000 T  0021:0014:0
                    READ(TAPE,10,TBUFF[*]); % INITIALIZE TAPE INPUT.                            02457000 T  0021:0017:3
                    LASTUSED:=2;                                                                02458000 T  0021:0022:0
                    END;                                                                        02459000 T  0021:0022:1
PRT(434) = *SEGMENT DESCRIPTOR*

                                                                                          21 IS   24 LONG, NEXT SEG   19
                         GO AGAIN;                                                              02460000 T  0019:0139:2
                         END;                                                                   02461000 T  0019:0139:3
                    IF Q = "4PAGE0" THEN                                                        02462000 T  0019:0139:3
                         BEGIN                                                                  02463000 T  0019:0140:0
                         IF LISTER THEN WRITE(LINE[PAGE]);                                      02464000 T  0019:0140:1
                         GO SKANAGAIN;                                                          02465000 T  0019:0145:3
                         END;                                                                   02466000 T  0019:0147:2
                    IF Q = "4INFO0" THEN                                                        02467000 T  0019:0147:2
                         BEGIN DUMPINFO; GO SKANAGAIN END;                                      02468000 T  0019:0147:3
                    IF Q = "4SEGS0" THEN                                                        02469000 T  0019:0151:2
                         BEGIN SWITCHIT(SEGSBIT); GO AGAIN END;                                 02470000 T  0019:0151:3
                    IF Q = "4NEST0" THEN                                                        02471000 T  0019:0155:2
                         BEGIN SWITCHIT(NESTBIT); GO AGAIN END;                                 02472000 T  0019:0155:3
                    IF Q = "4DECK0" THEN                                                        02473000 T  0019:0159:2
                         BEGIN SWITCHIT(DECKBIT); GO AGAIN END;                                 02474000 T  0019:0159:3
                    GO WHATISIT;                                                                02475000 T  0019:0163:2
               LENGTH5:                                                                         02476000 T  0019:0163:3
                    IF Q = "5RESET" THEN                                                        02477000 T  0019:0164:0
                         BEGIN XMODE:=2; GO SKANAGAIN END;                                      02478000 T  0019:0164:1
               IF Q = "5LISTP" THEN                                                             02479000 T  0019:0168:0
                    BEGIN SWITCHIT(LISTPBIT); GO AGAIN; END;                                    02480000 T  0019:0168:1
               IF Q = "5VOIDT" THEN                                                             02481000 T  0019:0172:0
                    BEGIN                                                                       02482000 T  0019:0172:1
                    IF XMODE=0 THEN                                                             02483000 T  0019:0173:2
                         BEGIN                                                                  02484000 T  0019:0174:0
                         GETVOID(VOIDRANGE,NCR,LCR,INFO[LASTSEQROW,LASTSEQUENCE]);              02485000 T  0019:0174:1
                         IF VOIDTCR=0 THEN VOIDTCR:=MKABS(VOIDTPLACE) ELSE                      02485500 T  0019:0177:2
                           IF COMPARE(MKABS(VOIDRANGE),VOIDTCR)#1 THEN GO TO EXIT;              02486000 T  0019:0180:0
                         MOVE(1,VOIDRANGE,VOIDTPLACE);                                          02487000 T  0019:0185:3
                         GO EXIT;                                                               02488000 T  0019:0186:1
                         END;                                                                   02489000 T  0019:0187:2
                    SWITCHIT(VOIDTBIT);                                                         02490000 T  0019:0187:2
                    GO AGAIN;                                                                   02493000 T  0019:0188:0
                    END;                                                                        02494000 T  0019:0188:1
               IF Q = "5CHECK" THEN                                                             02495000 T  0019:0188:1
                    BEGIN SWITCHIT(CHECKBIT); GO AGAIN END;                                     02496000 T  0019:0189:2
               IF Q = "5LIMIT" THEN                                                             02497000 T  0019:0192:0
                    BEGIN                                                                       02498000 T  0019:0192:1
                    SKAN;                                                                       02499000 T  0019:0193:2
                    IF RESULT#3 THEN % SHOULD BE NUMBER.                                        02500000 T  0019:0197:2
                         BEGIN FLAG(600); GO AGAIN END;                                         02501000 T  0019:0197:3
                    ERRMAX:=CONV(ACCUM[1],0,ACCUM[1],[12:6]);                                   02502000 T  0019:0201:2
                    GO SKANAGAIN;                                                               02503000 T  0019:0204:0
                    END;                                                                        02504000 T  0019:0204:1
               IF Q = "5PUNCH" THEN                                                             02505000 T  0019:0204:1
                    BEGIN SWITCHIT(PUNCHBIT); GO AGAIN; END;                                    02506000 T  0019:0205:3
               IF Q = "5PURGE" THEN                                                             02507000 T  0019:0209:2
```

```
                BEGIN SWITCHIT(PURGEBIT); GO AGAIN; END;                                                02508000 T  0019!020!3
            IF Q = "5LISTA" THEN                                                                        02509000 T  0019!021!2
                BEGIN                                                                                   02510000 T  0019!021!3
                SWITCHIT(LISTABIT);                                                                     02511000 T  0019!021!0
                GO AGAIN;                                                                               02513000 T  0019!021!2
                END;                                                                                    02514000 T  0019!021!2
            IF Q = "5STUFF" THEN                                                                        02515000 T  0019!021!2
                BEGIN SWITCHIT(STUFFBIT); GO AGAIN; END;                                                02516000 T  0019!021!3
            GO WHATISIT;                                                                                02517000 T  0019!022!2
        LENGTH6:                                                                                        02518000 T  0019!022!3
            IF Q = "6SEQER" THEN                                                                        02519000 T  0019!022!0
                BEGIN SWITCHIT(SEQERRBIT); GO AGAIN END;                                                02520000 T  0019!022!1
            IF Q = "6SINGL" THEN                                                                        02521000 T  0019!022!0
                BEGIN SWITCHIT(SINGLBIT); GO AGAIN  END;                                                02522000 T  0019!022!1
            IF Q = "6SEQXE" THEN                                                                        02523000 T  0019!023!0
                BEGIN                                                                                   02524000 T  0019!023!1
                SEQXEQTOG:=XMODE#2 AND XMODE# 4 OR SEQXEQTOG;%NEVER RESET.                              02525000 T  0019!023!2
                IF BUILDLINE.[45:1] THEN BUILDLINE:=TRUE;                                               02526000 T  0019!023!0
                GO SKANAGAIN;                                                                           02527000 T  0019!023!0
                END;                                                                                    02528000 T  0019!023!0
            IF Q = "6DEBUG" THEN                                                                        02529000 T  0019!023!0
                BEGIN                                                                                   02530000 T  0019!023!1
                SWITCHIT(DEBUGBIT);                                                                     02531000 T  0019!023!2
                IF DEBUGTOG THEN                                                                        02533000 T  0019!024!0
                    IF WOP[0]=0 THEN                                                                    02534000 T  0019!024!1
                        BEGIN                                                                           02535000 T  0019!024!0
                        FILL WOP[*] WITH                                                                02536000 T  0019!024!1
                "LITC","    ",                                                                          02537000 T  0019!024!2
                                                                                                START OF SEGMENT **********   22
                "OPDC","DESC",                                                                          02538000 T  0019!024!1
             11, "NOP ", 12, "PRT ", 13, "DFL ", 16, "ADD ", 18, "PRL ", 19, "LNG ",                    02539000 T  0019!024!1
             21, "GFQ ", 22, "BBC ", 24, "INX ", 35, "LOR ", 37, "GTR ", 38, "BFC ",                    02540000 T  0019!024!1
             39, "RTN ", 40, "COC ", 48, "SUB ", 64, "MUL ", 67, "LND ", 68, "STD ",                    02541000 T  0019!024!1
             69, "NEQ ", 71, "XIT ", 72, "MKS ",128, "DIV ",130, "COM ",131, "LQV ",                    02542000 T  0019!024!1
            132, "SND ",133, "XCH ",134, "CHS ",167, "RTS ",168, "CDC ",260, "LOD ",                    02543000 T  0019!024!1
            261, "DUP ",278, "LBC ",294,"LFC ",322,"ZP1 ",384, "IDV ",532, "ISD ",                      02544000 T  0019!024!1
            533, "LEQ ",534, "BBW ",548, "ISN ",549, "LSS ",550, "BFW ",581, "EQL ",                    02545000 T  0019!024!1
            582, "SSP ",790, "LBU ",806, "LFU ",896, "RDV ",                                            02546000 T  0019!024!1
            1003,"SCI ",1004,"SAN ",1019,"SCS ",                                                        02547000 T  0019!024!1
             1023,1023,1023,1023,1023,1023,1023,1023,1023,1023,1023,1023,1023,1023,                     02548000 T  0019!024!1
                    1023,1023,1023,1023,1023,1023,1023,1023,1023,1023,1023, 1023;                       02549000 T  0019!024!1
                                                                                                  22 IS  128 LONG, NEXT SEG   19
                        FILL COP[*] WITH  % CHARACTER MODE MNEMONICS                                    02550000 T  0019!024!1
             0,"   ",0,0,                                                                               02551000 T  0019!024!2
                                                                                                START OF SEGMENT **********   23
             0,"EXC ", 2,"BSD ", 3,"BSS ", 4,"RDA ", 5,"TRW ",6,"SED ", 7,"TDA ",                       02552000 T  0019!024!0
            12,"SDA ",13,"SSA ",14,"SFD ",15,"SRD ",18,"SES ",20,"TEQ ",21, "TNE ",                     02553000 T  0019!024!0
            22,"TEG ",23,"TGR ",24,"SRS ",25,"SFS ",28,"TEL ",29,"TLS ",30,"TAN ",                      02554000 T  0019!024!0
            31,"BIT ",32,"INC ",33,"STC ",34,"SEC ",35,"CRF ",36,"JNC ",37,"JFC ",                      02555000 T  0019!024!0
            38,"JNS ",39,"JFW ",40,"RCA ",41,"ENS ",42,"BNS ",43,"RSA ",44,"SCA ",                      02556000 T  0019!024!0
            45,"JRC ",46,"TSA ",47,"JRV ",48,"CEQ ",49,"CNE ",50,"CEG ",51,"CGR ",                      02557000 T  0019!024!0
           52,"BIT ",53,"BIR ",54,"OCV ",55,"ICV ",56,"CEL ",57,"CLS ",58,"FSU ",                       02558000 T  0019!024!0
            59,"FAD ",60,"TRP ",61,"TRN ",62,"TRZ ",63,"TRS ",64,0,64,0,64,0,64,0,                      02559000 T  0019!024!0
            64,0,64,0,64,0,64,0;                                                                        02560000 T  0019!024!0
                                                                                                  23 IS  128 LONG, NEXT SEG   19
                        END;                                                                            02563000 T  0019!024!0
                GO AGAIN;                                                                               02564000 T  0019!024!0
                END;                                                                                    02565000 T  0019!024!0
```

```
    IF Q = "6FORMA" THEN                                                    02566000 T  0019:024:0
        BEGIN SWITCHIT(FORMATBIT); GO AGAIN; END;                           02567000 T  0019:024:1
    GO WHATISIT;                                                            02568000 T  0019:025:0
LENGTH7:                                                                    02569000 T  0019:025:1
%   IF Q = "7INCLU" THEN                                                    02570000 T  0019:025:2
%       BEGIN DOLLARCARD:=STARTINCLUDING;  GO EXIT;  END;                   02571000 T  0019:025:2
%   IF Q = "7INCLN" THEN                                                    02572000 T  0019:025:2
%       BEGIN SWITCHIT(NEWINCLBIT);  GO AGAIN; END;                         02573000 T  0019:025:2
LENGTH8: % NO OPTIONS OF THIS LENGTH ARE CURRENTLY IMPLEMENTED.             02574000 T  0019:025:2
LENGTH9:                                                                    02575000 T  0019:025:2
    IF Q = "9INTRI" THEN                                                    02576000 T  0019:025:2
        BEGIN                                                               02576500 T  0019:025:3
        INTOG:=XMODE#2 AND XMODE# 4 OR INTOG; % NEVER RESET.                02577000 T  0019:025:4
        GO SKANAGAIN;                                                       02577250 T  0019:025:8:1
        END;                                                                02577500 T  0019:026:0
WHATISIT:                                                                   02578000 T  0019:026:0
    IF RESULT=3 THEN                                                        02579000 T  0019:026:0
        BEGIN                                                               02580000 T  0019:026:1
        BASENUM:=CONV(ACCUM[1],0,ACCUM[1],[12:6]);                          02581000 T  0019:026:2
        TOTALNO:=-10;                                                       02582000 T  0019:026:4:1
        NEWBASE:=TRUE;                                                      02583000 T  0019:026:5:3
        GO SKANAGAIN;                                                       02584000 T  0019:026:6:0
        END;                                                                02585000 T  0019:026:6:1
    IF RESULT=2 THEN                                                        02586000 T  0019:026:6:1
        BEGIN                                                               02587000 T  0019:026:7:3
        IF Q = "1+0000" THEN                                                02588000 T  0019:026:8:0
            BEGIN                                                           02589000 T  0019:026:8:1
            SKAN;                                                           02590000 T  0019:026:9:2
            IF RESULT=3 THEN                                                02591000 T  0019:027:3:2
             ADDVALUE:=CONV(ACCUM[1],0,ACCUM[1],[12:6])                     02592000 T  0019:027:3:3
             ELSE FLAG(600); % NUMBER EXPECTED.                             02593000 T  0019:027:6:0
            END;                                                            02594000 T  0019:027:9:3
        GO SKANAGAIN;                                                       02595000 T  0019:027:9:3
        END;                                                                02596000 T  0019:028:0:0
COMMENT  DID NOT RECOGNIZE OPTION;                                          02597000 T  0019:028:0:0
    IF RESULT#1 THEN % NOT AN IDENTIFIER,                                   02598000 T  0019:028:0:0
        BEGIN FLAG(601); GO SKANAGAIN END;                                  02599000 T  0019:028:1:2
    SWITCHIT(USEROPINX); % USEROPINX MEANS A USER-DEFINED OPTION.           02600000 T  0019:028:2:1
    GO AGAIN;                                                               02601000 T  0019:028:3:3
 EXIT:                                                                      02602000 T  0019:028:4:0
    LISTER:=DEBUGTOG OR LISTOG OR LISTATOG;                                 02602500 T  0019:028:4:0
    MOVF(10,DEFINEARRAY[0],ACCUM[0]); % RESTORE INFORMATION FOR             02602600 T  0019:028:7:2
    COUNT:=SCOUNT; RESULT:=SRESULT;   % "TABLE" TO RESUME SCAN.             02602700 T  0019:028:9:2
    RESTORESEQNUM(LCR,INFO,LASTSEQROW,LASTSEQUENCE); % FOR VOID TESTS       02602800 T  0019:029:0:1
    DOLLARTOG:=FALSE;                                                       02603000 T  0019:029:2:1
    END DOLLARCARD;                                                         02604000 T  0019:029:3:2
                                                                  19 IS  296 LONG, NEXT SEG    3
```

```
COMMENT  TABLE IS THE ROUTINE THAT MOST CODE IN THE COMPILER                02605000 T  0003:024:2:0
       USES WHEN IT IS DESIRED TO SCAN ANOTHER LOGICAL QUANTITY.            02606000 T  0003:024:2:0
       THE RESULT RETURNED IS THE CLASS OF THE ITEM DESIRED.                02607000 T  0003:024:2:0
       TABLE MAINTAINS THE VARIABLES I AND NXTELBT AND THE ARRAY            02608000 T  0003:024:2:0
       ELBAT.  ELBAT AND I ARE PRINCIPAL VARIABLES USED FOR                 02609000 T  0003:024:2:0
       COMUNICATION BETWEEN TABLE AND THE OUTSIDE WORLD.  NXTELBT           02610000 T  0003:024:2:0
```

```
                    IS ALMOST EXCLUSIVELY USED BY TABLE, ALTHOUGH AN OCCASION-      02611000 T  0003!024!0
                    AL OTHER USE IS MADE IN ORDER TO FORGET THAT SOMETHING WAS      02612000 T  0003!024!0
                    SCANNED.  (SEE, FOR EXAMPLE, COMPOUNDTAIL). FOR FURTHER        02613000 T  0003!024!0
                    GENERAL DISCUSSION SEE THE DECLARATION OF THESE VARIABLES.      02614000 T  0003!024!0
                      THE PARAMETER P IS THE ACTUAL INDEX OF THE QUANTITY           02615000 T  0003!024!0
                    DESIRED (USUALLY I-1,I, OR I+1).                               02616000 T  0003!024!0
                      THE GENERAL PLAN OF TABLE IS THIS:                           02617000 T  0003!024!0
                        I) IF P < NXTELBAT GO ON TO III).                          02618000 T  0003!024!0
                       II) PROCESS ONE QUANTITY.                                   02619000 T  0003!024!0
                         A) SCAN.                                                  02620000 T  0003!024!0
                         B) TEST FOR IDENTIFIER, NUMBER, OR SPECIAL CHARACTER.     02621000 T  0003!024!0
                          1) IDENTIFIER - LOOKUP IN DIRECTORY AND PROCESS          02622000 T  0003!024!0
                               IN SPECIAL MANNER IF COMMENT OR DEFINED ID.         02623000 T  0003!024!0
                          2) NUMBER - PROCESS INTEGER PART, FRACTIONAL PART,       02624000 T  0003!024!0
                               AND EXPONENT PART.                                  02625000 T  0003!024!0
                          3) TEST IF SPECIAL CHARACTER REQUIRES SPECIAL            02626000 T  0003!024!0
                               PROCESSING - OTHERWISE GET ELBAT WORD FROM           02627000 T  0003!024!0
                               SPECIAL.                                            02628000 T  0003!024!0
                         C) LOAD ELBAT AND INCREMENT NXTELBT.                      02629000 T  0003!024!0
                         D) IF ELBAT IS FULL ADJUST ELBAT, NXTELBT, I, AND P.      02630000 T  0003!024!0
                         E) GO BACK TO I).                                         02631000 T  0003!024!0
                      III) RETURN WITH CLASS OF ELBAT[P].                          02632000 T  0003!024!0
                    FURTHER DETAILS ARE GIVEN IN BODY OF TABLE,                    02633000 T  0003!024!0
                ;                                                                  02634000 T  0003!024!0
               INTEGER PROCEDURE TABLE(P); VALUE P; INTEGER P;                     02635000 T  0003!024!0
PRT(435) = TABLE
                BEGIN                                                              02636000 T  0003!024!0
                LABEL PERCENT,SPECIALCHAR,COMPLETE,COLON,DOT,ATSIGN,QUOTE,         02637000 T  0003!024!0
```

START OF SEGMENT ********** 24

```
                      STRNGXT,MOVEIT,ARGH,FINISHNUMBER,                            02638000 T  0024!000!0
                      SCANAGAIN,FPART,EPART,IPART,IDENT,ROSE,COMPOST,DOLLAR,RTPAREN,02639000 T  0024!000!0
                      CROSSHATCH,NUMBEREND;                                        02640000 T  0024!000!0
                SWITCH SPECIALSWITCH:=PERCENT,DOLLAR,DOT,ATSIGN,COLON,QUOTE,       02641000 T  0024!000!0
                                      RTPAREN,CROSSHATCH;                          02642000 T  0024!000!1
                SWITCH RESULTSWITCH:=IDENT,SPECIALCHAR,IPART;                      02643000 T  0024!000!2
                WHILE P ≥ NXTELBT                                                  02644000 T  0024!001!1
                DO   BEGIN                                                         02645000 T  0024!001!2
SCANAGAIN:                                                                         02646000 T  0024!001!0
                      COUNT:=RESULT:=ACCUM[1]:=0; SCANNER;                         02647000 T  0024!001!2
                      GO RESULTSWITCH[RESULT];                                     02648000 T  0024!001!3
ARGH:                                                                              02649000 T  0024!001!3
                      Q:=ACCUM[1]; FLAG(141); GO SCANAGAIN;                        02650000 T  0024!002!0
SPECIALCHAR:                                                                       02651000 T  0024!002!0
                      GT1:=ACCUM[1].[18:6] = 2;                                    02652000 T  0024!002!2
                      ENDTOG:=GT1 = 57 AND ENDTOG;                                 02653000 T  0024!002!2
COMMENT               OBTAIN ACTUAL CHARACTER FROM ACCUM;                          02654000 T  0024!002!1
                      T:=SPECIAL[GT1&GT1[42:41:3]];                                02655000 T  0024!002!1
COMMENT               NOTICE COMPRESSION TECHNIQUE USED TO SHORTEN TABLE OF         02656000 T  0024!002!1
                      ELBAT WORDS FOR SPECIAL CHARACTERS;                          02657000 T  0024!002!1
                      IF GT1:=T.INCR = 0 THEN GO COMPLETE;                         02658000 T  0024!002!1
                      GO SPECIALSWITCH[GT1];                                       02659000 T  0024!003!2
COMMENT               INCR FIELD OF SPECIAL CHARACTER IS NON-ZERO FOR SPECIAL       02660000 T  0024!003!2
                      CHARACTERS REQUIRING SPECIAL HANDLING.  INCR IS SWITCHED     02661000 T  0024!003!2
                      ON TO OBTAIN DISCRIMINATION;                                 02662000 T  0024!003!2
COLON:                RESULT:=7; SCANNER; COMMENT ELIMINATE BLANKS - CHECKING       02663000 T  0024!003!2
                                                  FOR := IN PLACE OF ← ;           02664000 T  0024!003!0
                      IF EXAMIN (NCR) = "=" THEN                                   02665000 T  0024!003!0
```

```
            BEGIN RESULT:=0; SCANNER; T:=SPECIAL[13]  END;                                  02666000 T  0024:003610
        RESULT:=2; GO COMPLETE;                                                             02667000 T  0024:003811
DOT:                                                                                        02668000 T  0024:004010
        IF EXAMIN(NCR)>9 OR ENDTOG THEN GO COMPLETE;                                        02680000 T  0024:004010
        NHI:=NLO:=0;                                                                        02681000 T  0024:004312
        C:=0; GO FPART;                                                                     02682000 T  0024:004410
ATSIGN:                                                                                     02683000 T  0024:004513
        RESULT:=0; SCANNER;          % SCAN PAST "@".                                    02684000 T  0024:004610
        IF COUNT>17 THEN GO ARGH;  % 16 CHARS. + "@".                                       02685000 T  0024:004712
        IF OCTIZE(ACCUM[1],C,17-COUNT,COUNT=1) THEN                                         02686000 T  0024:004811
            BEGIN  Q:=ACCUM[1]; FLAG(521); GO SCANAGAIN  END;                               02686500 T  0024:005113
        GO NUMBEREND;                                                                       02687000 T  0024:005411
COMMENT  DOT AND ATSIGN ENTER NUMBER CONVERSION AT CORRECT SPOT;                            02689000 T  0024:005512
QUOTE:                                                                                      02690000 T  0024:005512
        COUNT:=0;                                                                           02691000 T  0024:005512
        T:=IF STREAMTOG THEN 63                                                             02692000 T  0024:005513
    ELSE IF REAL(STREAMTOG)>1 THEN 8 ELSE 7;                                                02692500 T  0024:005610
        DO   BEGIN                                                                          02693000 T  0024:006010
            RESULT:=5; SCANNER;                                                             02694000 T  0024:006010
            IF COUNT>T THEN                                                                 02695000 T  0024:006112
                BEGIN  Q:=ACCUM[1]; FLAG(520); GO SCANAGAIN  END;                           02696000 T  0024:006210
            END UNTIL EXAMIN(NCR) = """;                                                    02697000 T  0024:006411
        Q:=ACCUM[1]; RESULT:=5; SCANNER; COUNT:=COUNT-1;                                    02698000 T  0024:006712
        IF COUNT<0 THEN COUNT:=COUNT+64;                                                    02699000 T  0024:007011
        ACCUM[1]:=Q; RESULT:=4;                                                             02700000 T  0024:007312
STRNGXT: T:=C:=0;                                                                           02701000 T  0024:007512
        IF COUNT < 8 THEN                                                                   02703000 T  0024:007610
MOVEIT:                                                                                     02704000 T  0024:007712
        MOVECHARACTERS(COUNT,ACCUM[1],3,C,8-COUNT);                                         02705000 T  0024:007810
        T.CLASS:=STRNGCON;                                                                  02705100 T  0024:008011
        GO COMPLETE;                                                                        02705200 T  0024:008210
COMMENT  CROSSHATCH HANDLES TWO SITUATIONS:                                                 02707000 T  0024:008211
        THE CROSSHATCH AT END OF DFFINE DECLARATIONS AND                                    02708000 T  0024:008211
        THE CROSSHATCH AT END OF ALPHA REPRESENTING DEFINED IDS.                            02709000 T  0024:008211
        THE TWO CASES ARE PROCESSED DIFFERENTLY.  THE FIRST CASE                            02710000 T  0024:008211
        MERELY PLACES THE CROSSHATCH IN ELBAT.  THE SECOND CASE                             02711000 T  0024:008211
        CAUSES AN EXIT FROM SCANNING THE ALPHA FOR THE DEFINED ID.                          02712000 T  0024:008211
        FOR A FULL DISCUSSION SEE DEFINEGEN;                                                02713000 T  0024:008211
CROSSHATCH:                                                                                 02714000 T  0024:008211
        IF DEFINECTR≠0 THEN GO COMPLETE;                                                    02715000 T  0024:008312
        PUTSEQNO(GT1,LCR);                                                                  02716000 T  0024:008410
        TURNONSTOPLIGHT(0,LCR);                                                             02717000 T  0024:008512
        IF DEFINEINDEX = 0 THEN GO ARGH;                                                    02718000 T  0024:008610
        LCR:=(GT1:=DEFINEARRAY[DEFINEINDEX-1]) DIV 262144;                                  02719000 T  0024:008713
        NCR:=GT1 MOD 262144;                                                                02720000 T  0024:009010
        GT2:=0&(T:=DEFINEARRAY[DEFINEINDEX:=DEFINEINDEX-3])[33:18:15];                      02721000 T  0024:009112
        LASTUSED:=T.[33:15];                                                                02722000 T  0024:009410
        FOR GT1:=1 STEP 1 UNTIL GT2 DO                                                      02723000 T  0024:009513
            BEGIN                                                                           02723500 T  0024:009810
            STACKHEAD[(T:=TAKE(LASTINFO+1)),[12:36] MOD 125]:=                              02724000 T  0024:009810
                        TAKE(LASTINFO).LINK;                                                02725000 T  0024:010112
            LASTINFO:=(NEXTINFO:=LASTINFO)-T.PURPT;                                         02726000 T  0024:010211
            END;                                                                            02727000 T  0024:010512
        GO SCANAGAIN;                                                                       02728000 T  0024:010712
DOLLAR:  COMMENT THIS CODE HANDLES CONTROL CARDS;                                           02729000 T  0024:010713
        DOLLARCARD;                                                                         02730000 T  0024:010810
PERCENT: IF NCR ≠ FCR THEN READACARD;                                                       02731000 T  0024:010811
```

```
          GO SCANAGAIN;                                                                                    02737000 T  0024:011011
COMMENT   MOST PERCENT SIGNS ACTING AS END OF CARD SENTINELS GET TO                                        02738000 T  0024:011112
          PERCENT.  PERCENT READS THE NEXT CARD AND STARTS OVER.  A                                        02739000 T  0024:011112
          SIDE EFFECT IS THAT ALL CHARACTERS ON A CARD ARE IGNORED                                         02740000 T  0024:011112
          AFTER A FREE PERCENT SIGN (ONE NOT IMBEDDED IN A STRING OR                                       02741000 T  0024:011112
          COMMENT);                                                                                        02742000 T  0024:011112
COMMENT   MIGHT BE FUNNY COMMA - HANDLE HERE;                                                              02743000 T  0024:011112
RTPAREN:  RESULT:=7; SCANNER;                                                                              02744000 T  0024:011112
          IF EXAMIN(NCR) = """ THEN                                                                        02745000 T  0024:011312
               BEGIN                                                                                       02746000 T  0024:011512
               RESULT:=0; SCANNER;                                                                         02747000 T  0024:011513
               DO   BEGIN                                                                                  02748000 T  0024:011611
                    RESULT:=5; SCANNER                                                                     02749000 T  0024:011712
                    END UNTIL EXAMIN(NCR) = """;                                                           02750000 T  0024:011713
               RESULT:=0; SCANNER;                                                                         02751000 T  0024:012011
               RESULT:=7; SCANNER;                                                                         02752000 T  0024:012113
               IF EXAMIN(NCR) ≠ "(" THEN GO ARGH;                                                          02753000 T  0024:012312
               RESULT:=0; SCANNER;  Q:=ACCUM[1];                                                           02754000 T  0024:012512
               T:=SPECIAL[24]                                                                              02755000 T  0024:012713
               END;                                                                                        02756000 T  0024:012713
          RESULT:=2; GO COMPLETE;                                                                          02757000 T  0024:012811
!PART:    TCOUNT:=0; C:=CONVERT;                                                                           02758000 T  0024:012913
%         RESULT:=7; SCANNER; % DEBLANK.                                                                   02759000 T  0024:013113
%         IF DEFINECTR=0 THEN                                                                              02760000 T  0024:013113
%          IF (C=3 OR C=4) AND EXAMIN(NCR)="""" THEN %OCTAL OR HEX STRING.                                 02761000 T  0024:013113
%               BEGIN    INTEGER SIZ;                                                                     02762000 T  0024:013113
%               RESULT:=5; SCANNER; % SKIP QUOTE.                                                        02763000 T  0024:013113
%               COUNT:=0;                                                                                02764000 T  0024:013113
%               DO   BEGIN                                                                               02765000 T  0024:013113
%                    RESULT:=5; SCANNER;                                                                 02766000 T  0024:013113
%                    IF COUNT > SIZ:=48 DIV C THEN % > 1 WORD LONG.                                      02767000 T  0024:013113
%                         BEGIN  ERR(520); GO SCANAGAIN  END;                                            02768000 T  0024:013113
%                    END UNTIL EXAMIN(NCR)="""";                                                         02769000 T  0024:013113
%               Q:=ACCUM[1]; RESULT:=5; SCANNER; COUNT:=COUNT-1;                                         02770000 T  0024:013113
%               IF C=3 THEN % OCTAL STRING.                                                              02771000 T  0024:013113
%                 IF OCTIZE(ACCUM[1],ACCUM[4],16-COUNT,COUNT) THEN                                       02772000 T  0024:013113
%                   FLAG(521) % NON OCTAL CHARACTER IN STRING.                                           02773000 T  0024:013113
%         ELSE ELSE IF HEXIZE(ACCUM[1],ACCUM[4],12-COUNT,COUNT) THEN                                       02774000 T  0024:013113
%                    FLAG(521); % NON CHARACTER IN HEX STRING.                                           02775000 T  0024:013113
%               IF COUNT < SIZ THEN                                                                      02776000 T  0024:013113
%                    BEGIN                                                                               02777000 T  0024:013113
%                    C:=ACCUM[4]; GO FINISHNUMBER;                                                       02778000 T  0024:013113
%                    END;                                                                                02779000 T  0024:013113
%               T.INCR:=COUNT:=8; T.CLASS:=STRING;                                                       02780000 T  0024:013113
%               MOVECHARACTERS(8,ACCUM[4],0,ACCUM[1],3);                                                 02781000 T  0024:013113
%               GO COMPLETE;                                                                             02782000 T  0024:013113
%               END OCTAL OR HEX STRING;                                                                 02783000 T  0024:013113
          IF DPTOG THEN                                                                                    02784000 T  0024:013113
               BEGIN NHI:=THI; NLO:=TLO; END;                                                              02785000 T  0024:013210
          IF EXAMIN(NCR)="," THEN                                                                          02786000 T  0024:013410
               BEGIN                                                                                       02787000 T  0024:013513
               RESULT:=0; SCANNER;                                                                         02788000 T  0024:013610
               C:=1.0× C;                                                                                  02789000 T  0024:013713
FPART:         TCOUNT:=COUNT;                                                                            02790000 T  0024:013811
               IF EXAMIN(NCR)≤9 THEN                                                                       02791000 T  0024:013913
                    BEGIN                                                                                  02792000 T  0024:014113
                    RESULT:=0; SCANNER;                                                                    02793000 T  0024:014210
```

```
                    IF DPTOG THEN                                                                  02794000 T  0024:014:2
                         BEGIN                                                                     02795000 T  0024:014:3
                         DOUBLE(CONVERT,TLO,TEN[(COUNT-TCOUNT)MOD 12],                             02796000 T  0024:014:0
                                    0,/,:=,THI,TLO);                                               02797000 T  0024:014:2
                         FOR T:=12 STEP 12 UNTIL COUNT = TCOUNT DO                                 02798000 T  0024:014:1
                           DOUBLE(THI,TLO,TEN[12],0,/,:=,THI,TLO);                                 02799000 T  0024:015:2
                         DOUBLE(THI,TLO,NHI,NLO,+,:=,NHI,NLO);                                     02800000 T  0024:015:0
                         C:=NHI                                                                    02801000 T  0024:016:0
                         END                                                                       02802000 T  0024:016:0
                    ELSE C:=TEN[TCOUNT-COUNT]×CONVERT+C;                                           02803000 T  0024:016:2
                    END                                                                            02804000 T  0024:016:4
               END;                                                                                02805000 T  0024:016:1
          RESULT:=7; SCANNER;                                                                      02806000 T  0024:016:1
          IF EXAMIN(NCR)="@" THEN                                                                  02807000 T  0024:016:1
               BEGIN                                                                               02808000 T  0024:016:0
               RESULT:=0; SCANNER;                                                                 02809000 T  0024:016:7
EPART:         TCOUNT:=COUNT;                                                                      02810000 T  0024:016:8
               C:=C×1,0;                                                                           02811000 T  0024:016:9
               RESULT:=7; SCANNER;                                                                 02812000 T  0024:017:0
               IF T:=EXAMIN(NCR)>9 THEN                                                            02813000 T  0024:017:2
                    BEGIN                                                                          02815000 T  0024:017:5
                    RESULT:=0; SCANNER;                                                            02816000 T  0024:017:6
                    TCOUNT:=COUNT;                                                                 02817000 T  0024:017:7
                    END;                                                                           02818000 T  0024:017:8
               RESULT:=0; SCANNER;                                                                 02820000 T  0024:017:8
               Q:=ACCUM[1];                                                                        02822000 T  0024:017:9
               IF GT1:=T:=(IF T="-"THEN -CONVERT ELSE CONVERT)<=46 OR                              02823000 T  0024:018:0
                 T>69 THEN FLAG(269)                                                               02824000 T  0024:018:5
               ELSE BEGIN                                                                          02825000 T  0024:018:6
                    T:=TEN[T];                                                                     02826000 T  0024:018:9
                    IF ABS(0&C[42:3:6]&C[1:2:1]+0&T[42:3:6]&T[1:2:1]                               02827000 T  0024:019:1
                      + 12) >63 THEN FLAG(269)                                                     02828000 T  0024:019:5
               ELSE IF DPTOG THEN                                                                  02829000 T  0024:019:7
                      IF GT1<0 THEN                                                                02830000 T  0024:019:8
                         BEGIN                                                                     02831000 T  0024:019:9
                         GT1:=-GT1;                                                                02832000 T  0024:020:0
                         DOUBLE(NHI,NLO,TEN[GT1 MOD 12],0,/,:=,NHI,NLO);                           02833000 T  0024:020:1
                         FOR GT2:=12 STEP 12 UNTIL GT1 DO                                          02834000 T  0024:020:5
                           DOUBLE(NHI,NLO,TEN[12],0,/,:=,NHI,NLO);                                 02835000 T  0024:020:6
                         END                                                                       02836000 T  0024:021:1
                    ELSE BEGIN                                                                     02837000 T  0024:021:1
                         DOUBLE(NHI,NLO,TEN[GT1 MOD 12],0,×,:=,NHI,NLO);                           02838000 T  0024:021:2
                         FOR GT2:=12 STEP 12 UNTIL GT1 DO                                          02839000 T  0024:021:5
                           DOUBLE( NHI,NLO,TEN[12],0,×,:=,NHI,NLO);                                02840000 T  0024:021:7
                         END                                                                       02841000 T  0024:022:1
               ELSE C:=C×T;                                                                        02842000 T  0024:022:1
                    END;                                                                           02843000 T  0024:022:4
               END;                                                                                02844000 T  0024:022:4
NUMBEREND:                                                                                         02845000 T  0024:022:4
          Q:=ACCUM[1]; RESULT:=3;                                                                  02846000 T  0024:022:5
FINISHNUMBER:                                                                                      02847000 T  0024:022:6
          T:=0;                                                                                    02848000 T  0024:022:7
          IF C.[1:37]=0 THEN                                                                       02849000 T  0024:022:7
               BEGIN   T.CLASS:=LITNO ; T.ADDRESS:=C  END                                          02850000 T  0024:022:9
          ELSE T.CLASS:=NONLITNO ;                                                                 02851000 T  0024:023:3
          GO COMPLETE;                                                                             02852000 T  0024:023:5
COMMENT   THE CODE BETWEEN IDENT AND COMPOST DOES A LOOKUP IN INFO.                                02853000 T  0024:023:6
```

```
          IF QUANTITY IS NOT FOUND THE ELBAT WORD EXPECTS TO BE                    02854000 T  0024:023:0
          ZERO.  THE SCRAMBLE FOR APPROPRIATE STACK IS FIRST THING                 02855000 T  0024:023:0
          TO BE DONE. THEN THE LOOP BETWEEN COMPOST AND                            02859000 T  0024:023:0
          ROSE IS ENTERED.  THE LAST THING DONE FOR ANY                            02860000 T  0024:023:0
          IDENTIFIER WHICH IS FOUND IS TO STUFF THE LOCATION                       02861000 T  0024:023:0
          OF THE ELBATWORD IN INFO INTO THE LINK FIELD.  THIS                      02862000 T  0024:023:0
          ALLOWS REFERENCE BACK TO INFO FOR ADDITIONAL DATA.                       02863000 T  0024:023:0
          SHOULD THIS BE REQUIRED.                                ;                02864000 T  0024:023:0
IDENT:    T:=STACKHEAD[SCRAM:=(Q:=ACCUM[1])MOD 125];                               02865000 T  0024:023:0
ROSE:     GT1:=T.LINKR;                                                            02875000 T  0024:023:3
          IF(GT2:=T.LINKC)+GT1= 0 THEN                                             02876000 T  0024:024:2
               BEGIN T:=0; GO COMPLETE END;                                        02877000 T  0024:024:3
        IF T = INFO[GT1, GT2] THEN BEGIN                                  %101-  02877010 C  0024:024:2
          T:= 0; GO TO COMPLETE END;                                      %101-  02877020 C  0024:024:3
          T:=INFO[GT1,GT2];                                                        02878000 T  0024:024:1
          IF INFO[GT1,GT2+1]&0[1:1:11] ≠ Q THEN GO ROSE;                           02879000 T  0024:025:1
          IF COUNT ≤ 5 THEN GO COMPOST ;                                           02880000 T  0024:025:0
          IF NOT EQUAL(COUNT-5,ACCUM[2],INFO[GT1,GT2+2])THEN GO ROSE;              02881000 T  0024:025:3
COMPOST:  T:=T&GT1[35:43:5]&GT2[40:40:8];                                          02882000 T  0024:026:0
          IF GT1 ≠1 AND NOT MACROID THEN % NOT RESERVED WORD              %110-  02882100 C  0024:026:1
            XREFIT(T.LINK,CARDNUMBER,NORMALREF); % BUILD XREF ENTRY        %110-  02882200 C  0024:026:0
COMMENT   CHECK HERE FOR COMMENTS AND DEFINED IDS;                                 02883000 T  0024:026:3
          IF NOT ENDTOG THEN                                                       02884000 T  0024:026:3
               BEGIN                                                               02885000 T  0024:026:0
               IF GT1:=T.CLASS = COMMENTV THEN                                     02886000 T  0024:026:1
                    BEGIN                                                          02887000 T  0024:027:1
                    WHILE EXAMIN(NCR) ≠ "]" DO                                     02888000 T  0024:027:2
                         BEGIN RESULT:=6; COUNT:=0; SCANNER END;                   02889000 T  0024:027:2
                    RESULT:=0;SCANNER;GO SCANAGAIN                                 02890000 T  0024:027:3
                    END                                                            02891000 T  0024:027:3
               END;                                                                02892000 T  0024:027:3
          IF STOPDEFINE THEN GO COMPLETE;                                          02893000 T  0024:027:3
          IF GT1 ≠ DEFINEDID THEN GO COMPLETE;                                     02894000 T  0024:027:1
COMMENT   SETUP FOR DEFINED IDS - SEE DEFINEGEN FOR MORE DETAILS;                  02895000 T  0024:027:3
          IF T.ADDRESS≠0 THEN T:=FIXDEFINEINFO(T);                                 02896000 T  0024:027:3
          IF DEFINEINDEX = 24 THEN                                                 02898000 T  0024:028:1
               BEGIN FLAG(139);GO ARGH END;                                        02899000 T  0024:028:3
          DEFINEARRAY[DEFINEINDEX]:=LASTUSED&T.ADDRESS [18:33:15];                 02900000 T  0024:028:2
          LASTUSED:=GIT(T);                                                        02901000 T  0024:028:3
          DEFINEARRAY[DEFINEINDEX+2]:=262144×LCR+NCR;                              02902000 T  0024:028:1
          LCR:=(NCR:=MKABS(DEFINEARRAY[DEFINEINDEX+1]))+1;                         02903000 T  0024:029:3
          PUTSEQNO(GT4,LCR);                                                       02904000 T  0024:029:2
          TURNONSTOPLIGHT("%",LCR); DEFINEINDEX:=DEFINEINDEX+3;                    02905000 T  0024:029:0
          GO PERCENT;                                                              02906000 T  0024:029:0
COMPLETE:                                                                          02909000 T  0024:030:0
          ELBAT[NXTELBT]:=T;                                                       02910000 T  0024:030:0
     IF NOT DEFINING THEN                                                  %108-  02910100 C  0024:030:2
     IF T.CLASS = BEGINV THEN                                              %108-  02910200 C  0024:030:3
          BEGINSTACK[BSPOINT:=BSPOINT+1]:=CARDNUMBER ELSE                  %108-  02910300 C  0024:030:3
     IF T.CLASS = ENDV THEN                                                %108-  02910400 C  0024:030:0
   BEGIN                                                                   %108-  02910500 C  0024:030:0
          IF LISTER THEN IF BEND THEN BEGINPRINT;                          %108-  02910600 C  0024:030:1
          BSPOINT:=BSPOINT - REAL(BSPOINT > 0); % PREVENT INVALID INDEX           02910700 C  0024:031:2
   END;                                                                    %108-  02910800 C  0024:031:1
          STOPDEFINE:=FALSE; COMMENT ALLOW DEFINES AGAIN;                          02911000 T  0024:031:1
          IF NXTELBT:=NXTELBT+1 > 74 THEN                                          02912000 T  0024:031:3
            IF NOT MACROID THEN                                                    02913000 T  0024:031:2
```

```
                    BEGIN                                                                    02914000 T  0024:036:0
COMMENT  ELBAT IS FULL;  ADJUST IT;                                                          02915000 T  0024:036:1
                    MOVE(10,ELBAT[65],ELBAT);                                                02916000 T  0024:036:1
                    I:=I-65; P:=P-65; NXTELBT:=10;                                           02917000 T  0024:038:1
                    END                                                                      02918000 T  0024:032:3
               END;                                                                          02919000 T  0024:032:3
          IF TABLE:=ELBAT[P].CLASS = COMMENTV THEN                                           02920000 T  0024:032:0
               BEGIN                                                                         02921000 T  0024:032:2
COMMENT  SPECIAL HANDLING OF CONSTANTS FOR SAKE OF FOR STATEMENTS;                           02922000 T  0024:032:3
               C:=INFO[0,ELBAT[P].ADDRESS];                                                  02923000 T  0024:032:3
               ELBAT[P].CLASS:=TABLE:=NONLITNO                                               02924000 T  0024:032:0
               END;                                                                          02925000 T  0024:032:2
          STOPDEFINE:=FALSE;  COMMENT ALLOW DEFINE;                                          02926000 T  0024:033:2
          END TABLE ;                                                                        02927000 T  0024:033:3
                                                                                    24 IS  337 LONG, NEXT SEG    3


               BOOLEAN PROCEDURE BOOLPRIM;        FORWARD;                                   02955000 T  0003:024:0
PRT(436) = BOOLPRIM
               PROCEDURE BOOLCOMP(B); BOOLEAN B; FORWARD;                                 02955500 T  0003:024:0
PRT(437) = BOOLCOMP
               INTEGER PROCEDURE NEXT;                                                    02956000 T  0003:024:0
PRT(440) = NEXT
                    BEGIN                                                                 02956500 T  0003:024:0
                    LABEL EXIT;                                                           02957000 T  0003:024:0
                                                                                 START OF SEGMENT **********   25
                    INTEGER T;                                                            02957500 T  0025:000:0
STACK(F+3) = T
                    DEFINE ERROR = BEGIN  FLAG(603); GO EXIT  END#;                       02958000 T  0025:000:0
                    SKAN;                                                                 02958500 T  0025:000:0
                    IF RESULT=3 THEN ERROR; % NUMBERS NOT ALLOWED.                        02959000 T  0025:000:3
                    IF RESULT=2 THEN % SPECIAL CHARACTER.                                 02959500 T  0025:000:6
                         BEGIN                                                            02960000 T  0025:000:2
                         T:=IF Q="1,0000" OR Q="1%0000" THEN 20 % FAKE OUT BOOLEXP.       02960500 T  0025:000:3
                            ELSE ((T:=Q.[18:6]-2) & T[42:41:3]);                          02961000 T  0025:001:0
                         IF T=11 OR T=19 OR T=20 THEN BATMAN:=SPECIAL[T] % (,),OR ;       02961500 T  0025:001:3
                    ELSE FLAG(603);                                                       02962000 T  0025:001:2
                         GO EXIT                                                          02962500 T  0025:002:3
                         END SPECIAL CHARACTERS;                                          02963000 T  0025:002:0
COMMENT  LOOK FOR BOOLEAN OPERATORS, THEN OPTIONS;                                        02963500 T  0025:002:0
                    T:=      IF Q="3NOT00" THEN NOTOP                                     02964000 T  0025:002:0
                       ELSE IF Q="3AND00" THEN ANDOP                                      02964500 T  0025:002:3
                       ELSE IF Q="2OR000" THEN OROP                                       02965000 T  0025:002:5
                       ELSE IF Q="3EQV00" THEN EQVOP                                      02965500 T  0025:002:7
                       ELSE 0;                                                            02966000 T  0025:002:9
                    IF T#0 THEN BATMAN.CLASS:=T                                           02966500 T  0025:003:2
               ELSE BATMAN:=1 & BOOID[2:7] & REAL(FINDOPTION(1))[1:1]; % OPTION.          02967000 T  0025:003:1
EXIT:                                                                                     02967500 T  0025:004:0
                    NEXT:=MYCLASS:=BATMAN.CLASS;                                          02968000 T  0025:004:2
               END NEXT;                                                                  02968500 T  0025:004:1
                                                                                    25 IS   48 LONG, NEXT SEG    3


               BOOLEAN PROCEDURE BOOLEXP;                                                 02969000 T  0003:024:0
```

```
                    BEGIN                                                                    02969500 T  0003:024210
                    BOOLEAN B;                                                               02970000 T  0003:024210
                                                                                           START OF SEGMENT **********   26
STACK(F+3) = B
                    B:=BOOLPRIM;                                                             02970500 T  0026:000010
                    WHILE MYCLASS≥EQVOP AND MYCLASS≤ANDOP DO BOOLCOMP(B);                    02971000 T  0026:000112
                    BOOLEXP:=B                                                               02971500 T  0026:000411
                    END BOOLEXP;                                                             02972000 T  0026:000411
                                                                                           26 IS    8 LONG, NEXT SEG    3


                BOOLEAN PROCEDURE BOOLPRIM;                                                  02972500 T  0003:024210
                    BEGIN                                                                    02973000 T  0003:024210
                    BOOLEAN B,KNOT;                                                          02973500 T  0003:024210
                                                                                           START OF SEGMENT **********   27
STACK(F+3) = B
STACK(F+4) = KNOT
                    DEFINE SKIPIT = MYCLASS:=NEXT #;                                         02974000 T  0027:000010
                    IF KNOT:=(NEXT=NOTOP) THEN SKIPIT;                                       02974500 T  0027:000010
                    IF MYCLASS=LEFTPAREN THEN                                                02975000 T  0027:000312
                        BEGIN                                                                02975500 T  0027:000313
                        B:=BOOLEXP;                                                          02976000 T  0027:000410
                        IF MYCLASS≠RTPAREN THEN FLAG(604);                                   02976500 T  0027:000512
                        END                                                                  02977000 T  0027:000712
                ELSE IF MYCLASS≠BOOID THEN FLAG(601)                                         02977500 T  0027:000712
                ELSE B:=BATMAN<0;                                                            02978000 T  0027:000912
                    IF KNOT THEN B:=NOT B; SKIPIT;                                           02978500 T  0027:001113
                    BOOLPRIM:=B                                                              02979000 T  0027:001410
                    END BOOLPRIM;                                                            02979500 T  0027:001410
                                                                                           27 IS   18 LONG, NEXT SEG    3


                PROCEDURE BOOLCOMP(B); BOOLEAN B;                                            02980000 T  0003:024210
                    BEGIN                                                                    02980500 T  0003:024210
                    REAL OPCLASS;                                                            02981000 T  0003:024210
                                                                                           START OF SEGMENT **********   28
STACK(F+2) = OPCLASS
                    BOOLEAN T;                                                               02981500 T  0028:000010
STACK(F+3) = T
                    OPCLASS:=MYCLASS;                                                        02982000 T  0028:000010
                    T:=BOOLPRIM;                                                             02982500 T  0028:000011
                    WHILE OPCLASS<MYCLASS DO BOOLCOMP(T);                                    02983000 T  0028:000113
                    B:=     IF OPCLASS=ANDOP THEN (B AND T)                                  02983500 T  0028:000411
                       ELSE IF OPCLASS=OROP  THEN (B OR T)                                   02984000 T  0028:000610
                       ELSE                       (B EQV T);                                 02984500 T  0028:000811
                    END BOOLCOMP;                                                            02985000 T  0028:001112
                                                                                           28 IS   14 LONG, NEXT SEG    3


                %                                                                            02985500 T  0003:024210
                COMMENT##################################################################    02986000 T  0003:024210
                            FORWARD DECLARATIONS                                             02986500 T  0003:024210
```

```
                ##########################################################################;          02987000 T  0003!024210
                %                                                                                  02987500 T  0003!024210
                 PROCEDURE AEXP; FORWARD;                                                          03001000 T  0003!024210
PRT(441) = AEXP
                 PROCEDURE ARITHSEC; FORWARD;                                                      03002000 T  0003!024210
PRT(442) = ARITHSEC
                 PROCEDURE SIMPARITH; FORWARD;                                                     03003000 T  0003!024210
PRT(443) = SIMPARITH
                 PROCEDURE ARITHCOMP; FORWARD;                                                     03004000 T  0003!024210
PRT(444) = ARITHCOMP
                 PROCEDURE PRIMARY; FORWARD;                                                       03005000 T  0003!024210
PRT(445) = PRIMARY
                 DEFINE BEXP = AEXP#;                                                              03006000 T  0003!024210
                 INTEGER PROCEDURE EXPRSS; FORWARD;                                                03007000 T  0003!024210
PRT(446) = EXPRSS
                 PROCEDURE POLISHER(EXPECT); VALUE EXPECT; REAL EXPECT; FORWARD;                   03009000 T  0003!024210
PRT(447) = POLISHER
                 PROCEDURE INLINE; FORWARD;                                                        03010000 T  0003!024210
PRT(450) = INLINE
                 PROCEDURE SUBHAND(FROM); VALUE FROM; BOOLEAN FROM; FORWARD;                       03011000 T  0003!024210
PRT(451) = SUBHAND
                 PROCEDURE IOSTMT; FORWARD;                                                        03012000 T  0003!024210
PRT(452) = IOSTMT
                 INTEGER PROCEDURE IFEXP; FORWARD;                                                 03013000 T  0003!024210
PRT(453) = IFEXP
                 PROCEDURE PARSE; FORWARD;                                                         03014000 T  0003!024210
PRT(454) = PARSE
                 PROCEDURE DOT; FORWARD;                                                           03015000 T  0003!024210
PRT(455) = DOT
                 PROCEDURE  IFCLAUSE; FORWARD;                                                     03018000 T  0003!024210
PRT(456) = IFCLAUSE
                INTEGER PROCEDURE GET(SYLLABLE);VALUE SYLLABLE; REAL SYLLABLE; FORWARD;            03019000 T  0003!024210
PRT(457) = GET
                INTEGER PROCEDURE GNAT(L); VALUE L; REAL L; FORWARD;                               03020000 T  0003!024210
PRT(460) = GNAT
                 PROCEDURE PANA; FORWARD;                                                          03021000 T  0003!024210
PRT(461) = PANA
                 PROCEDURE IFSTMT; FORWARD;                                                        03022000 T  0003!024210
PRT(462) = IFSTMT
                 PROCEDURE GOGEN(LABELBAT,BRANCHTYPE);                                             03023000 T  0003!024210
PRT(463) = GOGEN
                          VALUE LABELBAT,BRANCHTYPE;                                               03024000 T  0003!024210
                          REAL LABELBAT,BRANCHTYPE; FORWARD;                                       03025000 T  0003!024210
                 BOOLEAN PROCEDURE SIMPGO; FORWARD;                                                03026000 T  0003!024210
PRT(464) = SIMPGO
                 PROCEDURE STMT; FORWARD;                                                          03027000 T  0003!024210
PRT(465) = STMT
                 PROCEDURE EMIT(SYLLABLE); VALUE SYLLABLE; REAL SYLLABLE; FORWARD;                 03028000 T  0003!024210
PRT(466) = EMIT
                 PROCEDURE PROCSTMT(FROM); VALUE FROM; BOOLEAN FROM; FORWARD;                      03029000 T  0003!024210
PRT(467) = PROCSTMT
                 PROCEDURE STRMPROCSTMT; FORWARD;                                                  03030000 T  0003!024210
PRT(470) = STRMPROCSTMT
                 PROCEDURE CONSTANTCLEAN; FORWARD;                                                 03034000 T  0003!024210
PRT(471) = CONSTANTCLEAN
                 PROCEDURE SCATTERELBAT; FORWARD;                                                  03035000 T  0003!024210
PRT(472) = SCATTERELBAT
```

```
                PROCEDURE EMITB(BRANCH,FROM,TOWARDS); VALUE BRANCH,FROM,TOWARDS;                    03036000 T  0003!024!0
PRT(473) = EMITB
                        INTEGER BRANCH,FROM,TOWARDS; FORWARD;                                      03037000 T  0003!024!0
                PROCEDURE VARIABLE(FROM); INTEGER FROM; FORWARD;                                   03038000 T  0003!024!0
PRT(474) = VARIABLE
                PROCEDURE IMPFUN; FORWARD;                                                         03039000 T  0003!024!0
PRT(475) = IMPFUN
              PROCEDURE RIGHT(L); VALUE L; INTEGER L; FORWARD;                                     03039500 T  0003!024!0
PRT(476) = RIGHT
                PROCEDURE STREAMSTMT; FORWARD;                                                     03040000 T  0003!024!0
PRT(477) = STREAMSTMT
              PROCEDURE SEGMENTSTART(B);VALUE B;BOOLEAN B;FORWARD;                                 03041000 T  0003!024!0
PRT(500) = SEGMENTSTART
              PROCEDURE SEGMENT(SIZE,FR); VALUE SIZE,FR; INTEGER SIZE,FR; FORWARD;                 03042000 T  0003!024!0
PRT(501) = SEGMENT

                                                                                                   03043000 T  0003!024!0
                                                                                                   03044000 T  0003!024!0
                                                                                                   03045000 T  0003!024!0
              INTEGER PROCEDURE BAE; FORWARD;                                                      03046000 T  0003!024!0
PRT(502) = BAE
                    PROCEDURE PROGDESCBLDR(A,B,C,D);VALUE A,B,C,D;                                 03047000 T  0003!024!0
PRT(503) = PROGDESCBLDR
                                INTEGER A,C,D; BOOLEAN B; FORWARD;                                 03047100 T  0003!024!0
                PROCEDURE BANA; FORWARD;                                                           03048000 T  0003!024!0
PRT(504) = BANA
                PROCEDURE EMITNUM(A); VALUE A; REAL A; FORWARD;                                    03049000 T  0003!024!0
PRT(505) = EMITNUM
                PROCEDURE EMITD(A,B,T); VALUE A,B,T; INTEGER A,B,T; FORWARD;                       03050000 T  0003!024!0
PRT(506) = EMITD
              INTEGER PROCEDURE GETSPACE(S,L); VALUE S,L;                                          03051000 T  0003!024!0
PRT(507) = GETSPACE
                                              INTEGER L; BOOLEAN S; FORWARD;                       03051001 T  0003!024!0
                PROCEDURE FORSTMT; FORWARD;                                                        03052000 T  0003!024!0
PRT(510) = FORSTMT
              REAL PROCEDURE TAKE(INDEX); VALUE INDEX; INTEGER INDEX; FORWARD;                     03053000 T  0003!024!0
              PROCEDURE E;  FORWARD;                                                               03054000 T  0003!024!0
PRT(511) = E
              PROCEDURE ENTRY(TYPE); VALUE TYPE;REAL TYPE; FORWARD;                                03055000 T  0003!024!0
PRT(512) = ENTRY
              PROCEDURE PUTNBUMP(P1);  VALUE P1;  REAL P1;  FORWARD;                               03057000 T  0003!024!0
PRT(513) = PUTNBUMP
              PROCEDURE JUMPCHKNX;  FORWARD;                                                       03058000 T  0003!024!0
PRT(514) = JUMPCHKNX
              PROCEDURE JUMPCHKX;  FORWARD;                                                        03059000 T  0003!024!0
PRT(515) = JUMPCHKX
              PROCEDURE DBLSTMT; FORWARD;                                                          03060000 T  0003!024!0
PRT(516) = DBLSTMT
              PROCEDURE BLOCK(S); VALUE S; BOOLEAN S; FORWARD;                                     03067000 T  0003!024!0
PRT(517) = BLOCK
              PROCEDURE PURGE(STOPPER);VALUE STOPPER;REAL STOPPER;FORWARD;                         03068000 T  0003!024!0
PRT(520) = PURGE
              PROCEDURE ENTER(TYPEV);                                                              03069000 T  0003!024!0
PRT(521) = ENTER
                        VALUE TYPEV;                                                               03070000 T  0003!024!0
              REAL TYPEV;FORWARD;                                                                  03071000 T  0003!024!0
                                                                                                   03072000 T  0003!024!0
                                                                                                   03073000 T  0003!024!0
```

```
                                                                                03074000 T  0003:0242:0
                                                                                03075000 T  0003:0242:0
                    COMMENT THIS SECTION CONTAINS THE EMITTERS.  THEY ARE THE AGENTS WHICH      04000000 T  0003:0242:0
                            ACTUALLY PRODUCE CODE AND DEBUGING OUTPUT;                          04001000 T  0003:0242:0
                    COMMENT EMITL EMITS A LIT CALL;                                             04002000 T  0003:0242:0
                    PROCEDURE EMITL(LITERAL); VALUE LITERAL; INTEGER LITERAL;                   04003000 T  0003:0242:0
PRT(522) = EMITL
                            EMIT(0&LITERAL[36:38:10]);                                          04004000 T  0003:0242:0


                    COMMENT EMITO EMIT AN OPERATOR;                                             04005000 T  0003:0244:0
                    PROCEDURE EMITO(OPERATOR); VALUE OPERATOR; INTEGER OPERATOR;                04006000 T  0003:0244:0
PRT(523) = EMITO
                            EMIT(1&OPERATOR[36:38:10]);                                         04007000 T  0003:0244:0


                    COMMENT EMITC IS PRIMARILY FOR USE BY STRMSTMT TO EMIT CHARACTOR MODE       04008000 T  0003:0246:0
                            OPERATORS.  HOWEVER IT ALSO HANDLES DIA, DIB, AND TRB;              04009000 T  0003:0246:0
                  PROCEDURE EMITC(REPEAT,OPERATOR); VALUE REPEAT,OPERATOR;                      04010000 T  0003:0246:0
PRT(524) = EMITC
                            INTEGER REPEAT,OPERATOR;                                            04011000 T  0003:0246:0
                            BEGIN                                                               04012000 T  0003:0246:0
                              IF REPEAT≥64 THEN FLAG(268);                                      04013000 T  0003:0246:0
                              EMIT(OPERATOR&REPEAT[36:42:6]) END EMITC;                         04014000 T  0003:0248:0
```

```
                    COMMENT EMITV EMITS AN OPERAND CALL.  IF THE ADDRESS IS FOR THE SECOND      04015000 T  0003:0250:0
                            HALF OF THE PRT, THEN IT ALSO EMITS A PRTE;                         04016000 T  0003:0250:0
                    PROCEDURE EMITV(ADDRESS); VALUE ADDRESS; INTEGER ADDRESS;                   04017000 T  0003:0250:0
PRT(525) = EMITV
                            BEGIN IF ADDRESS > 1023 THEN EMITO(PRTE);                           04018000 T  0003:0250:0
                                 EMIT(2 & ADDRESS [36:38:10]) END EMITV;                        04019000 T  0003:0252:0


                    COMMENT EMITN EMITS A DESCRIPTOR CALL.  IF THE ADDRESS IS FOR THE           04020000 T  0003:0254:0
                            SECOND HALF OF THE PRT, THEN IT ALSO EMITS A PRTE;                  04021000 T  0003:0254:0
                    PROCEDURE EMITN(ADDRESS); VALUE ADDRESS; INTEGER ADDRESS;                   04022000 T  0003:0254:0
PRT(526) = EMITN
                            BEGIN IF ADDRESS > 1023 THEN EMITO(PRTE);                           04023000 T  0003:0254:0
                                 EMIT(3 & ADDRESS [36:38:10]) END EMITN;                        04024000 T  0003:0256:0


                    COMMENT EMITPAIR EMITS A LITC ADDRESS FOLLOWED BY OPERATOR.  IF THE         04025000 T  0003:0258:0
                            ADDRESS IS FOR THE SECOND HALF OF THE PRT, THEN IT ALSO             04026000 T  0003:0258:0
                            EMITS PRTE;                                                         04027000 T  0003:0258:0
                    PROCEDURE EMITPAIR(ADDRESS,OPERATOR);                                       04028000 T  0003:0258:0
PRT(527) = EMITPAIR
```

```
                        VALUE ADDRESS,OPERATOR;                                          04029000 T  0003:025:0
                        INTEGER ADDRESS,OPERATOR;                                        04030000 T  0003:025:0
                        BEGIN                                                            04031000 T  0003:025:0
                             EMITL(ADDRESS);                                             04032000 T  0003:025:0
                             IF ADDRESS > 1023 THEN EMITO(PRTE);                         04033000 T  0003:025:1
                             EMITO(OPERATOR) END EMITPAIR;                               04034000 T  0003:026:1


          COMMENT ADJUST ADJUST L TO THE BEGINING OF A WORD AND FILLS IN THE             04080000 T  0003:026:3
                  INERVENING SPACE WITH NOPS, IT CHECKS STREAMTOG TO DECIDE              04081000 T  0003:026:3
                  WHICH SORT OF NOP TO USE;                                              04082000 T  0003:026:3
          PROCEDURE ADJUST;                                                              04083000 T  0003:026:3
PRT(530) = ADJUST
                  BEGIN                                                                  04084000 T  0003:026:3
                                                                                         04085000 T  0003:026:3
        WHILE L.[46:2]#0 DO EMIT(45);                                                    04086000 T  0003:026:3
                  END ADJUST;                                                            04087000 T  0003:026:5:2


          PROCEDURE EMITLNG;                                                             04098000 T  0003:026:5:2
PRT(531) = EMITLNG
                  BEGIN LABEL E;                                                         04099000 T  0003:026:5:2
                                                                                  START OF SEGMENT **********   29
                    IF NOT LINKTOG THEN GO TO E;                                         04100000 T  0029:000:0
             COMMENT  GO TO E IF LAST THING IS A LINK;                                   04101000 T  0029:000:1
                    IF GET(L) # 0 THEN GO TO E;                                          04102000 T  0029:000:1
             COMMENT  EITHER LAST EXPRESSION WAS CONDITIONAL OR THERE IS NO              04103000 T  0029:000:2:1
                    LNG OR RELATIONAL OPERATOR;                                          04104000 T  0029:000:2:1
                    IF GT1 ← GET(L-1) = 77 THEN L ← L-1                                  04105000 T  0029:000:2:1
             COMMENT LAST THING WAS AN LNG - SO CANCEL IT;                               04106000 T  0029:000:5:3
                ELSE IF GT1.[42:6]=21 AND GT1.[37:2]=0 THEN % AHA                        04107000 T  0029:000:5:3
             COMMENT LAST THING WAS A RELATIONAL;                                        04108000 T  0029:000:9:3
                      BEGIN L←L-1; EMITO(REAL(BOOLEAN(GT1.[36:10]) EQV                   04109000 T  0029:000:9:3
                        BOOLEAN(IF GT1.[40:2] = 0 THEN 511 ELSE 463)))                   04110000 T  0029:001:2:1
             COMMENT NEGATE THE RELATIONAL; END ELSE                                     04111000 T  0029:001:5:3
           E:         EMITO(LNG) END EMITLNG;                                          04112000 T  0029:001:5:3
                                                                                    29 IS    18 LONG, NEXT SEG    3


             COMMENT   EMITB EMITS A BRANCH OPERATOR AND ITS ASSOCIATED NUMBER;          04113000 T  0003:026:5:2
          PROCEDURE EMITB(BRANCH,FROM,TOWARDS);                                          04114000 T  0003:026:5:2
                    VALUE BRANCH,FROM,TOWARDS;                                           04115000 T  0003:026:5:2
                    INTEGER BRANCH,FROM,TOWARDS;                                         04116000 T  0003:026:5:2
                    BEGIN                                                                04117000 T  0003:026:5:2
                      INTEGER TL;                                                        04118000 T  0003:026:5:2
                                                                                  START OF SEGMENT **********   30
STACK(F+2) = TL
                      TL ← L;                                                            04119000 T  0030:000:0
                    IF TOWARDS > FOULED THEN FOULED ← TOWARDS;                           04119500 T  0030:000:1
                      L ← FROM-2;                                                        04120000 T  0030:000:2:1
                      GT1 ← TOWARDS-FROM;                                                04120100 T  0030:000:4:0
                      IF TOWARDS.[46:2] = 0                                              04120200 T  0030:000:5:2
```

```
                    THEN BEGIN                                                    04120300 T  0030:0006:0
                     BRANCH ← BRANCH&1[39:47:1];                                  04120400 T  0030:0007:2
                     GT1 ← TOWARDS DIV 4 - (FROM-1) DIV 4 END;                    04120500 T  0030:0008:1
                  EMITNUM(ABS(GT1));                                              04121000 T  0030:0011:3
                  EMITO(BRANCH&(REAL(GT1≥ 0)+1)[42:46:2]);                        04122000 T  0030:0012:1
                                                                                  04123000 T  0030:0015:2
                  L ← TL                                                          04124000 T  0030:0015:2
                END EMITB;                                                        04125000 T  0030:0015:2
                                                                        30 IS   19 LONG, NEXT SEG    3


          COMMENT   DEBUGWORD FORMATS TWO FIELDS FOR DEBUGGING OUTPUT IN          04126000 T  0003:0265:2
                    OCTAL, NAMELY :                                               04127000 T  0003:0265:2
                        1. 4 CHARACTERS FOR THE L REGISTER,                       04128000 T  0003:0265:2
                        2.16 CHARACTERS FOR THE WORD BEING EMITTED.   ;           04129000 T  0003:0265:2
      STREAM  PROCEDURE DEBUGWORD( SEQ,CODE,FEIL); VALUE SEQ,CODE ;               04130000 T  0003:0265:2
PRT(532) = DEBUGWORD
                        BEGIN                                                     04131000 T  0003:0265:2
                          DI←FEIL; SI← LOC SEQ; SI← SI+4; DS ← 4 CHR;             04132000 T  0003:0266:0
                          DS ← 2 LIT"  ";                                         04133000 T  0003:0267:2
                          SI ← LOC CODE ;                                         04134000 T  0003:0267:3
                          16(  DS ← 3 RESET; 3( IF SB THEN DS←SET ELSE            04135000 T  0003:0267:3
                               DS ← RESET ; SKIP 1 SB));                          04136000 T  0003:0269:3
                        29(DS ← 2 LIT"  " );                                      04137000 T  0003:0270:1
                        END ;                                                     04138000 T  0003:0271:3


          COMMENT   EMITWORD PLACES THE PARAMETER,"WORD",INTO EDOC. IF            04139000 T  0003:0271:3
                    DEBUGGING IS REQUIRED, "L" AND "WORD" ARE OUTPUT ON           04140000 T  0003:0271:3
                    THE  PRINTER FILE IN OCTAL FORMAT.     ;                      04141000 T  0003:0271:3
        PROCEDURE  EMITWORD (WORD); VALUE WORD; REAL WORD;                        04142000 T  0003:0271:3
PRT(533) = EMITWORD
                        BEGIN                                                     04143000 T  0003:0271:3
                        ADJUST;                                                   04144000 T  0003:0271:3
                        IF L≥ 4088 THEN BEGIN ERR(200); L←0; END                  04145000 T  0003:0272:1
                            ELSE BEGIN                                            04146000 T  0003:0275:2
                    MOVE(1,WORD, CODE(L DIV 4+1));                                04147000 T  0003:0277:2
                            IF DEBUGTOG THEN                                      04148000 T  0003:0282:0
                            BEGIN DEBUGWORD(B2D(L),WORD,LIN);                     04149000 T  0003:0282:1
                                WRITELINE  END;                                   04150000 T  0003:0285:2
                            FOULED ← L ← L+4; END                                 04151000 T  0003:0295:3
                        END  EMITWORD;                                            04152000 T  0003:0297:2


          COMMENT CONSTANTCLEAN IS CALLED AFTER AN UNCONDITIONAL BRANCH HAS       04153000 T  0003:0297:3
                  BEEN EMITTED. IF ANY CONSTANTS HAVE BEEN ACCUMULATED BY         04154000 T  0003:0297:3
                  EMITNUM IN INFO[0,*], CONSTANTCLEAN WILL FIX  THE CHAIN         04155000 T  0003:0297:3
                  OF  C-RELATIVE  OPDC S LEFT BY EMITNUM, IF  C-RELATIVE          04156000 T  0003:0297:3
                  ADDRESSING IS IMPOSSIBLE (I.E. THE ADDRESS                      04157000 T  0003:0297:3
                  IF GREATER THAN 127 WORDS) THEN THE CONSTANT ALONG WITH         04158000 T  0003:0297:3
                  THE  1ST  LINK OF THE OPDC CHAIN IS ENTERED IN INFO.            04159000 T  0003:0297:3
```

```
                        AT  PURGE  TIME THE REMAINING OPDC S ARE EMITTED WITH          04160000 T  0003:029:3
                        F -RELATIVE ADDRESSING AND CODE EMITTED TO STORE THE          04161000 T  0003:029:3
                        CONSTANTS INTO THE PROPER F-RELATIVE CELLS,  ;                04162000 T  0003:029:3
                PROCEDURE CONSTANTCLEAN ;                                             04163000 T  0003:029:3
                        IF MRCLEAN THEN                                               04164000 T  0003:029:3
                          BEGIN                                                       04165000 T  0003:029:0
                            INTEGER J,TEMPL,D,LINK;                                   04166000 T  0003:029:1
PRT(534) = *SEGMENT DESCRIPTOR*

                                                                        START OF SEGMENT **********   31
STACK(F+2) = J
STACK(F+3) = TEMPL
STACK(F+4) = D
STACK(F+5) = LINK
                            BOOLEAN CREL;                                         04167000 T  0031:000:0
STACK(F+6) = CREL
                            LABEL  ALLTHU ;                                       04168000 T  0031:000:0
                            FOR J ← 1 STEP 2 UNTIL LASTENTRY DO                   04169000 T  0031:000:0
                              BEGIN                                               04170000 T  0031:000:0
                             ADJUST; TEMPL←L; L←INFO[0,255-J+1];                  04171000 T  0031:000:2
                                CREL ← FALSE;                                     04172000 T  0031:000:2
                                DO BEGIN                                          04173000 T  0031:000:2
                                     IF D←(TEMPL-L+3)DIV 4≥128 THEN               04174000 T  0031:000:3
                             IF MODE ≠ 0 THEN                                     04175000 T  0031:000:0
                BEGIN  FLAG(50);       GO TO ALLTHU END;                         04175500 T  0031:000:1
                                                                                  04176000 T  0031:001:0
                                                                                  04177000 T  0031:001:3
                                                                                  04178000 T  0031:001:3
                                LINK←GET(L);                                      04179000 T  0031:001:3
                                CREL ← TRUE;                                      04180000 T  0031:001:3
                                  IF MODE ≠ 0 THEN EMITV(D+768) ELSE              04181000 T  0031:001:3
                                  EMITV(REAL(TEMPL≥2048)×1024+TEMPL DIV 4);       04182000 T  0031:001:3
                              END  UNTIL L←   LINK   = 4095  ;                    04183000 T  0031:001:2
                    ALLTHU:       L ← TEMPL;                                      04184000 T  0031:001:3
                             IF  CREL  THEN  EMITWORD( INFO[0,255-J ]);           04184500 T  0031:001:0
                            END;                                                  04185000 T  0031:001:3
                          LASTENTRY ← 0;                                          04186000 T  0031:002:2
                        END ;                                                     04187000 T  0031:002:1
PRT(535) = *SEGMENT DESCRIPTOR*                                                   04188000 T  0031:002:3
                                                                                  04189000 T  0031:002:0
                                                                                  04190000 T  0031:002:1
                                                                        31 IS   33 LONG, NEXT SEG    3

                COMMENT   EMITNUM HANDLES THE EMISSION OF CODE FOR CONSTANTS,BOTH     04191000 T  0003:030:2
                          EXPLICIT AND IMPLICIT, IN EVERY CASE,EMITNUM WILL           04192000 T  0003:030:2
                          PRODUCE CODE TO GET THE DESIRED CONSTANT ON TOP OF          04193000 T  0003:030:2
                          THE STACK. IF THE NUMBER IS A LITERAL A SIMPLE LITC         04194000 T  0003:030:2
                          SYLLABLE IS PRODUCED. HOWEVER,NON-LITERALS ARE KEPT         04195000 T  0003:030:2
                          IN THE ZERO-TH ROW OF INFO WITH THE SYLLABLE                04196000 T  0003:030:2
                          POSITION,L. THE FIRST EMITNUM ON A PARTICULAR               04197000 T  0003:030:2
                          CONSTANT CAUSES THE VALUES OF L AND THE CONSTANT            04198000 T  0003:030:2
                          TO BE STORED IN INFO[0,*] (NOTE:ITEMS ARE STORED            04199000 T  0003:030:2
                          IN REVERSE STARTING WITH INFO[0,255],ETC.). THEN            04200000 T  0003:030:2
                          ITS THE JOB OF CONSTANTCLEAN TO EMIT THE ACTUAL             04201000 T  0003:030:2
```

```
                    OPDC (SEE CONSTANTCLEAN PROCEDURE FOR DETAILS) ;                    04202000 T  0003!030!2
          PROCEDURE EMITNUM( C ); VALUE C; REAL C;                                     04203000 T  0003!030!2
                    BEGIN LABEL FINISHED,FOUND ; REAL N;                               04204000 T  0003!030!2
                                                                                  START OF SEGMENT **********   32
STACK(F+2) = N
                    IF C.[1:37]=0 THEN EMITL(C)                                        04205000 T  0032!000!0
                    ELSE                                                               04206000 T  0032!000!1
                         BEGIN                                                         04207000 T  0032!000!2
                         FOULED ← L;                                                   04207500 T  0032!000!3
                         FOR  N ← 1 STEP 2 UNTIL LASTENTRY DO                          04208000 T  0032!000!0
                      IF INFO[0,255-N] = C THEN  GO TO FOUND ;                         04209000 T  0032!000!2
                         INFO[0,255 -LASTENTRY] ← L;                                   04210000 T  0032!001!0
                         INFO[0,255 -LASTENTRY-1]← C ;                                 04211000 T  0032!001!1
                         EMITN(1023);                                                  04212000 T  0032!001!3
                         IF MODE=0 THEN EMITO(NOP);                                    04212100 T  0032!001!0
                         LINKTOG←FALSE;                                                04213000 T  0032!001!0
                         IF LASTENTRY ← LASTENTRY+2 ≥ 128  THEN                        04214000 T  0032!001!2
                              BEGIN                                                    04215000 T  0032!002!1
                              C ← BUMPL;                                               04216000 T  0032!002!2
                              CONSTANTCLEAN;                                           04217000 T  0032!002!2
                              EMITB(BFW,C,L);                                          04218000 T  0032!002!3
                              END;                                                     04219000 T  0032!002!1
                         GO TO FINISHED;                                               04220000 T  0032!002!1
               FOUND:    EMIT(INFO[0,255 -N+1]);                                       04221000 T  0032!002!2
                         LINKTOG←FALSE;                                                04222000 T  0032!002!1
                         INFO[0,255-N+1]← L-1;                                         04223000 T  0032!002!3
                         IF MODE=0 THEN EMITO(NOP);                                    04223100 T  0032!003!2
                         END;                                                          04224000 T  0032!003!2
           FINISHED:END  EMITNUM  ;                                                    04225000 T  0032!003!2
                                                                                 32 IS   38 LONG, NEXT SEG    3


                    COMMENT   SEARCH PERFORMS A BINARY SEARCH ON THE COP AND WOP       04226000 T  0003!030!2
                              ARRAYS. GIVEN THE OPERATOR BITS SEARCH  YIELDS THE BCD   04227000 T  0003!030!2
                              MNEUMONIC FOR THAT OPERATOR. IF THE OPERATOR CANNOT      04228000 T  0003!030!2
                              BE FOUND SEARCH YIELDS BLANKS.                           04229000 T  0003!030!2
                              NOTE: DIA,DIB,TRB ARE RETURNED AS BLANKS.        ;       04230000 T  0003!030!2
                   ALPHA PROCEDURE   SEARCH (Q,KEY); VALUE KEY; ARRAY Q[0]; REAL KEY ; 04231000 T  0003!030!2
PRT(536) = SEARCH
                    BEGIN  LABEL L;                                                    04232000 T  0003!030!2
                                                                                  START OF SEGMENT **********   33
                    COMMENT   GT1 AND GT2 ARE INITIALIZED ASSUMMING THAT Q IS ORDERED  04233000 T  0033!000!0
                              BY PAIRS (ARGUMENT,FUNCTION,ARGUMENT,FUNCTION,ETC.)      04234000 T  0033!000!0
                              AND THAT THE FIRST ARGUMENT IS IN Q[4].  FURTHERMORE     04235000 T  0033!000!0
                              THE LENGTH OF Q IS 128.                          ;       04236000 T  0033!000!0
                              INTEGER N,I ;                                            04237000 T  0033!000!0
STACK(F+3) = N
STACK(F+4) = I
                    N ← 64 ;                                                           04238000 T  0033!000!0
                    FOR I ← 66 STEP IF Q[I]<KEY THEN N ELSE - N                        04239000 T  0033!000!1
                              WHILE N←N DIV 2 ≥ 1 DO                                   04240000 T  0033!000!2
                    IF Q[ I ]= KEY THEN GO TO L ;                                      04241000 T  0033!000!2
                     I ←0;   COMMENT ARGUMENT NOT FOUND,SEARCH=Q[1] ;                  04242000 T  0033!001!2
                  L: SEARCH←Q[  I +1] ;                                                04243000 T  0033!001!3
                    END       SEARCH  ;                                                04244000 T  0033!001!3
```

```
                COMMENT   B2D  CONVERTS THE FOUR LOW ORDER OCTAL DIGITS TO BCD                    04245000 T  0003:030:2
                          CODE ;                                                                 04246000 T  0003:030:2
          ALPHA   PROCEDURE   B2D(B); VALUE B; REAL B;                                           04247000 T  0003:030:2
                          B2D←0&B[45:45:3]&B[39:42:3]&B[33:39:3]&B[27:36:3] ;                    04248000 T  0003:030:2



                COMMENT   PACK IS A STREAM PROCEDURE WHICH INSERTS THE SYLLABLE                  04265000 T  0003:031:2
                          INTO THE EDOC ARRAY.   THE SPECIFIC ELEMENT OF EDOC                    04266000 T  0003:031:2
                          IS  PRECISILY = EDOC[(L DIV 4) DIV 128,(< DIV 4)MOD 128]               04267000 T  0003:031:2
                          SYLLABLE  POSITION=(L MOD 4 ), WHERE L IS THE SYLLABLE                 04268000 T  0003:031:2
                          NUMBER RELATIVE TO THE BEGINNING OF THE SEGMENT;                       04269000 T  0003:031:2
          STREAM PROCEDURE PACK(WORD,POSITION,SYLLABLE);                                         04270000 T  0003:031:2
PRT(537) = PACK
                          VALUE POSITION,SYLLABLE;                                               04271000 T  0003:031:2
                          BEGIN                                                                  04272000 T  0003:031:2
                          DI←WORD ; DI ← DI+POSITION ; DI ← DI+POSITION;                         04273000 T  0003:032:0
                          SI←LOC SYLLABLE ; SI←SI+6;                                             04274000 T  0003:033:2
                          DS←2 CHR ;                                                             04275000 T  0003:033:3
                          END  PACK  ;                                                           04276000 T  0003:034:0
```

```
                     COMMENT DEBUG PRINTS OUT OBJECT CODE IF "DEBUGN" IS SET;                    04277000 T  0003:034:0
                     PROCEDURE DEBUG(S); VALUE S; REAL S;                                        04277500 T  0003:034:0
PRT(540) = DEBUG
                       BEGIN  REAL T1;                                                           04278000 T  0003:034:0
                                                                                      START OF SEGMENT **********   34
STACK(F+2) = T1
                       IF SINGLTOG THEN                                                          04278500 T  0034:000:0
                         WRITE(LINE,BUG,B2D(L),                                                  04279000 T  0034:000:1
PRT(541) = *LIST, LABEL, OR SEGMENT DESCRIPTOR*
                              IF STREAMTOG THEN                                                  04279500 T  0034:000:0
                                 SEARCH(COP,S,[42:6])                                            04280000 T  0034:000:2
                            ELSE IF T1:=S.[46:2]=1 THEN SEARCH(WOP,S,[36:10])                    04280500 T  0034:001:1
                            ELSE WOP[T1] , IF STREAMTOG THEN                                     04281000 T  0034:001:2
                           B2D(S,[36:6]) ELSE IF T1=1 THEN WOP[1]                                04281500 T  0034:001:2
                      ELSE B2D(S,[36:10]),B2D(S))                                                04282000 T  0034:002:2
                     ELSE WRITE(LINE,BUG,B2D(L),                                                 04282500 T  0034:002:2
PRT(542) = *LIST, LABEL, OR SEGMENT DESCRIPTOR*
                              IF STREAMTOG THEN                                                  04283000 T  0034:003:0
                                 SEARCH(COP,S,[42:6])                                            04283500 T  0034:003:2
                            ELSE IF T1:=S.[46:2]=1 THEN SEARCH(WOP,S,[36:10])                    04284000 T  0034:004:1
                            ELSE WOP[T1] , IF STREAMTOG THEN                                     04284500 T  0034:004:2
                           B2D(S,[36:6]) ELSE IF T1=1 THEN WOP[1]                                04285000 T  0034:004:2
                      ELSE B2D(S,[36:10]),B2D(S));                                               04285500 T  0034:005:2
                       END DEBUG;                                                                04286000 T  0034:006:2
                                                                                       34 IS   64 LONG, NEXT SEG   3
```

```
   COMMENT    EMIT PLACES SYLLABLES INTO EDOC, CALLS DEBUG FOR                    04288000 T  0003!031410
              DEBUGGING OUTPUT ON THE PRINTER, AND CHECKS FOR SEGMENTS            04289000 T  0003!031410
              GREATER THAN 4093 SYLLABLES.        ;                              04290000 T  0003!031410
PROCEDURE EMIT ( S ) ; VALUE S; REAL   S  ;                                       04291000 T  0003!031410
              BEGIN                                                               04292000 T  0003!031410
              IF L < 4088 THEN                                                    04293000 T  0003!031410
              BEGIN                                                               04294000 T  0003!031513
              LINKTOG ← TRUE;                                                     04295000 T  0003!031610
         PACK(CODE(L DIV 4+1),L,[46:2],S);                                        04296000 T  0003!031712
              IF DEBUGTOG THEN DEBUG(S);                                          04297000 T  0003!032211
              L←L+1;                                                              04298000 T  0003!032411
              END  ELSE                                                           04299000 T  0003!032513
              BEGIN ERR(200); L←1; END;                                           04300000 T  0003!032513
   COMMENT    200 EMIT - SEGMENT GREATER THAN 4093 SYLLABLES          *;          04301000 T  0003!032913
              END  EMIT ;                                                         04302000 T  0003!032913


   COMMENT    EMITD EMITS THE DIA,DIB,TRB SEQUENCE OF CODE, THE                   04305000 T  0003!032913
              PREVIOUS SETTING OF THE G-H AND K-V REGISTERS IS COMPARED            04306000 T  0003!032913
              THE CURRENT . IF THE G-H,K-V OR BOTH ARE ALREADY SET THEN           04307000 T  0003!032913
              THE APPROIATE SYLLABLE(S) ARE OMITTED                               04308000 T  0003!032913
              IF 0 BITS ARE TO BE TRANSFERED THEN NO SYLLABLES ARE                04309000 T  0003!032913
              EMITTED     ;                                                       04310000 T  0003!032913
PROCEDURE     EMITD(A,B,T); VALUE A,B,T ; INTEGER A,B,T;                          04311000 T  0003!032913
         BEGIN LABEL EXIT,NORMAL;                                                 04311010 T  0003!032913
                                                                  START OF SEGMENT **********   35
               REAL Q;                                                            04311020 T  0035!000010
STACK(F+2) = Q
               IF T = 15 THEN                                                     04311030 T  0035!000010
               BEGIN                                                              04311040 T  0035!000011
               IF A = 33 THEN Q ← 512                                             04311050 T  0035!000112
               ELSE IF A ≠ 18 THEN GO TO NORMAL;                                  04311060 T  0035!000211
               IF B = 18 THEN Q ← Q+256                                           04311070 T  0035!000512
               ELSE IF B ≠ 33 THEN GO TO NORMAL;                                  04311080 T  0035!000611
               EMITO(Q+197);  COMMENT -- THIS GETS OUT FIXED FIELD;               04311090 T  0035!000912
               GO TO EXIT;                                                        04311100 T  0035!001011
               END;                                                               04311110 T  0035!001112
         NORMAL:                                                                  04311120 T  0035!001112
               IF  T≠ 0 THEN                                                      04312000 T  0035!001112
               BEGIN                                                              04313000 T  0035!001113
                                                                                  04314000 T  0035!001210
                EMIT(((DIALA+A) DIV 6)×512 + ( A← A MOD 6)× 64 + DIA);            04315000 T  0035!001210
                                                                                  04316000 T  0035!001712
                EMIT(((DIALB+B) DIV 6)×512 + ( B← B MOD 6)× 64 + DIB);            04317000 T  0035!001712
                EMIT(TRB+64×T);                                                   04318000 T  0035!002113
                                                                                  04319000 T  0035!002313
                                                                                  04320000 T  0035!002313
                                                                                  04321000 T  0035!002313
               END       EMITD  ;                                                 04322000 T  0035!002313
         EXIT: END;                                                               04322100 T  0035!002313
                                                                  35 IS   27 LONG, NEXT SEG    3


              PROCEDURE EMITI(E,A,B); VALUE E,A,B; REAL E,A,B;                    04500000 T  0003!032913
```

```
PRT(543) = EMITI

                    BEGIN LABEL EXIT,IS;                                                    04501000 T  0003!032913
                                                                                  START OF SEGMENT **********   36
                          INTEGER S,T1,T2;                                                  04502000 T  0036!000010

STACK(F+2) = S
STACK(F+3) = T1
STACK(F+4) = T2

                          PROCEDURE EMIT21(E,B); VALUE E,B;                                 04503000 T  0036!000010

PRT(544) = EMIT21
                              REAL E;                                                       04504000 T  0036!000010
                              BOOLEAN B;                                                    04505000 T  0036!000010
                              BEGIN IF E = 0 THEN                                           04506000 T  0036!000010
                                        BEGIN IF B THEN EMITO(XCH); END                     04507000 T  0036!000011
                                    ELSE BEGIN GT1 := E.ADDRESS;                            04508000 T  0036!000211
                                               IF E := E.CLASS ≤ INTID THEN                 04509000 T  0036!000411
                                                   EMITV(GT1)                               04521000 T  0036!000610
                                               ELSE IF E ≤ INTARRAYID THEN                  04522000 T  0036!000712
                                                   EMITPAIR(GT1,LOD) ELSE                   04523000 T  0036!000811
                                               EMITN(GT1)                                   04524000 T  0036!001010
                                          END                                               04525000 T  0036!001112
                              END;                                                          04526000 T  0036!001113

                              IF B = 0 THEN                                                 04526100 T  0036!001113
                                  BEGIN EMIT21(E,FALSE); GO TO EXIT END;                    04526200 T  0036!001211
                              IF STACKCT ≠ 0 THEN GO TO IS;                                 04527000 T  0036!001411
                              IF B = 15 THEN                                                04528000 T  0036!001610
                                  BEGIN IF A = 33 THEN                                      04529000 T  0036!001611
                                            BEGIN EMIT21(E,FALSE);                          04530000 T  0036!001810
                                                  EMIT(0); EMITO(INX);                      04531000 T  0036!001913
                                                  GO TO EXIT;                               04532000 T  0036!002112
                                            END;                                            04533000 T  0036!002113
                                        IF A = 18 THEN                                      04534000 T  0036!002113
                                            BEGIN EMIT(0);                                  04535000 T  0036!002210
                                                  EMIT21(E,TRUE);                           04536000 T  0036!002313
                                                  EMITO(197);                               04537000 T  0036!002411
                                                  GO TO EXIT;                               04538000 T  0036!002512
                                            END;                                            04539000 T  0036!002513
                                        GO TO IS;                                           04540000 T  0036!002513
                                  END;                                                      04541000 T  0036!002610
                              IF B ≤ 10 AND A+B = 48 THEN                                   04542000 T  0036!002610
                                  BEGIN EMIT21(E,FALSE);                                    04543000 T  0036!002811
                                        EMITL(2*B-1);                                       04544000 T  0036!003010

PRT(545) = LN
PRT(546) = EXP
PRT(547) = X TO THE I
                                        EMITO(LND);                                         04545000 T  0036!003312
                                        GO TO EXIT;                                         04546000 T  0036!003410
                                  END;                                                      04547000 T  0036!003411
                          IS:     IF (S := (48-A-B) MOD 6)+B ≤ 39 THEN                      04548000 T  0036!003411
                                  BEGIN EMIT21(E,FALSE);                                    04549000 T  0036!003810
                                        EMIT(T2+(T1+A DIV 6)×512+(A MOD 6)×64+DIA);         04550000 T  0036!003913
                                        EMIT(((A+B-1) DIV 6 -T1+1)×512+64×S+37);            04551000 T  0036!004411
                                        GO TO EXIT;                                         04552000 T  0036!004913
                                  END;                                                      04553000 T  0036!005010
```

```
                    EMIT(0);                                                                    04554000 T  0036:0050:0
                    EMIT21(E,TRUE);                                                             04555000 T  0036:0051:2
                    EMITD(A,48-B,B);                                                            04556000 T  0036:0052:0
                                                                                                04557000 T  0036:0053:3
                EXIT: END;                                                                      04558000 T  0036:0053:3
                                                                                          36 IS   57 LONG. NEXT SEG    3


            COMMENT THIS SECTION CONTAINS MISCELLANEOUS SERVICE ROUTINES;                       05000000 T  0003:0329:3
            COMMENT STEPI AND STEPIT ARE SHORT CALLS ON TABLE;                                  05001000 T  0003:0329:3
            PROCEDURE STEPIT; ELCLASS ← TABLE(I←I+1);                                           05002000 T  0003:0329:3
PRT(550) = STEPIT


            INTEGER PROCEDURE STEPI; STEPI←ELCLASS←TABLE(I←I+1);                                05003000 T  0003:0332:1
PRT(551) = STEPI


            COMMENT TAKE FETCHS A WORD FROM INFO;                                               05004000 T  0003:0338:0
            REAL PROCEDURE TAKE(INDEX); VALUE INDEX; INTEGER INDEX;                             05005000 T  0003:0338:0
                    TAKE ← INFO[INDEX.LINKR,INDEX.LINKC];                                       05006000 T  0003:0338:0


            COMMENT PUT PLACES A WORD INTO INFO;                                                05007000 T  0003:0344:0
            PROCEDURE PUT(WORD,INDEX); VALUE WORD,INDEX; REAL WORD,INDEX;                       05008000 T  0003:0344:0
PRT(552) = PUT
                    INFO[INDEX.LINKR,INDEX.LINKC] ← WORD;                                       05009000 T  0003:0344:0


            COMMENT FLAG FLAGS ERROR MESSAGES, COUNTS THEM AND SUPPRESS FUTURE                  05010000 T  0003:0348:0
                    ERROR MESSAGES UNTIL THE COMPILER THINKS IT HAS RECOVERED;                  05011000 T  0003:0348:0
            PROCEDURE FLAG(ERRNUM); VALUE ERRNUM; INTEGER ERRNUM;                               05012000 T  0003:0348:0
                    BEGIN                                                                       05013000 T  0003:0348:0
               COMMENT WRITERROR IS THE STREAM PROCEDURE WHICH ACTUALLY PRODUCES                 05014000 T  0003:0348:0
                    THE ERROR MESSAGE ON THE PRINTER;                                           05015000 T  0003:0348:0
                 STREAM PROCEDURE WRITERROR(ERRNUM,ACCUM,LINE,COUNT,LSTSEQ);                    05016000 T  0003:0348:0
                                                                                      START OF SEGMENT **********   37
PRT(553) = WRITERROR
                    VALUE ERRNUM,COUNT;                                                         05017000 T  0037:0000:0
                    BEGIN                                                                       05018000 T  0037:0000:0
                        DI ← LINE; 44(DS←2LIT" "); COMMENT CLEAR BUFFER;                        05019000 T  0037:0000:0
                        SI ←LSTSEQ; SI ← SI-8; DS ←WDS;                                         05020000 T  0037:0001:2
                        SI ← LINE; DS ← 2 WDS;                                                  05021000 T  0037:0002:0
                        4(DS ← 2 LIT "XX"); COMMENT  SET RIGHT MARGIN FLAG;                     05022000 T  0037:0002:1
                        SI ← LSTSEQ; DI ← LSTSEQ; DI ← DI-8; DS ← WDS;                          05023000 T  0037:0003:3
                        DI ← LINE; DI ← DI+8; COMMENT INDENT MESSAGE;                           05024000 T  0037:0004:1
                        DS ← 13 LIT "ERROR NUMBER ";                                            05025000 T  0037:0005:2
```

```
                    SI ← LOC ERRNUM; DS ← 3 DEC; COMMENT CONVERT ERRNUM;          05026000 T  0037:0007:2
                    DS ← 4 LIT " -- ";                                           05027000 T  0037:0007:3
                    SI ← ACCUM; SI ← SI+3; DS ← COUNT CHR;                       05028000 T  0037:0008:0
                       COMMENT PLACE ALPHA IN BUFFER;                            05029000 T  0037:0009:2
                    DS ← LIT ".";                                                05030000 T  0037:0009:2
                END WRITERROR;                                                   05031000 T  0037:0009:3



                IF ERRORTOG THEN % DO NOTHING IF WE SUPPRESS MSSGS.              05032000 T  0037:0010:0
                    BEGIN                                                        05033000 T  0037:0010:0
                    SPECTOG := FALSE;                                            05034000 T  0037:0010:1
                    ERRORCOUNT := ERRORCOUNT+1; COMMENT COUNT ERRORS;            05035000 T  0037:0011:3
                    IF NOT LISTER THEN                                           05036000 T  0037:0012:1
                      BEGIN                                                      05037000 T  0037:0013:2
                      EDITLINE(LIN,FCR,L DIV 4,L,[46:2],MEDIUM,0);               05038000 T  0037:0013:3
                     MOVE(1,INFO[LASTSEQROW,LASTSEQUENCE],LIN[12]);              05039500 T  0037:0017:2
                      IF NOHEADING THEN DATIME; WRITELINE;                       05039600 T  0037:0019:3
                      END;                                                       05041000 T  0037:0031:2
            COMMENT   PRINT CARDIMAGE IF WE ARE NOT LISTING;                     05042000 T  0037:0031:2
                    ACCUM[1] ← Q;    COMMENT RESTORE ACCUMULATOR;                05043000 T  0037:0031:2
                    WRITERROR(ERRNUM,ACCUM[1],LIN[*],Q,[12:6],                   05044000 T  0037:0032:0
                      INFO[LASTSEQROW,LASTSEQUENCE]);                            05045000 T  0037:0034:1
                IF NOT NOHEADING THEN WRITELINE ;                                05046000 T  0037:0036:0
                    ERRORTOG ← FALSE; COMMENT INHIBIT MESSAGES;                  05047000 T  0037:0047:2
            IF PUNCHTOG THEN                                                     05048000 T  0037:0048:0
              BEGIN REAL T1,T2,T3,T4; LABEL L,L1,EXIT;                           05049000 T  0037:0048:1
PRT(554) = *SEGMENT DESCRIPTOR*
```

START OF SEGMENT ********** 38

```
STACK(F+2) = T1
STACK(F+3) = T2
STACK(F+4) = T3
STACK(F+5) = T4
                STREAM PROCEDURE P1(L,P); VALUE P;                               05050000 T  0038:0000:0
PRT(555) = P1
                    BEGIN DI ← L; 15(DS←8LIT" ");                                05051000 T  0038:0000:0
                SI ← LOC P; DI←L; SI←SI+5; SKIP 3 SB;                            05052000 T  0038:0002:0
                5                                                                05053000 T  0038:0003:2
                (DS←3 RESET;3(IF SB THEN DS←SET ELSE DS←RESET;SKIP 1 SB));       05054000 T  0038:0003:2
                SI←P;                                                            05055000 T  0038:0005:3
                DI←DI+2; 2(8                                                     05056000 T  0038:0006:0
                (DS←3 RESET;3(IF SB THEN DS←SET ELSE DS←RESET;SKIP 1 SB));       05057000 T  0038:0006:1
                DS←LIT" ") END;                                                  05058000 T  0038:0009:2



                STREAM PROCEDURE P2(L,P); VALUE P;                               05059000 T  0038:0010:0
PRT(556) = P2
                    BEGIN DI←L; DI ←DI+26;DS←LIT"0";                             05060000 T  0038:0010:0
                       SI←P; SI←SI-2; SKIP 1SB;                                  05061000 T  0038:0012:0
                3                                                                05062000 T  0038:0012:1
                (DS←3 RESET;3(IF SB THEN DS←SET ELSE DS←RESET;SKIP 1 SB));       05063000 T  0038:0012:1
                END;                                                             05064000 T  0038:0015:3
```

```
                REAL STREAM PROCEDURE ABS(A); VALUE A;                                                    05065000 T  0038:0015:3
PRT(557) = ABS
                     BEGIN DI ← LOC ABS; SI ←A; DS← WDS;DI←DI+8; DS←RESET END;                             05066000 T  0038:0015:3



                STREAM PROCEDURE  BITEDUST(X,N,ID,LINE,COUNT); VALUE ID,N,COUNT;                          05067000 T  0038:0018:0
PRT(560) = BITEDUST
                 BEGIN  LOCAL  T,F,H;                                                                     05068000 T  0038:0018:0
                         DI←LOC F; SI←LINE; DS←WDS;DI←F;                                                  05069000 T  0038:0019:2
                SI←LOC [D;SI←SI+2;DS← 6CHR;  57(DS←2 LIT" ")];                                            05070000 T  0038:0020:0
                              SI ← LOC COUNT ;SI←SI+8 ;                                                   05071000 T  0038:0021:3
                              DI ←LOC H ; DS ←WDS ;                                                       05071100 T  0038:0022:0
                              DI ← LOC LINE ; SI ← H;                                                     05071200 T  0038:0022:1
                       SI←LOC X;SKIP 2 SB;                                                                05072000 T  0038:0023:2
                      IF SB THEN                                                                          05073000 T  0038:0023:3
                        BEGIN   SI←X;  T←SI;                                                              05074000 T  0038:0024:0
                                N(                                                                        05075000 T  0038:0025:2
                         DI←LOC F; SI←LINE; DS←WDS;DI←F;                                                  05076000 T  0038:0026:0
                             SI← LOC COUNT; SI←SI+6;                                                      05077000 T  0038:0027:2
                4                                                                                         05078000 T  0038:0027:3
                (DS←3 RESET;3(IF SB THEN DS←SET ELSE DS←RESET;SKIP 1 SB)));                               05079000 T  0038:0027:3
                              DS←2 LIT" "; SI←COUNT; SI←SI+48;COUNT←SI;                                   05080000 T  0038:0030:1
                SI←T;                                                                                     05081000 T  0038:0031:3
                        6(2(8                                                                             05082000 T  0038:0032:0
                (DS←3 RESET;3(IF SB THEN DS←SET ELSE DS←RESET;SKIP 1 SB)));                               05083000 T  0038:0032:1
                DS← LIT" ")); DS←LIT" " );                                                                05084000 T  0038:0035:2
                              DI ← LOC LINE ; SI ← H;                                                     05085000 T  0038:0036:1
                SI←T;SI←SI+48;T←SI));END END;                                                             05086000 T  0038:0037:2
```

```
                          BITEDUST(ERRORCOUNT, 63 ,"PRT    ",LINE, 21);                                   05087000 T  0038:0038:1
                FOR  T1← 0 STEP 1 UNTIL 31DO BITEDUST(INFO[ T1,*],43,"INFO  ",LINE,0);                    05088000 T  0038:0041:3
                BITEDUST(      ELBAT[*],13,"ELBAT ",LINE,0);                                              05089000 T  0038:0049:3
                      BITEDUST(STACKHEAD[*],21,"STHEAD",LINE,0);                                          05090000 T  0038:0052:1
                         T1←MKABS(ERRNUM)-1; T3←T1;                                                       05091000 T  0038:0055:3
                L:    T2←ABS(T3);                                                                         05092000 T  0038:0058:1
                      IF T2.[33:15]=0 THEN BEGIN T3←T2.[18:15];                                           05093000 T  0038:0060:1
                         IF T3=0 THEN GO TO EXIT ELSE GO TO L END ;                                       05094000 T  0038:0063:3
                   T4 ← IF T2.[33:15] < 512 THEN 0 ELSE T2.[33:15]&T2[30:10:2];                           05095000 T  0038:0065:3
                 L1:P1(LIN[0],T1);IF T1=T3 THEN BEGIN IF T#0 THEN P2(LIN[0],T4);                          05096000 T  0038:0070:0
                    T1←T1-1;                                                                              05097000 T  0038:0076:0
                     WRITELINE;                                                                           05097500 T  0038:0077:2
                                    T3←T2.[18:15];GO TO L  END;                                           05098000 T  0038:0091:2
                     T1:=T1-1; WRITELINE;                                                                 05099000 T  0038:0093:2
                     GO TO L1;                                                                            05099100 T  0038:0104:1
                EXIT; END ;                                                                               05100000 T  0038:0105:2
PRT(561) = *SEGMENT DESCRIPTOR*

                                    END END FLAG;                                                         38 IS  106 LONG, NEXT SEG   37
                                                                                                             05101000 T  0037:0050:0
                                                                                                          37 IS   53 LONG, NEXT SEG    3
```

```
                    LABEL ENDOFITALL;
COMMENT ERR, IS THE SAME AS FLAG EXCEPT THAT IT MAKES AN ATTEMPT TO
                    RECOVER FROM ERROR SITUATIONS BY SEARCHING FOR A
                    SEMICOLON, END, OR BEGIN;
PROCEDURE ERR(ERRNUM); VALUE ERRNUM; INTEGER ERRNUM;
                    BEGIN FLAG(ERRNUM);
                         I ← I-1;
                         IF ERRNUM = 200 THEN I := I/0; % SEGMENT TOO LARGE.
                         IF ERRNUM = 611 THEN I := I/0; % ERRMAX EXCEEDED.
                         DO IF STEP1 = BEGINV THEN STMT UNTIL
                            ELCLASS = ENDV OR ELCLASS = SEMICOLON END ERR;
```

```
05101100 T  0003:034810
05102000 T  0003:034810
05103000 T  0003:034810
05104000 T  0003:034810
05105000 T  0003:034810
05106000 T  0003:034810
05107000 T  0003:034913
05107100 P  0003:035112
05107200 P  0003:035313
05108000 T  0003:035610
05109000 T  0003:035810
```

```
DEFINE ERROR = ERR#;  COMMENT ERROR IS A SYNONM FOR ERR;
COMMENT CHECKER IS A SMALL PROCEDURE THAT CHECKS TO SEE THAT THE
                    UPLEVEL ADDRESSING CONVENTIONS ARE OBEYED;
                    PROCEDURE CHECKER(ELBATWORD); VALUE ELBATWORD; REAL ELBATWORD;
PRT(562) = CHECKER
                    BEGIN
                         IF MODE ≥ 2 THEN
                         IF GT1 ← ELBATWORD.LVL ≥ FRSTLEVEL THEN
                         IF GT1 < SUBLEVEL THEN
                    IF ELBATWORD.[9:2] ≠ 1
                         THEN BEGIN FLAG(101); ERRORTOG ← TRUE END
                    END CHECKER;
```

```
05110000 T  0003:036011
05111000 T  0003:036011
05112000 T  0003:036011
05113000 T  0003:036011
05114000 T  0003:036011
05115000 T  0003:036011
05116000 T  0003:036113
05117000 T  0003:036410
05118000 T  0003:036512
05119000 T  0003:036611
05120000 T  0003:036912
```

```
COMMENT GIT IS USED TO OBTAIN THE INDEX TO ADDITIONAL INFORMATION
                    GIVEN THE LINK TO THE ELBAT WORD;
               INTEGER PROCEDURE GIT(L); VALUE L; REAL L;
                    GIT ← TAKE(L).INCR+L.LINK;
```

```
05121000 T  0003:036912
05122000 T  0003:036912
05123000 T  0003:036912
05124000 T  0003:036912
```

```
COMMENT GNAT IS USED TO OBTAIN THE PRT ADDRESS OF A GIVEN DESCRIPTOR.
                    IF THE ADDRESS HAS NOT BEEN ASSIGNED, THEN IT USES
                    GETSPACE TO OBTAIN THE PRT ADDRESS;
               INTEGER PROCEDURE GNAT(L); VALUE L; REAL L;
                    BEGIN
                    REAL A;
STACK(F+3) = A
                    IF GNAT ←(A←TAKE(L)).ADDRESS=0
                    THEN PUT(A&(GNAT:=GETSPACE(TRUE,L,LINK+1))[16:37:11],L)
                    END GNAT;
```

```
05125000 T  0003:037512
05126000 T  0003:037512
05127000 T  0003:037512
05128000 T  0003:037512
05129000 T  0003:037512
05130000 T  0003:037512
START OF SEGMENT **********   39

05131000 T  0039:000010
05132000 T  0039:000210
05133000 T  0039:000712
39 IS   10 LONG, NEXT SEG    3
```

```
               REAL PROCEDURE TAKEFRST;
PRT(563) = TAKEFRST
```

```
05188000 T  0003:037512
```

```
                    TAKEFRST ← TAKE(ELBAT[I].LINK+ELBAT[I].INCR);                                  05189000 T  0003:0375:2


          COMMENT STUFFF DIALS THE F-REGISTER INTO THE F-REGISTER FIELD OF A                       05196000 T  0003:0381:2
                  DESCRIPTOR.  THE DESCRIPTOR REMAINS ON THE TOP OF THE                            05197000 T  0003:0381:2
                  STACK;                                                                           05198000 T  0003:0381:2
          PROCEDURE STUFFF(ADDRESS); VALUE ADDRESS; INTEGER ADDRESS;                               05199000 T  0003:0381:2
PRT(564) = STUFFF
                  BEGIN                                                                            05200000 T  0003:0381:2
                       EMITPAIR(ADDRESS,LOD);                                                      05201000 T  0003:0381:2
                            EMITN(512);                                                            05202000 T  0003:0383:2
                            EMITD(33,18,15) END STUFFF;                                            05203000 T  0003:0383:3


          COMMENT    LOCAL IS USED TO SEE WHETHER OR NOT A LABEL IS LOCAL TO OUR                   05204000 T  0003:0385:2
                     PRESENT CODE;                                                                 05205000 T  0003:0385:2
          BOOLEAN PROCEDURE    LOCAL(ELBATWORD);                                                   05206000 T  0003:0385:2
PRT(565) = LOCAL
                  VALUE ELBATWORD; REAL ELBATWORD;                                                 05207000 T  0003:0385:2
                  BEGIN IF ELBATWORD.LVL = LEVEL AND                                               05208000 T  0003:0385:2
                        NOT BOOLEAN(ELBATWORD,FORMAL) THEN                                         05209000 T  0003:0387:2
                            LOCAL ← TRUE END    LOCAL;                                             05210000 T  0003:0388:1


          COMMENT PASSFORMAT COMPILES CODE THAT PASSES A FORMAT.  TWO ITEMS ARE                    05211000 T  0003:0392:0
                  PASSED - THE ARRAY REFERENCING FORMAT TABLE AND THE                              05212000 T  0003:0392:0
                  STARTING INDEX.  THE ROUTINE HANDLES SUPERFORMATS ALSO;                          05213000 T  0003:0392:0
          PROCEDURE PASSFORMAT;                                                                    05214000 T  0003:0392:0
PRT(566) = PASSFORMAT
                  BEGIN INTEGER ADRES;                                                             05215000 T  0003:0392:0
                                                                                  START OF SEGMENT **********   40
STACK(F+2) = ADRES
                       CHECKER(ELBAT[I]);                                                          05216000 T  0040:0000:0
                       ADRES ← ELBAT[I].ADDRESS;                                                   05217000 T  0040:0001:2
                       IF BOOLEAN(ELBAT[I].FORMAL)                                                 05218000 T  0040:0002:1
                          THEN BEGIN EMITV(ADRES); ADRES ← ADRES-1 END                             05219000 T  0040:0002:1
                          ELSE BEGIN                                                               05220000 T  0040:0006:0
                            IF TABLE(I) = SUPERFRMTID                                              05221000 T  0040:0006:1
                              THEN EMITL(TAKEFRST) ELSE EMITL(ELBAT[I].INCR)                       05222000 T  0040:0007:2
                          END;                                                                     05223000 T  0040:0010:0
                       IF TABLE(I) = SUPERFRMTID                                                   05224000 T  0040:0011:2
                          THEN BEGIN  BANA; I ← I-1;                                               05225000 T  0040:0012:0
                            EMITO(SSP); EMITO(ADD); EMITV(ADRES) END;                              05226000 T  0040:0014:1
                       EMITPAIR(ADRES,LOD) END PASSFORMAT;                                         05227000 T  0040:0017:2
                                                                                        40 IS   21 LONG, NEXT SEG    3


          COMMENT  STREAMWORDS EITHER RESERVES OR UNRESERVES STREAM RESERVED                        05228000 T  0003:0392:0
                   WORDS - IT COMPLEMENTS THEIR STATE;                                             05229000 T  0003:0392:0
```

```
                    PROCEDURE STREAMWORDS;                                                                  05230000 T  0003:0392:0
PRT(567) = STREAMWORDS
                              BEGIN GT1 ← 0;                                                                05231000 T  0003:0392:0
                               DO BEGIN                                                                     05232000 T  0003:0393:3
                                INFO[1,GT1].LINK←STACKHEAD[GT2←(T←INFO[1,GT1]).ADDRESS];                    05233000 T  0003:0394:0
                                STACKHEAD[GT2] ← T.LINK;                                                    05234000 T  0003:0398:1
                                GT1 ← GT1+2;                                                                05235000 T  0003:0401:3
                               END UNTIL BOOLEAN(T,FORMAL) END STREAMWORDS;                                 05236000 T  0003:0403:2



                                                                                                            05237000 T  0003:0404:1
                                                                                                            05238000 T  0003:0404:1
                                                                                                            05239000 T  0003:0404:1
                                                                                                            05240000 T  0003:0404:1
                                                                                                            05241000 T  0003:0404:1
                                                                                                            05242000 T  0003:0404:1
                                                                                                            05243000 T  0003:0404:1
                                                                                                            05244000 T  0003:0404:1
                    PROCEDURE PROGDESCBLDR(PRTADR,SAV,SIZE,TYPE);                                           05245000 T  0003:0404:1
                                        VALUE PRTADR,SAV,SIZE,TYPE;                                         05246000 T  0003:0404:1
                                        INTEGER PRTADR,SIZE,TYPE; BOOLEAN SAV;                              05247000 T  0003:0404:1
                              BEGIN PRTADR←PRTADR.[38:10];                                                  05247500 T  0003:0404:1
                              IF SAV THEN BEGIN PRT[PRTADR] ← ( IF TYPE = LDES                              05248000 T  0003:0406:0
                                   THEN SIZE ELSE CORADR)                                                   05248500 T  0003:0408:0
                                     &SIZE[8:38:10]&TYPE[1:43:5]&3[6:46:2];                                 05249000 T  0003:0410:0
                                    IF TYPE≠LDES THEN CORADR←CORADR+SIZE;                                   05250000 T  0003:0413:3
                                   END
                               ELSE BEGIN PRT[PRTADR]←0&DISKADR[18:33:15]&SIZE[8:38:10]                      05251000 T  0003:0416:0
                                                &TYPE[1:43:5]&1[6:46:2];                                    05252000 T  0003:0416:0
                                    DISKADR←(SIZE+29) DIV 30+DISKADR;                                       05253000 T  0003:0419:3
                              END;                                                                          05254000 T  0003:0422:0
                              END PROGDESCBLDR;                                                             05254500 T  0003:0424:0
                              END PROGDESCBLDR;                                                             05255000 T  0003:0424:0



                    COMMENT  DOTSYNTAX ANALYSES THE SYNTAX OF A PARTIAL WORD DESIGNATOR.                    05267000 T  0003:0424:1
                             IT REPORTS IF AN ERROR IS FOUND.  IT RETURNS WITH THE                          05268000 T  0003:0424:1
                             LITERALS INVOLVED;                                                             05269000 T  0003:0424:1
                    BOOLEAN PROCEDURE DOTSYNTAX(FIRST,SECOND);                                              05270000 T  0003:0424:1
PRT(570) = DOTSYNTAX
                              INTEGER FIRST,SECOND;                                                         05271000 T  0003:0424:1
                              BEGIN                                                                         05272000 T  0003:0424:1
                                LABEL EXIT;                                                                 05273000 T  0003:0424:1
                                                                                          START OF SEGMENT **********   41
                                 IF STEPI = FIELDID THEN % GET INFO FROM INFO      %112-                  05273100 C  0041:0000:0
                                   BEGIN                                           %112-                  05273200 C  0041:0001:2
                                     FIRST := ELBAT[I].SBITF;                      %112-                  05273300 C  0041:0001:3
                                     SECOND := ELBAT[I].NBITF;                     %112-                  05273400 C  0041:0003:2
                                     GO TO EXIT;                                   %112-                  05273500 C  0041:0005:2
                                   END                                             %112-                  05273600 C  0041:0005:3
                                 ELSE                                              %112-                  05273700 C  0041:0005:3
                                   IF ELCLASS = LFTBRKET THEN                      %112-                  05273800 C  0041:0005:3
                                     IF STEPI = FIELDID THEN                       %112-                  05273900 C  0041:0006:1
                                       BEGIN                                       %112-                  05274000 P  0041:0008:0
                                         FIRST := ELBAT[I].SBITF;                  %112-                  05274100 C  0041:0008:1
```

```
                    SECOND := ELBAT[I].NBITF;                          %112-
                    IF STEPI = RTBRKET THEN                            %112-
                      GO TO EXIT;                                      %112-
                  END                                                  %112-
                ELSE                                                   %112-
                  IF ELCLASS = LITNO THEN                              %112-
              IF STEPI = COLON THEN
              IF STEPI = LITNO THEN
              IF STEPI = RTBRKET THEN
COMMENT IF TESTS ARE PASSED THEN SYNTAX IS CORRECT;
              IF (FIRST + ELBAT[I-3].ADDRESS) ×
                (SECOND + ELBAT[I-1].ADDRESS)≠0 THEN
              IF FIRST + SECOND ≤ 48 THEN
COMMENT IF TESTS ARE PASSED THEN RANGES OF LITERALS ARE O.K.;
                GO TO EXIT;
              ERR(114); COMMENT ERROR IF SYNTAX OR RANGE FAILS;
              DOTSYNTAX + TRUE; EXIT; END DOTSYNTAX;
```

```
05274200 C  0041:0010:1
05274300 C  0041:0012:0
05274400 C  0041:0013:2
05274500 C  0041:0013:3
05274600 C  0041:0013:3
05275000 P  0041:0013:3
05276000 T  0041:0015:2
05277000 T  0041:0016:1
05278000 T  0041:0018:0
05279000 T  0041:0019:3
05280000 T  0041:0019:3
05281000 T  0041:0022:0
05282000 T  0041:0025:2
05283000 T  0041:0027:2
05284000 T  0041:0027:2
05285000 T  0041:0027:3
05286000 T  0041:0028:0
41 IS   32 LONG, NEXT SEG    3
```

```
05287000 T  0003:0424:1
05288000 T  0003:0424:1
05289000 T  0003:0424:1
05290000 T  0003:0424:1
05291000 T  0003:0424:1
05292000 T  0003:0424:1
05293000 T  0003:0424:1
05294000 T  0003:0424:1
05295000 T  0003:0424:1
05296000 T  0003:0424:1
05297000 T  0003:0424:1
```

```
                BOOLEAN PROCEDURE RANGE(LOWER,UPPER);
PRT(571) = RANGE
                          VALUE LOWER,UPPER;
                           REAL LOWER,UPPER;
                 COMMENT RANGE TESTS THE CLASS OF THE ITEM IN ELBAT[I] TO SEE IF
                IT IS GREATER THAN OR EQUAL TO LOWER OR LESS THAN OR EQUAL TO
                UPPER AND SETS RANGE TO TRUE OR FALSE ACCORDINGLY.  THE ITEMS
                CLASS MUST BE IN ELCLASS;
                RANGE←ELCLASS ≥ LOWER AND ELCLASS ≤ UPPER;
```

```
05298000 T  0003:0424:1
05299000 T  0003:0424:1
05300000 T  0003:0424:1
05301000 T  0003:0424:1
05302000 T  0003:0424:1
05303000 T  0003:0424:1
05304000 T  0003:0424:1
```

```
              COMMENT GET OBTAINS A SYLLABLE FROM EDOC, THE ARRAY INTO WHICH CODE IS
                      EMITTED;
              INTEGER PROCEDURE GET(L); VALUE L; REAL L;
                      BEGIN
                        INTEGER STREAM PROCEDURE GETSYL(W,S); VALUE S;
PRT(572) = GETSYL
                          BEGIN DI ← LOC GETSYL; DI ← DI+6;
                                SI ← W; SI ← SI+S; SI ← SI+S; DS ← 2 CHR END;
```

```
05305000 T  0003:0429:2
05306000 T  0003:0429:2
05307000 T  0003:0429:2
05308000 T  0003:0429:2
05309000 T  0003:0429:2
START OF SEGMENT **********   42
05310000 T  0042:0000:0
05311000 T  0042:0000:1
```

```
                    GET←GETSYL(CODE(     L DIV 4+1),L,[46:2]); END  GET;                                  05312000 T  0042:0003:2
                                                                                                  42 IS   12 LONG, NEXT SEG    3

                COMMENT  CALL SWITCH PERFORMS THE FINAL MESS OF GETTING A PROPER DE-                        05313000 T  0003:0429:2
                         SCRIPTOR TO THE TOP OF THE STACK;                                                  05314000 T  0003:0429:2
                PROCEDURE CALLSWITCH(H); VALUE H; REAL H;                                                   05315000 T  0003:0429:2
PRT(573) = CALLSWITCH
                    BEGIN EMITV(GNAT(H)); EMITO(PRTE); EMITO(LOD) END CALLSWITCH;                           05316000 T  0003:0429:2

                                                                                                            05317000 T  0003:0433:2
                                                                                                            05318000 T  0003:0433:2
                                                                                                            05319000 T  0003:0433:2
                                                                                                            05320000 T  0003:0433:2
                                                                                                            05321000 T  0003:0433:2
                                                                                                            05322000 T  0003:0433:2
                                                                                                            05323000 T  0003:0433:2
                                                                                                            05324000 T  0003:0433:2
                                                                                                            05325000 T  0003:0433:2
            PROCEDURE WRITEPRT(PORS,N,GS); VALUE PORS,N,GS; INTEGER PORS,N,GS;                              05325010 T  0003:0433:2
PRT(574) = WRITEPRT
                BEGIN                                                                                       05325020 T  0003:0433:2
                LABEL EXIT;                                                                                 05325030 T  0003:0433:2

                                                                                                  START OF SEGMENT **********   43
            STREAM PROCEDURE FILLIT(LIN,PORS,CELL,N,ID);                                                    05325040 T  0043:0000:0
PRT(575) = FILLIT
                                    VALUE PORS,CELL,N;                                                      05325050 T  0043:0000:0
                BEGIN                                                                                       05325060 T  0043:0000:0
                LOCAL COUNT;                                                                                05325070 T  0043:0000:0
                LABEL M0,M1,M2,M3,M4,M5,M6,M7,XIT;                                                          05325080 T  0043:0000:0
                SI:=LOC PORS; SI:=SI+3; DI:=LIN; % "PRT" OR "STACK",                                        05325090 T  0043:0000:0
                IF SC="P" THEN                                                                              05325100 T  0043:0000:1
                     BEGIN  DS:=3 CHR; DS:=LIT"(";  END                                                     05325110 T  0043:0001:2
                ELSE BEGIN                                                                                  05325120 T  0043:0002:1
                     DS:=5 CHR; DS:=LIT"("; SI:=LOC CELL; SI:=SI+5;                                         05325130 T  0043:0002:1
                     IF SC≥"6" THEN DS:=2 LIT"F-" ELSE DS:=2 LIT"F+";                                       05325140 T  0043:0004:0
                     COUNT:=DI; DI:=LOC CELL; DI:=DI+4;                                                     05325150 T  0043:0005:3
                     DS:=11 RESET; DI:=COUNT;                                                               05325160 T  0043:0006:1
                     END;                                                                                   05325170 T  0043:0007:2
                SI:=LOC CELL; SI:=SI+4; TALLY:=4; % LOCATION,                                               05325180 T  0043:0007:2
                3(IF SC="0" THEN % DONT PRINT LEADING ZEROES,                                               05325190 T  0043:0007:3
                     BEGIN SI:=SI+1; TALLY:=TALLY+63  END ELSE JUMP OUT);                                   05325200 T  0043:0008:1
                COUNT:=TALLY; DS:=COUNT CHR; TALLY:=0; COUNT:=TALLY;                                        05325210 T  0043:0011:2
                DS:=4 LIT") = "; CELL:=DI; % SAVE OUR PLACE,                                                05325220 T  0043:0012:0
                CI:=CI+N;                                                                                   05325230 T  0043:0013:2
                GO M0;                                                                                      05325240 T  0043:0013:3
                GO M1;                                                                                      05325250 T  0043:0014:0
                GO M2;                                                                                      05325260 T  0043:0014:0
                GO M3;                                                                                      05325270 T  0043:0014:1
                GO M4;                                                                                      05325280 T  0043:0014:1
                GO M5;                                                                                      05325290 T  0043:0015:2
                GO M6;                                                                                      05325300 T  0043:0015:2
```

```
                    GO M7;                                                                  05325310 T  00431001513
               M0:  SI:=ID; SI:=SI+2; DI:=LOC COUNT;                                        05325320 T  00431001513
                    DI:=DI+7; DS:=CHR; DI:=CELL;  DS:=COUNT CHR;                            05325330 T  00431001611
                    GO XIT;                                                                 05325340 T  00431001810
               M1:  DI:=CELL; DS:=19 LIT"*TEMPORARY STORAGE*"; GO XIT;                       05325350 T  00431001810
               M2:  DI:=CELL;                                                               05325360 T  00431002210
                      DS:=36 LIT"*LIST, LABEL, OR SEGMENT DESCRIPTOR*"; GO XIT;             05325370 T  00431002312
               M3:  DI:=CELL; DS:=27 LIT"*CASE STATEMENT DESCRIPTOR*"; GO XIT;              05325380 T  00431002810
               M4:  DI:=CELL; DS:=19 LIT"*FORMAT DESCRIPTOR*"; GO XIT;                      05325390 T  00431003312
PRT(576) = *LIST, LABEL, OR SEGMENT DESCRIPTOR*
               M5:  DI:=CELL; DS:=24 LIT"*OUTER BLOCK DESCRIPTOR*"; GO XIT;                 05325400 T  00431003712
PRT(577) = *LIST, LABEL, OR SEGMENT DESCRIPTOR*
               M6:  DI:=CELL; DS:=20 LIT"*SEGMENT DESCRIPTOR*"; GO XIT;                     05325410 T  00431004113
PRT(600) = *LIST, LABEL, OR SEGMENT DESCRIPTOR*
               M7:  DI:=CELL; DS:=18 LIT"*LABEL DESCRIPTOR*";                               05325420 T  00431004512
PRT(601) = *LIST, LABEL, OR SEGMENT DESCRIPTOR*
               XIT:                                                                         05325430 T  00431004811
                    END FILLIT;                                                             05325440 T  00431004811
PRT(602) = *LIST, LABEL, OR SEGMENT DESCRIPTOR*
```

```
          BLANKET(14,LIN);                                                                  05325450 T  00431004912
          IF N=1 THEN FILLIT(LIN,PORS,GS,0,ACCUM[1])                                        05325460 T  00431005112
     ELSE IF N>1 THEN FILLIT(LIN,PORS,GS,0,INFO[N,LINKR,N,LINKC])                           05325470 T  00431005513
     ELSE FILLIT(LIN,PORS,GS,ABS(N),N);                                                     05325480 T  00431006312
          IF NOHEADING THEN DATIME; WRITELINE;                                              05325490 T  00431006713
          END WRITEPRT;                                                                     05325500 T  00431007912
                                                                                   43 IS   80 LONG, NEXT SEG    3
```

```
     COMMENT  GETSPACE MAKES ASSIGNMENTS TO VARIABLES AND DESCRIPTORS IN                    05326000 T  00031043312
                   THE STACK AND PRT.  PERMANENT TELLS WHETHER IT IS A                      05327000 T  00031043312
                   PERMANENTLY ASSIGNED CELL (ALWAYS IN PRT) OR NOT.  NON                   05328000 T  00031043312
                   PERMANENT CELLS ARE EITHER IN STACK OR PRT ACCORDING TO                  05329000 T  00031043312
                   MODE.  CARE IS TAKEN TO REUSE NON PERMANENT PRT CELLS;                   05330000 T  00031043312
     INTEGER PROCEDURE GETSPACE(PERMANENT,L); VALUE PERMANENT,L;                            05331000 T  00031043312
                                              BOOLEAN PERMANENT; INTEGER L;                 05333000 T  00031043312
               BEGIN LABEL L1,L2,EXIT;                                                      05334000 T  00031043312
                                                                                  START OF SEGMENT **********   44
               STREAM PROCEDURE DOIT(C,A,I,S); VALUE C,A;                                   05334100 T  00441000010
PRT(603) = DOIT
               BEGIN LOCAL N;                                                               05334200 T  00441000010
                    DI←S; DS←8 LIT" "; SI←S; DS←9 WDS;                                      05334300 T  00441000010
                    SI←I; SI←SI+2; DI←LOC N; DI←DI+7; DS←CHR;                               05334400 T  00441000210
                    DI←S; SI←LOC C; 2(DS←4 DEC);                                            05334500 T  00441000312
                    SI←I; SI←SI+3; DS←N CHR;                                                05334600 T  00441000411
               END;                                                                         05334700 T  00441000513
```

```
                 BOOLEAN M,Q;                                                               05343000 T  00441000513
STACK(F+3) = M
```

```
STACK(F+4) = Q
                        INTEGER ROW,COL,GS;                                                    05344000 T  0044!0005!3
STACK(F+5) = ROW
STACK(F+6) = COL
STACK(F+7) = GS
             IF NOT(STREAMTOG AND (LEVEL>2))THEN                                               05344400 T  0044!0005!3
                  IF STEPI=RELOP THEN                                                          05344500 T  0044!0007!3
                       BEGIN                                                                   05344510 T  0044!0009!2
                            IF STEPI>IDMAX                                                     05344520 T  0044!0009!3
                              THEN                                                             05344530 T  0044!0010!0
                                   BEGIN                                                       05344540 T  0044!0010!1
                                        IF ELCLASS=ADOP                                        05344550 T  0044!0011!2
                                             THEN                                              05344560 T  0044!0011!2
                                                  IF ELBAT[I],ADDRESS=SUBOP                    05344570 T  0044!0011!3
                                                       THEN GS←FZERO ELSE GS←512               0534458- T  0044!0013!2
                                   ELSE                                                        05344590 T  0044!0015!3
                            BEGIN GS←0;I←I-1 END;                                              05344600 T  0044!0016!0
                                   IF STEPI≠LITNO THEN FLAG(51);                               05344610 T  0044!0018!1
                                   IF ELBAT[I].ADDRESS≥512 THEN GS←1024;                       05344615 T  0044!0021!2
                                   GS←GS+ELBAT[I],ADDRESS                                      05344620 T  0044!0023!3
                            END                                                                05344630 T  0044!0024!0
                              ELSE                                                             05344640 T  0044!0025!3
                                   BEGIN                                                       05344650 T  0044!0025!3
                                        GS←ELBAT[I],ADDRESS;                                   05344660 T  0044!0028!0
                                        IF GS=0 THEN FLAG(51);                                 05344661 T  0044!0029!3
                                        IF GS≥FZERO AND GS≤1023 THEN GS←-GS;                   05344662 T  0044!0031!3
                                        IF STEPI≠ADOP THEN I←I-1 ELSE                          05344670 T  0044!0034!1
                                             BEGIN                                             05344680 T  0044!0037!3
                                             STEPI;                                            05344690 T  0044!0038!0
                                        GS←ELBAT[I],ADDRESS+                                   05344700 T  0044!0038!1
                                             (IF ELBAT[I-1],ADDRESS=SUBOP                      05344710 T  0044!0039!3
                                                  THEN -GS ELSE +GS);                          05344720 T  0044!0041!2
                                        END;                                                   05344730 T  0044!0044!0
                                   GS←ABS(GS);                                                 05344740 T  0044!0044!0
                             END; Q←GS<512 OR GS>1023;                                         05344750 T  0044!0045!2
                        GO TO EXIT                                                             05344760 T  0044!0047!2
                        END ELSE I←I-1;                                                        05344770 T  0044!0047!3
                   IF MODE = 0 OR PERMANENT                                                    05345000 T  0044!0049!3
                       THEN BEGIN                                                              05346000 T  0044!0050!0
                        IF PRTIMAX > 1023 THEN FLAG(148);                                      05347000 T  0044!0051!2
                        IF ASTOG THEN FLAG(505);                                               05348000 T  0044!0053!2
                        PRTI ←                                                                 05349000 T  0044!0054!1
                        PRTIMAX←(GS←PRTIMAX)+1;                                                05350000 T  0044!0054!1
                        IF STUFFTOG THEN IF (M←(LEVEL=1 AND KLASSF>19)) OR                     05350100 T  0044!0057!2
                           (LEVEL≥3 AND ELBAT[I].CLASS=LABELID) THEN BEGIN                     05350120 T  0044!0060!1
                         IF NOT M THEN                                                         05350140 T  0044!0063!3
                        DOIT(LABELID,GS,INFO[(ELBAT[I]).LINKR,                                 05350160 T  0044!0064!0
                        (ELBAT[I],LINKC+1)],TWXA[0])  ELSE                                     05350180 T  0044!0067!2
                        DOIT(KLASSF,GS,INFO[(LASTINFO+1).LINKR,(LASTINFO+1).LINKC]             05350200 T  0044!0069!3
                             ,TWXA[0]);  WRITE(STUFF,10,TWXA[*]) END; END                      05350300 T  0044!0074!0
                     ELSE BEGIN                                                                05369000 T  0044!0079!3
                        IF STACKCTR > 767 THEN FLAG(149);                                      05370000 T  0044!0080!0
                        STACKCTR ← (GS ← STACKCTR)+1; Q ← FALSE;                               05371000 T  0044!0082!0
                        GO TO EXIT END;                                                        05372000 T  0044!0084!1
L2;          IF GS ≥ 512 THEN GS ← GS+1024;                                                    05373000 T  0044!0085!2
             Q ← TRUE;                                                                         05374000 T  0044!0087!3
EXIT;        GETSPACE ← GS;                                                                    05375000 T  0044!0088!0
             IF GS≥NESTCTR AND GS<FZERO THEN NESTCTR←GS+1;                                     05375100 T  0044!0089!3
```

```
                IF GS > 1023 THEN GS ← GS-1024;                                          05376000 T  0044:0093:2
          IF PRTOG THEN WRITEPRT(IF Q THEN "PRT  " ELSE "STACK",L,B2D(GS));              05376100 T  0044:0095:3
              END GETSPACE;                                                              05378000 T  0044:0100:0
                                                                                  44 IS  107 LONG, NEXT SEG    3


          REAL PROCEDURE DEPTH(I); VALUE I; REAL I;                                      05400000 T  0003:0433:2
PRT(604) = DEPTH
                BEGIN REAL   J,K,T,S,M;                                                  05401000 T  0003:0433:2
                                                                                 START OF SEGMENT **********   45
STACK(F+3) = J
STACK(F+4) = K
STACK(F+5) = T
STACK(F+6) = S
STACK(F+7) = M

                IF T←NESTPRT[I]<0 THEN                                                   05402000 T  0045:0000:0
                  BEGIN DEPTH←CALL[T,[22:13]-1],[35:13];                                 05402100 T  0045:0002:0
                        IF NESTPRT[I],[2:1]=0 THEN NESTCUR←NESTCUR+1;                    05402200 T  0045:0007:2
                        NESTPRT[I],[2:1]←1;                                              05402300 T  0045:0011:2
                  END                                                                    05402400 T  0045:0014:0
                ELSE IF T.[9:13]≠0 THEN DEPTH←T.[9:13]                                   05403000 T  0045:0014:0
                ELSE BEGIN M←0; NESTPRT[I]←-T;                                           05404000 T  0045:0016:0
                     J←T.[22:13]; K←CALL[J-1],[22:13];                                   05405000 T  0045:0020:1
                     FOR J←J STEP 1 UNTIL K DO                                           05406000 T  0045:0026:0
                     IF S←DEPTH(CALL[J])>M THEN M←S;                                     05407000 T  0045:0027:2
                      M←DEPTH←M←CALL[T,[22:13]-1],[35:13];                               05409000 T  0045:0034:1
                     IF NESTCUR≠0 THEN                                                   05409100 T  0045:0040:1
                     IF NESTPRT[I],[2:1]=0 THEN ELSE                                     05409200 T  0045:0041:2
                     BEGIN T←T&M[9:35:13]; NESTCUR←NESTCUR-1 END                         05409300 T  0045:0044:0
                     ELSE T←T&M[9:35:13];                                                05409400 T  0045:0047:3
                     NESTPRT[I]←T;                                                       05409500 T  0045:0050:0
                     END;                                                                05410000 T  0045:0051:3
                END;                                                                     05411000 T  0045:0051:3
                                                                                  45 IS   56 LONG, NEXT SEG    3


          PROCEDURE NESTSORT(L,U); VALUE L,U; REAL L,U; FORWARD;                         05411100 T  0003:0433:2
PRT(605) = NESTSORT
          PROCEDURE SORTNEST;                                                            05412000 T  0003:0433:2
PRT(606) = SORTNEST
                BEGIN ARRAY A[0:14];                                                     05413000 T  0003:0433:2
                                                                                 START OF SEGMENT **********   46
STACK(F+2) = A
                      REAL I,J,K,T;                                                      05414000 T  0046:0001:3

STACK(F+3) = I
STACK(F+4) = J
STACK(F+5) = K
STACK(F+6) = T
                      REAL P,Q;                                                          05414100 T  0046:0001:3
STACK(F+7) = P
STACK(F+10) = Q
            STREAM PROCEDURE NESTFORM(I,N,L,A); VALUE I,N;                               05415000 T  0046:0001:3
PRT(607) = NESTFORM
```

```
BEGIN LOCAL S;
      DI←A; 15(DS←8 LIT " ");
      DI←LOC S; DI←DI+7; SI←L; SI←SI+10; DS←CHR;
      DI←A; DI←DI+1; A←DI;
      DI←DI+6; DS← S CHR;
      DI←A; SI←LOC N; DS←4 DEC;
      DI←A; DS←3 FILL;
END;


FOR I←PRTBASE STEP 1 UNTIL PRTOP DO
IF    NESTPRT[I]≠0 THEN
      BEGIN SORTPRT[Q]←I; Q←Q+1 END;
NESTSORT(0,Q←Q-1);
FOR P←0 STEP 1 UNTIL Q DO
BEGIN I←SORTPRT[P]; T←NESTPRT[I];
      NESTFORM(0,DEPTH(I),INFO[T,LINKR,T,LINKC],A);
      WRITE(LINE[DBL],15,A[*]);
      J←T,[22:13]; K←CALL[J-1],[22:13];
      FOR J←J STEP 1 UNTIL K DO
      BEGIN I←CALL[J];
            T←NESTPRT[I];
            NESTFORM(32,DEPTH(I),INFO[T,LINKR,T,LINKC],A);
            WRITE(LINE,15,A[*]);
      END;
      WRITE(LINE[DBL]);
      END;
```

```
46 IS   66 LONG, NEXT SEG    3
```

```
                  PROCEDURE NESTSORT(L,U); VALUE L,U; REAL L,U;
                        BEGIN REAL I,J,K,M;
```

```
STACK(F+2) = I
STACK(F+3) = J
STACK(F+4) = K
STACK(F+5) = M
```

```
                        LABEL AGAIN,TOP,BOTTOM,EXIT;
                        IF L≠U THEN
                        BEGIN M← (U+L) DIV 2;
                              NESTSORT(L,M);
                              NESTSORT(M+1,U);
                              I←K←L; J←M+1;
                  AGAIN:      IF I>M THEN GO TO TOP;
                              IF J>U THEN GO TO BOTTOM;
                              GT1←NESTPRT[SORTPRT[I],[33:15]],LINK;
                              GT2←NESTPRT[SORTPRT[J],[33:15]],LINK;
                              IF INFO[GT1,LINKR,(GT1+1),LINKC],[18:30]≤
                                 INFO[GT2,LINKR,(GT2+1),LINKC],[18:30] THEN
                                   GO TO BOTTOM;
                  TOP:        SORTPRT[K],[18:15]←SORTPRT[J];
                              J←J+1;
```

```
                IF K←K+1≤U THEN GO TO AGAIN ELSE GO TO EXIT;                    05454000 T  0047:002?:3
BOTTOM:         SORTPRT[K].[18:15]←SORTPRT[I];                                  05455000 T  0047:003?:0
                I←I+1;                                                          05456000 T  0047:003?:2
                IF K←K+1≤U THEN GO TO AGAIN ELSE GO TO EXIT;                    05457000 T  0047:003?:1
EXIT:           FOR I←L STEP 1 UNTIL U DO                                       05458000 T  0047:003?:2
                SORTPRT[I]←SORTPRT[I].[18:15];                                  05459000 T  0047:003?:2
            END;                                                                05460000 T  0047:004?:3
        END;                                                                    05461000 T  0047:004?:3
                                                                      47 IS   46 LONG, NEXT SEG    3


COMMENT     ROUTINES IN THIS SECTION COMPILE CODE FOR ALL EXPRESSIONS;          06000000 T  0003:043?:2
COMMENT   AEXP IS THE ARITHMETIC EXPRESSION ROUTINE;                            06001000 T  0003:043?:2
PROCEDURE AEXP;                                                                 06002000 T  0003:043?:2
            BEGIN                                                               06003000 T  0003:043?:2
                IF ELCLASS = IFV                                                06004000 T  0003:043?:2
                  THEN BEGIN IF IFEXP ≠ ATYPE THEN ERR(102) END                 06005000 T  0003:043?:2
                  ELSE BEGIN ARITHSEC; SIMPARITH END                            06006000 T  0003:043?:1
            END AEXP;                                                           06007000 T  0003:043?:0


COMMENT ARITHSEC COMPILES FIRST PRIMARY IN AN ARITHMETIC EXPRESSION.            06008000 T  0003:043?:0
            IN PARTICULAR IT HANDLES P, +P, -P, AND -P*Q WHERE P                06009000 T  0003:043?:0
            AND Q ARE PRIMARIES;                                                06010000 T  0003:043?:0
PROCEDURE ARITHSEC;                                                             06011000 T  0003:043?:0
            BEGIN                                                               06012000 T  0003:043?:0
              IF ELCLASS = ADOP                                                 06013000 T  0003:043?:0
                 THEN BEGIN                                                     06014000 T  0003:043?:2
                   STEPIT;                                                      06015000 T  0003:044?:0
                   IF ELBAT[I-1].ADDRESS ≠ SUB THEN PRIMARY                     06016000 T  0003:044?:1
                     ELSE BEGIN                                                 06017000 T  0003:044?:2
                       PRIMARY;                                                 06018000 T  0003:044?:0
                       ENDTOG ← LINKTOG; EMITO(CHS);                            06021000 T  0003:044?:1
                       LINKTOG ← ENDTOG; ENDTOG ← FALSE END END                 06022000 T  0003:044?:0
                 ELSE PRIMARY END ARITHSEC;                                     06023000 T  0003:044?:3


COMMENT SIMPARITH COMPILES SIMPLE ARITHMETIC EXPRESSIONS ON THE                 06024000 T  0003:044?:2
            ASSUMPTION THAT AN ARITHMETIC PRIMARY HAS ALREADY BEEN              06025000 T  0003:044?:2
            COMPILED.  IT ALSO HANDLES THE CASE OF A CONCATENATE                06026000 T  0003:044?:2
            WHERE ACTUALPARAPART CAUSED THE VARIABLE ROUTINE TO                 06027000 T  0003:044?:2
            COMPILE ONLY PART OF A PRIMARY.  MOST OF THE WORK OF                06028000 T  0003:044?:2
            SIMPARITH IS DONE BY ARITHCOMP. AN ARTIFIAL ROUTINE                 06029000 T  0003:044?:2
            WHICH DOES THE HIERARCHY ANALYSIS USING RECURSION.                  06030000 T  0003:044?:2
            ARITHCOMP IS A SUBROUTINE ONLY TO GET THIS RECURSION;               06031000 T  0003:044?:2
PROCEDURE SIMPARITH;                                                            06032000 T  0003:044?:2
            BEGIN                                                               06033000 T  0003:044?:2
              WHILE ELCLASS = AMPERSAND                                         06034000 T  0003:044?:2
                 DO BEGIN STEPIT; PRIMARY; PARSE END;                           06035000 T  0003:044?:2
              WHILE ELCLASS ≥ EQVOP DO ARITHCOMP END;                           06036000 T  0003:045?:0
```

```
                    COMMENT ARITHCOMP IS THE GUTS OF THE ARITHMETIC EXPRESSION ROUTINE          06037000 T  0003:045513
                            ANALYSIS.  IT CALLS PRIMARY AT APPROPRIATE TIMES AND                06038000 T  0003:045513
                            EMITS THE ARITHMETIC OPERATORS.  THE HIERARCHY ANALYSIS             06039000 T  0003:045513
                            IS OBTAINED BY RECURSION;                                           06040000 T  0003:045513
                    PROCEDURE ARITHCOMP;                                                        06041000 T  0003:045513
                            BEGIN INTEGER OPERATOR, OPCLASS;                                    06042000 T  0003:045513
                                                                                      START OF SEGMENT **********   48
STACK(F+2) = OPERATOR
STACK(F+3) = OPCLASS
                             DO BEGIN                                                           06043000 T  0048:000010
                              OPERATOR ← 1 & ELBAT[I] [36:17:10];                               06044000 T  0048:000010
                        COMMENT THIS SETS UP THE OPERATOR WHICH WILL BE EMITTED.  THE HIGH      06045000 T  0048:000210
                              ORDER TEN BITS OF THE OPERATOR ARE LOCATED IN [17:10]             06046000 T  0048:000210
                              OF THE ELBAT WORD;                                                06047000 T  0048:000210
                                OPCLASS ← ELCLASS;                                              06048000 T  0048:000210
                                STEPIT; PRIMARY;                                                06049000 T  0048:000211
                                   BEGIN                                                        06051000 T  0048:000313
                                     WHILE OPCLASS < ELCLASS DO ARITHCOMP;                      06052000 T  0048:000313
                        COMMENT THE CLASSES ARE ARRANGED IN ORDER OF HIERARCHY;                 06053000 T  0048:000610
                                     EMIT(OPERATOR);                                            06054000 T  0048:000610
                                     EMIT(0); L ← L-1;                                          06054100 T  0048:000712
                              STACKCT ← 1;                                                      06054150 T  0048:000912
                                   END;                                                         06054200 T  0048:000913
                               END UNTIL OPCLASS ≠ ELCLASS END ARITHCOMP;                       06055000 T  0048:000913
                                                                                      48 IS    14 LONG, NEXT SEG    3
```

```
                INTEGER PROCEDURE EXPRSS; BEGIN AEXP; EXPRSS ← ATYPE END;                       06057000 T  0003:045513


                PROCEDURE POLISHER(EXPECT); VALUE EXPECT; REAL EXPECT;                          06060000 T  0003:045912
                        BEGIN LABEL EXIT;                                                       06061000 T  0003:045912
                                                                                      START OF SEGMENT **********   49
                              LABEL EL;                                                         06061900 T  0049:000010
                              REAL COUNT,T1, T2;                                                06062000 T  0049:000010
STACK(F+2) = COUNT
STACK(F+3) = T1
STACK(F+4) = T2
                              BOOLEAN S;                                                        06063000 T  0049:000010
STACK(F+5) = S
                              REAL SSS;    INTEGER Z;                                           06063500 T  0049:000010
STACK(F+6) = SSS
STACK(F+7) = Z
                              STREAM PROCEDURE WRITEOUT(C,N,L); VALUE C,N;                      06064000 T  0049:000010
PRT(6,0) = WRITEOUT
                                   BEGIN DI ← L; DS ← 2 LIT "S=";                               06065000 T  0049:000010
                                         SI ← LOC C; SI ← SI+7; DS ← CHR;                       06066000 T  0049:000011
                                         SI ← LOC N; DS ← DEC;                                  06067000 T  0049:000113
                                         58(DS←2LIT " ");                                       06067500 T  0049:000210
                                   END;                                                         06068000 T  0049:000312
```

```
SSS← STACKCTR;                                                    06068500 T  0049:000312
IF STEPI ≠ LEFTPAREN THEN GO TO EXIT;                             06069000 T  0049:000411
DO BEGIN                                                          06070000 T  0049:000610
     IF STEPI ≥ OPERATORS THEN                                    06071000 T  0049:000712
          BEGIN T1 ← (T2 ← ELBAT[I]).ADDRESS;                     06072000 T  0049:000810
               S ← S OR COUNT ≠ T2.[11:3] < 0;                    06074000 T  0049:001011
               COUNT ← T2.[14:2]+COUNT-2;                         06075000 T  0049:001312
               IF ELCLASS ≥ OPERATOR THEN                         06076000 T  0049:001513
               BEGIN IF T1 ≠ 0 THEN EMITO(T1)                     06077000 T  0049:001610
               ELSE BEGIN                                         06078000 T  0049:001810
                    T1 ← T2.LINK+2;                               06079000 T  0049:001912
                    T2 ← T2.INCR+T1;                              06080000 T  0049:002112
                    FOR T1 ← T1 STEP 1 UNTIL T2 DO                06081000 T  0049:002211
                        EMIT(TAKE(T1));                           06082000 T  0049:002410
                    END;                                          06083000 T  0049:002713
                END ELSE BEGIN T2 ← ELCLASS;                      06084000 T  0049:002713
                    IF STEPI ≠ LITNO THEN                         06085000 T  0049:002811
                        BEGIN ERR(500); GO TO EXIT END;           06086000 T  0049:002913
                    IF T2 = BITOP THEN EMIT(T1&C                  06087000 T  0049:003113
                        [36:42:6]) ELSE                           06088000 T  0049:003312
                    IF T2 = HEXOP THEN EMIT(T1&                   06089000 T  0049:003411
                        (T2+C DIV 6)[36:45:3]&(C-T2×6)            06090000 T  0049:003611
                        [39:45:3]) ELSE                           06091000 T  0049:003913
                    IF T2 = ISOLATE THEN                          06092000 T  0049:004112
                        BEGIN T2 ← C;                             06093000 T  0049:004210
                            IF STEPI ≠ LITNO                      06094000 T  0049:004313
                            THEN BEGIN ERR(500);                  06095000 T  0049:004410
                            GO TO EXIT END;                       06096000 T  0049:004513
                            EMIT(Z+((T2+C-1)DIV 6-C DIV           06097000 T  0049:004610
                           6+1)×512+(48-T2-C)MOD 6×64+            06098000 T  0049:004610
                            37);                                  06099000 T  0049:004610
                        END END;                                  06099100 T  0049:004610
               STEPIT;                                            06099200 T  0049:004811
               S ← S OR COUNT < 0;                                06100000 T  0049:005211
          END ELSE BEGIN                                          06101000 T  0049:005410
               IF ELCLASS = LABELID THEN                          06102000 T  0049:005410
                    BEGIN T1:=2;                                  06103000 T  0049:005411
EL:                      GT4 ← TAKE(T2←GIT(ELBAT[I]));            06104000 T  0049:005610
                         PUT(L,T2);                               06104100 T  0049:005611
                         IF GT4 = 0 THEN GT4 ← L;                 06104200 T  0049:005713
                         IF (GT4:=L-GT4)DIV 4 ≥ 128 THEN          06104300 T  0049:005811
                            BEGIN GT4:=0;FLAG(50);END;            06104400 T  0049:006113
                         EMIT(GT4×4+T1);                          06104500 T  0049:006211
                         STEPIT;                                  06104510 T  0049:006411
                    END ELSE                                      06104520 T  0049:006611
               IF ELCLASS ≠ PERIOD THEN AEXP ELSE BEGIN           06104600 T  0049:006811
               T2←0;                                              06104700 T  0049:007011
               IF STEPI=PERIOD THEN                               06104800 T  0049:007112
               BEGIN T2←1; STEPIT END;                            06105000 T  0049:007112
               IF ELCLASS>IDMAX THEN                              06106000 T  0049:007313
                    BEGIN ERR(500); GO TO EXIT END;               06106100 T  0049:007411
               IF ELCLASS = LABELID THEN                          06106200 T  0049:007513
                    BEGIN T1 ← 0; GO TO EL END;                   06106300 T  0049:007712
               IF T1 ← ELBAT[I].ADDRESS = 0 THEN                  06107000 T  0049:007810
                                                                  06107100 T  0049:007913
                                                                  06107200 T  0049:008011
                                                                  06108000 T  0049:008210
```

```
                                    BEGIN ERR(100); GO TO EXIT END;
                                 EMITL(T1);
                                IF T1>1023 THEN
                                IF T2=0THEN FLAG(500)
                                ELSE EMITO(PRTE);
                                 STEPIT;
                                END; COUNT ← COUNT+1;
                                END;
                        END UNTIL ELCLASS ≠ COMMA;
                        IF ELCLASS ≠ RTPAREN THEN
                             BEGIN ERR(104); GO TO EXIT END;
                        STEPIT;
                        IF FALSE THEN
                             BEGIN COUNT ← COUNT-EXPECT;
                                  WRITEOUT(IF COUNT < 0 THEN "-" ELSE
                                             IF COUNT = 0 THEN " " ELSE "+",
                                             ABS(COUNT),LIN[0]);
                                  WRITELINE;
                             END;
           EXIT; STACKCTR ← SSS; END;
```

```
06109000 T  00491008410
06110000 T  00491008610
06110100 T  00491008611
06110200 T  00491008713
06110300 T  00491008913
06111000 T  00491009112
06112000 T  00491009113
06113000 T  00491009312
06114000 T  00491009312
06115000 T  00491009410
06116000 T  00491009512
06117000 T  00491009611
06118000 T  00491009712
06119000 T  00491009713
06120000 T  00491009912
06121000 T  00491010113
06122000 T  00491010313
06123000 T  00491010512
06124000 T  00491011512
06125000 T  00491011512
49 IS  120 LONG, NEXT SEG    3
```

```
PROCEDURE PRIMARY;
        BEGIN LABEL
                        L1, L2, L3, L4, L5, L6, L7, L8, L9, L10,
                        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, L36, L37, L38, L39;
                        SWITCH S ←
                        L1, L2, L3, L4, L5, L6, L7, L8, L9, L10,
                        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, L36, L37, L38, L39;
                        LABEL EXIT,RP,LDOT,LAMPER;
                        GO TO S[ELCLASS];
                        IF ELCLASS = LFTBRKET THEN
                             BEGIN STEPIT; VARIABLE(FL);
                                  IF ELCLASS ≠ RTBRKET THEN
                                       BEGIN ERR(118); GO TO EXIT END;
                                  STEPIT;
                                  GO TO LDOT;
                             END;
                        IF ELCLASS = NOTOP THEN
                             BEGIN STEPIT; PRIMARY;
                                  EMITLNG; EMIT(0); L←L-1;
                                  GO TO EXIT;
                             END;
                        IF ELCLASS = UNKNOWNID THEN ERR(100);
L1;L2;L3;L4;L5;L6;L8;L9;L10;L12;L13;L16;L17;L20;L21;L24;L25;L28;L29;
L32;
                        ERR(103); GO TO EXIT;
                L7;
                        SUBHAND(FALSE); GO TO LDOT;
```

```
06126000 T  00031045912
06127000 T  00031045912
START OF SEGMENT **********   50
06128000 T  00501000010
06129000 T  00501000010
06130000 T  00501000010
06131000 T  00501000010
06132000 T  00501000010
06133000 T  00501000211
06134000 T  00501000211
06135000 T  00501000211
06136000 T  00501000211
06137000 T  00501002211
06138000 T  00501002211
06139000 T  00501002512
06140000 T  00501002513
06141000 T  00501002713
06142000 T  00501002810
06143000 T  00501003010
06144000 T  00501003011
06145000 T  00501003112
06146000 T  00501003112
06147000 T  00501003113
06148000 T  00501003312
06149000 T  00501003513
06150000 T  00501003610
06151000 T  00501003610
06152000 T  00501003810
06153000 T  00501003912
06154000 T  00501003912
06155000 T  00501004010
06156000 T  00501004112
```

```
L11:
     IMPFUN; STACKCT ← STACKCT-1; GO TO LDOT;                                  06157000 T  0050:004:0
L14:L15:
     STRMPROCSTMT; GO TO LDOT;                                                 06158000 T  0050:004:2
L18:L19:
     PROCSTMT(FALSE); GO TO LDOT;                                              06159000 T  0050:004:2
L22:L23:L26:L27:L30:L31:
     VARIABLE(FP); GO TO LAMPER;                                               06160000 T  0050:004:0
L33:L35:
     EMIT(O&ELBAT[I] [36:17:10]); STEPIT; GO TO LAMPER;                        06161000 T  0050:004:2
L34:L36:
     EMITNUM(C); STEPIT; GO TO LAMPER;                                         06162000 T  0050:004:2
L38:
     POLISHER(1); GO TO LDOT;                                                  06163000 T  0050:004:0
L39:
     STEPIT; PRIMARY; STACKCT ← STACKCT-1;                                     06164000 T  0050:004:2
     EMITO(LOD); GO TO LDOT;                                                   06165000 T  0050:005:0
L37:
     STEPIT; AEXP;                                                             06166000 T  0050:005:2
     STACKCT ← STACKCT-1;                                                      06167000 T  0050:005:0
     IF ELCLASS ≠ RTPAREN THEN                                                 06168000 T  0050:005:0
          BEGIN ERR(104); GO TO EXIT END;                                      06169000 T  0050:005:3
     STEPIT;                                                                   06170000 T  0050:005:0
LDOT:DOT;                                                                      06171000 T  0050:005:2
LAMPER:                                                                        06172000 T  0050:005:0
     STACKCT ← STACKCT +1;                                                     06172500 T  0050:006:0
     WHILE ELCLASS = AMPERSAND DO                                              06173000 T  0050:006:3
          BEGIN STEPIT; PRIMARY; PARSE END;                                    06174000 T  0050:006:0
     EXIT; END PRIMARY;                                                        06174500 T  0050:006:2
                                                                               06175000 T  0050:006:0
                                                                               06176000 T  0050:006:2
                                                                               06177000 T  0050:006:1
                                                                               06178000 T  0050:006:2
                                                                               06179000 T  0050:006:1
                                                                               06179500 T  0050:006:2
                                                                               06180000 T  0050:007:0
                                                                               06181000 T  0050:007:2
                                                                               06182000 T  0050:007:0
                                                                            50 IS   76 LONG, NEXT SEG    3

     PROCEDURE IMPFUN;                                                         06183000 T  0003:045:2
          BEGIN REAL T1,T2;                                                    06184000 T  0003:045:2
                                                                            START OF SEGMENT **********   51
STACK(F+2) = T1
STACK(F+3) = T2
               T1 ← (T2 ← ELBAT[I]),ADDRESS;                                   06185000 T  0051:000:0
               PANA;                                                           06186000 T  0051:000:2
               IF T1 ≠ 0 THEN EMITO(T1)                                        06187000 T  0051:000:1
                    ELSE BEGIN                                                 06188000 T  0051:000:0
                    T1 ← T2.LINK+T2.INCR+1;                                    06189000 T  0051:000:5
                    T2 ← T2.LINK+2;                                            06190000 T  0051:000:7
                    FOR T2 ← T2 STEP 1 UNTIL T1 DO EMIT(TAKE(T2));             06191000 T  0051:000:9
                    END;                                                       06192000 T  0051:001:4
          END;                                                                 06193000 T  0051:001:4
                                                                            51 IS   17 LONG, NEXT SEG    3

     PROCEDURE SUBHAND(FROM); VALUE FROM; BOOLEAN FROM;                        06194000 T  0003:045:2
          BEGIN LABEL EXIT;                                                    06195000 T  0003:045:2
                                                                            START OF SEGMENT **********   52
               REAL T1;                                                        06196000 T  0052:000:0
```

```
STACK(F+2) = T1
                    T1 ← TAKEFRST;                                                  06197000 T  00521000:0
                    IF ELCLASS ≠ SUBID AND FROM THEN                                06198000 T  00521000:2
                       BEGIN IF STEPI ≠ ASSIGNOP THEN                               06199000 T  00521000:0
                             BEGIN FLAG(503); GO TO EXIT END;                       06200000 T  00521000:3
                             STEPIT;                                                06201000 T  00521000:3
                             AEXP;                                                  06202000 T  00521000:0
                             EMITO(XCH);                                            06203000 T  00521000:1
                             GO TO EXIT;                                            06204000 T  00521000:2
                       END;                                                         06205000 T  00521000:3
                    EMITL((L+6) DIV 4-(T1.[24:12]-1) DIV 4);                        06206000 T  00521000:3
                    EMITB(BBW,BUMPL,T1.[36:12]);                                    06207000 T  00521001:3
                    STEPIT;                                                         06208000 T  00521001:0
                    ADJUST;                                                         06208500 T  00521001:1
             EXIT:  END SUBHAND;                                                    06209000 T  00521001:2
                                                                                  52 IS   19 LONG, NEXT SEG    3


             COMMENT IFEXP COMPILES CONDITIONAL EXPRESSIONS.  IT REPORTS THE TYPE    06292000 T  00031045:2
                    OF EXPRESSIONS AS EXPRSS REPORTS;                               06293000 T  00031045:2
             INTEGER PROCEDURE IFEXP;                                               06294000 T  00031045:2
                    BEGIN INTEGER TYPE,THENBRANCH,ELSEBRANCH;                       06295000 T  00031045:2
                                                                                 START OF SEGMENT **********   53
STACK(F+3) = TYPE
STACK(F+4) = THENBRANCH
STACK(F+5) = ELSEBRANCH
                       IFCLAUSE;                                                    06296000 T  00531000:0
                    STACKCT ← 0;                                                    06296500 T  00531000:1
                       THENBRANCH ← BUMPL;                                          06297000 T  00531001:2
             COMMENT SAVE L FOR LATER FIXUP;                                        06298000 T  00531003:2
                       IFEXP ← TYPE ← EXPRSS; COMMENT COMPILE 1ST EXPRSS;           06299000 T  00531003:2
                    STACKCT ← 0;                                                    06299500 T  00531004:1
                       ELSEBRANCH ← BUMPL;                                          06300000 T  00531005:2
                       EMITB(BFC,THENBRANCH,L);                                     06301000 T  00531007:2
                       IF ELCLASS ≠ ELSEV THEN ERR(155) ELSE BEGIN                  06302000 T  00531008:0
                       STEPIT;                                                      06303000 T  00531010:1
                    AEXP; STACKCT ← 1;                                              06305000 T  00531011:2
                       COMMENT THIS COMPILES PROPER TYPE SECOND EXPRSS;             06306000 T  00531012:1
                       EMITB(BFW,ELSEBRANCH,L);                                     06307000 T  00531012:1
                       EMIT(1); L ← L-1;                                            06308000 T  00531013:3
             COMMENT THIS IS USED BY EMITLNG TO CLEANUP CODE.  COMPARE WITH         06309000 T  00531015:3
                    BOOSEC, BOOCOMP, AND RELATION;                                  06310000 T  00531015:3
                    END END IFEXP;                                                  06311000 T  00531015:3
                                                                                  53 IS   19 LONG, NEXT SEG    3


             COMMENT PARSE COMPILES CODE FOR THE CONCATENATE;                       06312000 T  00031045:2
             PROCEDURE PARSE;                                                       06313000 T  00031045:2
                    BEGIN INTEGER FIRST,SECOND,THIRD;                               06314000 T  00031045:2
                                                                                 START OF SEGMENT **********   54
STACK(F+2) = FIRST
STACK(F+3) = SECOND
STACK(F+4) = THIRD
```

```
        LABEL EXIT,NEXTCHK;                                          %107-
IF ELCLASS = FIELDID THEN                                            %112-
  BEGIN                                                              %112-
    FIRST := ELBAT[I].SBITF;                                         %112-
    SECOND := 48 - (THIRD := ELBAT[I].NBITF);                        %112-
    GO TO NEXTCHK;                                                   %112-
  END                                                                %112-
ELSE                                                                 %112-
        IF ELCLASS = LFTBRKET THEN
IF STEPI = FIELDID THEN                                              %112-
  BEGIN                                                              %112-
    FIRST := ELBAT[I].SBITF;                                         %112-
    SECOND := 48 - (THIRD := ELBAT[I].NBITF);                        %112-
    IF STEPI # RTBRKET THEN                                          %112-
      BEGIN                                                          %112-
        ERR(94);                                                     %112-
        GO TO EXIT;                                                  %112-
      END;                                                           %112-
    GO TO NEXTCHK;                                                   %112-
  END                                                                %112-
ELSE                                                                 %112-
  IF ELCLASS = LITNO THEN                                            %112-
        IF STEPI = COLON THEN
        IF STEPI = LITNO THEN
      IF STEPI = RTBRKET THEN % WE WILL TAKE CARE OF THE RES
        IF (FIRST := ELBAT[I-3].ADDRESS) <                           %107-
           (THIRD := ELBAT[I-1].ADDRESS) # 0 THEN                    %107-
          BEGIN                                                      %107-
            SECOND := 48 - THIRD; % SOURCE IS RIGHT JUST.            %107-
            GO TO NEXTCHK;                                           %107-
          END                                                        %107-
        ELSE % BAD BITS, FALL THROUGH TO ERROR.                      %107-
      ELSE % MAYBE A COLON.                                          %107-
        IF ELCLASS = COLON THEN %                                    %107-
      IF STEPI = LITNO THEN
      IF STEPI = RTBRKET THEN
COMMENT IF TEST ARE PASSED THEN SYNTAX IS CORRECT;
      IF (FIRST  + ELBAT[I-5].ADDRESS) <
         (SECOND + ELBAT[I-3].ADDRESS) <
         (THIRD  + ELBAT[I-1].ADDRESS) # 0 THEN
NEXTCHK:  IF FIRST + THIRD ≤ 48 THEN % SO FAR SO GOOD.               %107-
      IF SECOND+ THIRD ≤ 48 THEN
COMMENT IF TEST ARE PASSED THEN RANGES OF LITERALS ARE O.K.;
        BEGIN
          STEPIT;
          EMITD(SECOND,FIRST,THIRD);
    STACKCT := 1;
          GO TO EXIT END;
      ERR(113); COMMENT ERROR IF SYNTAX OR RANGE FAILS;
EXIT:   END PARSE;
```

```
06315000 P  00541000:0
06315100 C  00541000:0
06315200 C  00541000:1
06315300 C  00541000:2
06315400 C  00541000:1
06315500 C  00541000:2
06315600 C  00541000:3
06315700 C  00541000:3
06316000 T  00541000:3
06316100 C  00541000:2
06316200 C  00541000:1
06316300 C  00541000:2
06316400 C  00541001:1
06316500 C  00541001:2
06316600 C  00541001:0
06316700 C  00541001:1
06316800 C  00541001:2
06316900 C  00541001:3
06317000 P  00541001:3
06317100 C  00541001:0
06317200 C  00541001:0
06317300 C  00541001:0
06318000 T  00541001:3
06319000 T  00541001:2
06320000 P  00541002:1
06320100 C  00541002:0
06320200 C  00541002:1
06320300 C  00541002:2
06320400 C  00541002:3
06320500 C  00541002:2
06320600 C  00541002:3
06320700 C  00541002:3
06320800 C  00541002:3
06320900 C  00541003:0
06321000 T  00541003:2
06322000 T  00541003:1
06323000 T  00541003:0
06324000 T  00541003:0
06325000 T  00541003:1
06326000 T  00541003:2
06327000 P  00541004:3
06328000 T  00541004:0
06329000 T  00541004:0
06330000 T  00541004:0
06331000 T  00541004:1
06332000 T  00541004:2
06332500 T  00541004:0
06333000 T  00541004:2
06334000 T  00541004:3
06335000 T  00541005:0
```

54 IS 54 LONG, NEXT SEG 3

```
COMMENT DOT COMPILES CODE FOR PARTIAL WORD DESIGNATORS, EXCEPT FOR
        THOSE CASES HANDLED BY THE VARIABLE ROUTINE;
```

```
06336000 T  00031045:2
06337000 T  00031045:2
```

```
                PROCEDURE DOT;                                                          06338000 T  0003!045!2
                        BEGIN INTEGER FIRST,SECOND; LABEL EXIT;                         06339000 T  0003!045!2
                                                                                  START OF SEGMENT **********   55
STACK(F+2) = FIRST
STACK(F+3) = SECOND
                          IF ELCLASS = PERIOD THEN BEGIN                                06340000 T  0055!000!0
                                IF DOTSYNTAX(FIRST,SECOND) THEN GO TO EXIT;             06341000 T  0055!000!2
                                                                                        06342000 T  0055!000!2
                                                                                        06343000 T  0055!000!2
                          EMIT(0,FIRST,SECOND);                                         06344000 T  0055!000!2
                                STEPIT;                                                 06345000 T  0055!000!0
             EXIT; END END DOT;                                                         06346000 T  0055!000!1
                                                                                  55 IS     8 LONG, NEXT SEG    3



                PROCEDURE IFCLAUSE;                                                     06409000 T  0003!045!2
                        BEGIN STEPIT; BEXP;                                             06410000 T  0003!045!2
                          IF ELCLASS # THENV THEN ERR(116)ELSE STEPIT END IFCLAUS;      06411000 T  0003!046!2
COMMENT  PANA COMPILES THE CONSTRUCT: (<ARIT. EXP.>);                                   06412000 T  0003!046!0



                PROCEDURE PANA;                                                         06413000 T  0003!046!0
                        BEGIN                                                           06414000 T  0003!046!0
                         IF STEPI # LEFTPAREN THEN ERR(105)                             06415000 T  0003!046!0
                         ELSE BEGIN STEPIT; AEXP; IF ELCLASS # RTPAREN THEN             06416000 T  0003!046!1
                             ERR(104) ELSE STEPIT END END PANA;                         06417000 T  0003!046!3
```

```
                COMMENT  BANA COMPILES THE CONSTRUCT:  [<ARITH. EXP.>];                 06418000 T  0003!047!0
                PROCEDURE BANA;                                                         06419000 T  0003!047!0
                        BEGIN                                                           06420000 T  0003!047!0
                         IF STEPI # LFTBRKET THEN ERR(117)                              06421000 T  0003!047!0
                             ELSE BEGIN STEPIT; AEXP; IF ELCLASS # RTBRKET THEN         06422000 T  0003!047!3
                               ERR(118) ELSE STEPIT END END BANA      ;                 06423000 T  0003!047!1



                COMMENT THIS SECTION CONTAINS THE STATEMENT ROUTINES;                   07000000 T  0003!047!2
                COMMENT COMPOUNDTAIL COMPILES COMPOUNDTAILS.  IT ALSO ELIMINATES        07001000 T  0003!047!2
                        COMMENTS FOLLOWING ENDS.  AFTER ANY ERROR, ERROR MESSAGES       07002000 T  0003!047!2
                        ARE SUPPRESSED.  COMPOUNDTAIL IS PARTIALLY RESPONSIBLE          07003000 T  0003!047!2
                        FOR RESTORING THE ABILITY TO WRITE ERROR MESSAGES.  SOME        07004000 T  0003!047!2
                        CARE IS ALSO TAKEN TO PREVENT READING BEYOND THE "END.";        07005000 T  0003!047!2
              PROCEDURE COMPOUNDTAIL;                                                   07006000 T  0003!047!2
PRT(611) = COMPOUNDTAIL
                        BEGIN LABEL ANOTHER;                                            07007000 T  0003!047!2
                                                                                  START OF SEGMENT **********   56
                          I = I+1; BEGINCTR = BEGINCTR+1;                               07008000 T  0056!000!0
                 ANOTHER:   ERRORTOG = TRUE;   COMMENT  ALLOW ERROR MESSAGES;         07009000 T  0056!000!1
                          STEPIT;                                                       07010000 T  0056!000!3
```

```
                IF STREAMTOG THEN STREAMSTMT ELSE STMT;                                  07011000 T  0056:0004:0
                IF ELCLASS = SEMICOLON THEN GO TO ANOTHER;                               07012000 T  0056:0006:1
                IF ELCLASS ≠ ENDV                                                        07013000 T  0056:0007:3
                  THEN BEGIN                                                             07014000 T  0056:0008:0
                   ERR(119); GO TO ANOTHER END;                                          07015000 T  0056:0009:2
              ENDTOG←TRUE;                                                               07016000 T  0056:0010:0
              DO STOPDEFINE←TRUE UNTIL                                                   07017000 T  0056:0011:2
                   STEP1≤ENDV AND ELCLASS≥UNTILV                                         07018000 T  0056:0011:3
                   OR NOT ENDTOG;                                                        07019000 T  0056:0013:2
              ENDTOG←FALSE;                                                              07020000 T  0056:0015:2
                   IF BEGINCTR ← BEGINCTR=1 ≠ 0 EQV ELCLASS = PERIOD                     07021000 T  0056:0015:3
                     THEN BEGIN                                                          07022000 T  0056:0017:3
                        IF BEGINCTR = 0 THEN                                             07023000 T  0056:0019:2
                         BEGIN FLAG(143); BEGINCTR ← 1; GO ANOTHER END;                  07024000 T  0056:0019:3
FLAG (120);                                                                      %104-   07025000 P  0056:0022:0
FCR:= (LCR:=MKABS(CBUFF[9]))=9;                                                  %104-   07025010 C  0056:0023:2
   IF LISTER THEN PRINTCARD;                                                     %104-   07025020 C  0056:0026:0
FCR:= (LCR:=MKABS(TBUFF[9]))=9  END;                                             %104-   07025030 C  0056:0042:0
                   IF ELCLASS = PERIOD THEN                                              07026000 T  0056:0045:2
                     BEGIN                                                               07027000 T  0056:0045:3
                        GT5 ← "ND;END,"&"E"[1:43:5];                                     07028000 T  0056:0046:0
                        MOVE(1,GT5,CBUFF[0]);                                            07029000 T  0056:0048:0
                        LASTUSED←4;                                                      07030000 T* 0056:0049:3
                        ELBAT[I←I=2] ←SPECIAL[20];                                       07031000 T  0056:0050:0
                        ELCLASS ← SEMICOLON END                                          07032000 T  0056:0052:1
              END COMPOUNDTAIL;                                                          07033000 T  0056:0053:3
                                                                          56 IS   55 LONG, NEXT SEG    3
```

```
                REAL AXNUM;                                                              07034000 T  0003:0479:2
PRT(612) = AXNUM
                PROCEDURE ACTUALPARAPART(SBIT,INDEX); VALUE SBIT,INDEX;                  07035000 T  0003:0479:2
PRT(613) = ACTUALPARAPART
                     BOOLEAN SBIT; REAL INDEX;                                           07036000 T  0003:0479:2
                     BEGIN LABEL EXIT,COMMON,ANOTHER,POL;                                07037000 T  0003:0479:2
                                                                        START OF SEGMENT **********   57
                          REAL PCTR,SCLASS,ACLASS;                                       07038000 T  0057:0000:0
STACK(F+2) = PCTR
STACK(F+3) = SCLASS
STACK(F+4) = ACLASS
                STREAM PROCEDURE WRITEAX(LINE,ACCUM,N,SEQ); VALUE N;                     07038100 T  0057:0000:0
PRT(614) = WRITEAX
                     BEGIN DI ← LINE; 15(DS ← 8 LIT " ");                                07038200 T  0057:0000:0
                          DI ← LINE; SI ← SEQ; SI ← SI=16; DS ← WDS;                     07038300 T  0057:0002:0
                          DI ← DI+4; DS ← 20 LIT "ACCIDENTAL ENTRY AT ";                 07038400 T  0057:0003:2
                                SI ← ACCUM; SI ← SI+3; DS ← N CHR;                       07038500 T  0057:0006:0
                          SI ← SEQ; DI ← SEQ; DI ← DI=16; DS ← WDS;                      07038600 T  0057:0007:2
                     END;                                                                07038700 T  0057:0008:0



                          BOOLEAN VBIT,IDBIT;                                            07039000 T  0057:0008:0
STACK(F+5) = VBIT
STACK(F+6) = IDBIT
```

```
                PCTR ← 1;                                                                07040000 T  0057:0008:0
ANOTHER:        ACLASS ← STEPI&0[47:47:1];                                               07041000 T  0057:0009:3
                STACKCT ← 0;                                                             07041200 T  0057:0012:0
                GT1 ← TAKE(INDEX+PCTR);                                                  07042000 T  0057:0012:1
                VBIT ← BOOLEAN(GT1,VO);                                                  07043000 T  0057:0014:1
                SCLASS ← GT1.CLASS&0[47:47:1];                                           07044000 T  0057:0015:3
                IF VBIT THEN BEGIN AEXP; GO TO COMMON END;                               07045000 T  0057:0018:0
                IF SBIT THEN SCLASS ← NAMEID;                                            07046000 T  0057:0019:3
                IDBIT ← BOOID < ACLASS AND ACLASS < LABELID;                             07047000 T  0057:0021:2
                IF SCLASS = NAMEID THEN                                                  07048000 T  0057:0023:3
                     BEGIN                                                               07049000 T  0057:0024:0
                          IF IDBIT THEN VARIABLE(FL)                                     07050000 T  0057:0024:1
                          ELSE                                                           07051000 T  0057:0025:3
POL:                           IF ELCLASS = POLISHV THEN POLISHER(1)                     07052000 T  0057:0026:0
                          ELSE       ERR(IF ELCLASS=0 THEN 0 ELSE 123);                07053000 T  0057:0028:1
                          GO TO COMMON;                                                  07054000 T  0057:0032:0
                     END;                                                                07055000 T  0057:0032:1
                IF SCLASS = REALARRAYID THEN                                             07056000 T  0057:0032:1
                     IF ACLASS = REALARRAYID THEN                                        07057000 T  0057:0033:3
                          BEGIN VARIABLE(FL); GO TO COMMON END                           07058000 T  0057:0034:1
                     ELSE GO TO POL;                                                     07059000 T  0057:0036:1
                IF SCLASS ≠ REALID THEN                                                  07060000 T  0057:0036:1
                     BEGIN FLAG(503);                                                    07061000 T  0057:0037:2
                          AEXP;                                                          07062000 T  0057:0038:1
                          ERRORTOG ← TRUE;                                               07063000 T  0057:0039:2
                          GO TO COMMON;                                                  07064000 T  0057:0039:3
                     END;                                                                07065000 T  0057:0040:0
                GT1 ← TABLE(I+1);                                                        07066000 T  0057:0040:0
IF GT1 = COMMA OR GT1 = RTPAREN THEN                                                     07067000 T  0057:0042:0
     BEGIN IF IDBIT THEN                                                                 07068000 T  0057:0043:3
               BEGIN IF ACLASS = REALID AND                                              07069000 T  0057:0044:1
                    BOOLEAN(ELBAT[I],FORMAL)THEN BEGIN                                   07070000 T  0057:0045:3
                    CHECKER (ELBAT[I]);                                                  07070500 T  0057:0047:3
                    EMITPAIR(ELBAT[I],ADDRESS,LOD);                                      07071000 T  0057:0048:1
                    STEPIT; END                                                          07072000 T  0057:0050:0
                    ELSE VARIABLE(FL);                                                   07073000 T  0057:0050:1
                    GO TO COMMON END;                                                    07074000 T  0057:0052:0
          IF ELCLASS ≤ STRNGCON AND ELCLASS > LABELID                                    07075000 T  0057:0052:1
               THEN BEGIN PRIMARY; GO TO COMMON END;                                     07076000 T  0057:0053:3
     END;                                                                                07077000 T  0057:0055:3
EMITO(NOP); EMITO(NOP);                                                                  07078000 T  0057:0055:3
SCLASS ← L;                                                                              07079000 T  0057:0057:2
ADJUST;                                                                                  07080000 T  0057:0058:0
ACLASS ← L.[36:10];                                                                      07081000 T  0057:0058:1
IF IDBIT THEN                                                                            07082000 T  0057:0059:3
     BEGIN VARIABLE(FL);                                                                 07083000 T  0057:0060:0
          IF ELCLASS < AMPERSAND THEN GO TO COMMON;                                      07084000 T  0057:0061:2
                                                                                         07084500 T  0057:0062:1
          SIMPARITH;                                                                     07085000 T  0057:0062:1
     END ELSE AEXP;                                                                      07086000 T  0057:0063:2
IF LISTER THEN                                                                           07086100 T  0057:0064:0
     BEGIN ACCUM[1] ← Q;                                                                 07086200 T  0057:0064:0
          WRITEAX(LINE0],ACCUM[1],Q,[12:6],                                              07086300 T  0057:0066:0
               INFO[LASTSEQROW,LASTSEQUENCE]);                                           07086400 T  0057:0068:0
          WRITELINE;                                                                     07086500 T  0057:0069:3
     END;                                                                                07086600 T  0057:0079:3
AXNUM ← AXNUM+1;                                                                         07086700 T  0057:0079:3
```

```
                    EMITO(RTS);                                                  07087000 T  0057:008:2
                    EMITB(BFW,SCLASS,L);                                         07088000 T  0057:008:3
                    EMITNUM(ACLASS);                                             07089000 T  0057:008:2
                    EMITPAIR(TAKE(PROINFO).ADDRESS,LOD);                         07090000 T  0057:008:3
                    EMITO(INX);                                                  07091000 T  0057:008:3
                    EMITN(512);                                                  07092000 T  0057:008:1
                    EMITD(33,18,15);                                             07093000 T  0057:008:2
                    EMIT(0);                                                     07093100 T  0057:008:1
                    EMITD(5,5,1);                                                07093200 T  0057:008:2
COMMON:             PCTR ← PCTR+1;                                               07094000 T  0057:009:1
                    IF ELCLASS = COMMA THEN GO TO ANOTHER;                       07095000 T  0057:009:0
                    IF ELCLASS ≠ RTPAREN THEN                                    07096000 T  0057:009:3
                         BEGIN ERR(129); GO TO EXIT END;                         07097000 T  0057:009:0
                    IF TAKE(INDEX).NODIMPART+1 ≠ PCTR THEN                       07098000 T  0057:009:0
                         BEGIN ERR(128); GO TO EXIT END;                         07099000 T  0057:009:0
                    STEPIT;                                                      07100000 T  0057:010:0
                    STACKCT ← 0;                                                 07100500 T  0057:010:1
EXIT:          END ACTUAL PARAPART;                                              07101000 T  0057:010:2
                                                                          57 IS  106 LONG, NEXT SEG    3


               PROCEDURE PROCSTMT(FROM); VALUE FROM; BOOLEAN FROM;               07391000 T  0003:047:2
                    BEGIN                                                        07392000 T  0003:047:2
                         REAL HOLE,ADDRESS;                                      07393000 T  0003:047:2
                                                                          START OF SEGMENT **********   58
STACK(F+2) ← HOLE
STACK(F+3) ← ADDRESS
                         REAL J; LABEL OK;                                       07393100 T  0058:000:0
STACK(F+4) ← J
                         LABEL EXIT;                                             07394000 T  0058:000:0
                         SCATTERELBAT;                                           07395000 T  0058:000:0
                         HOLE← ELBAT[1];                                         07396000 T  0058:000:1
                         ADDRESS ← ADDRSF;                                       07397000 T  0058:000:3
                         IF NESTOG THEN                                          07397100 T  0058:000:0
                         IF MODE≠0 THEN                                          07397200 T  0058:000:2
                         IF TABLE(I+1)≠ASSIGNOP THEN                             07397210 T  0058:000:0
                         BEGIN FOR J←CALLINFO STEP 1 UNTIL CALLX DO              07397300 T  0058:000:1
                                   IF CALL[J]=ADDRESS THEN GO TO OK;             07397400 T  0058:000:0
                              CALL[CALLX←CALLX+1]←ADDRESS;                       07397500 T  0058:001:0
               OK:            END;                                             07397600 T  0058:001:1
                         CHECKER(HOLE);                                          07398000 T  0058:001:2
                         IF ELCLASS ≠ PROCID THEN                                07399000 T  0058:001:3
                         IF NOT FORMALF THEN                                     07400000 T  0058:002:1
                         IF TABLE(I+1) = ASSIGNOP THEN                           07401000 T  0058:002:3
                              BEGIN VARIABLE(2-REAL(FROM)); GO TO EXIT END;      07402000 T  0058:002:3
PRT(615) = *LIST, LABEL, OR SEGMENT DESCRIPTOR*
                    COMMENT  CALL VARIABLE TO HANDLE THIS ASSIGNMENT OPERATION;  07403000 T  0058:002:1
                         IF ELCLASS ≠ PROCID EQV FROM                            07404000 T  0058:002:1
                              THEN BEGIN ERR(159); GO TO EXIT END;               07405000 T  0058:002:2
                    COMMENT  IT IS PROCEDURE IF AND ONLY WE COME FROM STMT;      07406000 T  0058:003:3
                         STEPIT;                                                 07407000 T  0058:003:3
                         EMITO(MKS);                                             07408000 T  0058:003:0
                         IF ELCLASS = LEFTPAREN                                  07409000 T  0058:003:1
                              THEN ACTUALPARAPART(FALSE,GIT(HOLE))               07410000 T  0058:003:2
                              ELSE IF FORMALF THEN L ← L-1                       07411000 T  0058:003:5
```

```
                        ELSE IF TAKE(GIT(HOLE)).NODIMPART≠0 THEN ERR(128);          07412000 T  0058:0037:2
                EMITV(ADDRESS);                                                     07413000 T  0058:0042:0
        EXIT:   END  PROCSTMT;                                                      07425000 T  0058:0042:1
                                                                                58 IS   46 LONG, NEXT SEG    3


        PROCEDURE STRMPROCSTMT;                                                     07426000 T  0003:0479:2
                BEGIN REAL WHOLE,FIX,T1;                                            07427000 T  0003:0479:2
                                                                            START OF SEGMENT **********   59
STACK(F+2) = WHOLE
STACK(F+3) = FIX
STACK(F+4) = T1
                                                                                    07428000 T  0059:0000:0
                                                                                    07429000 T  0059:0000:0
        WHOLE ← ELBAT[I]; FIX ← -1;                                                 07430000 T  0059:0000:0
        IF ELCLASS ≠ STRPROCID THEN EMIT(0);                                        07431000 T  0059:0002:0
        IF WHOLE.LVL  ≠ 1 THEN                                                      07432000 T  0059:0004:0
             BEGIN FIX ← L; L ← L+1 END;                                            07433000 T  0059:0005:2
        EMITO(MKS);                                                                 07434000 T  0059:0007:3
        T1 ← TAKEFRST.[1:6];                                                        07435000 T  0059:0008:1
        FOR GT1 ← 1 STEP 1 UNTIL T1 DO EMIT(0);                                     07436000 T  0059:0010:0
        IF STEPI ≠ LEFTPAREN THEN ERR(128)                                          07437000 T  0059:0014:0
        ELSE BEGIN ACTUALPARAPART(TRUE,GIT(WHOLE));                                 07438000 T  0059:0015:3
             IF FIX < 0 THEN EMITV(WHOLE,ADDRESS)                                   07439000 T  0059:0018:0
             ELSE BEGIN T1 ← L; L ← FIX;                                            07440000 T  0059:0019:3
                  WHOLE ← TAKE(GIT(WHOLE));                                         07441000 T  0059:0022:1
                  EMITNUM(T1+2-WHOLE,[16:12]);                                      07442000 T  0059:0024:1
                  L ← T1;                                                           07443000 T  0059:0026:1
                  EMITB(BBW,BUMPL,WHOLE,[28:12]);                                   07444000 T  0059:0027:3
             END;                                                                   07445000 T  0059:0030:0
        END END STRMPROCSTMT;                                                       07446000 T  0059:0030:0
                                                                                59 IS   33 LONG, NEXT SEG    3


        INTEGER PROCEDURE BAE;                                                      07458000 T  0003:0479:2
                BEGIN BAE ← BUMPL; CONSTANTCLEAN; ADJUST END BAE;                   07459000 T  0003:0479:2


        COMMENT RELSESTMT COMPILES THE RELEASE STATEMENT;                           07460000 T  0003:0484:0
        COMMENT  DOSTMT HANDLES THE DO STATEMENT;                                   07481000 T  0003:0484:0
        PROCEDURE DOSTMT;                                                           07482000 T  0003:0484:0
PRT(616) = DOSTMT
                BEGIN INTEGER TL;                                                   07483000 T  0003:0484:0
                                                                            START OF SEGMENT **********   60
STACK(F+2) = TL
                FOULED ← L;                                                         07483500 T  0060:0000:0
                                                                                    07484000 T  0060:0000:1
             STEPIT; TL←L; STMT; IF ELCLASS ≠ UNTILV THEN ERR(131)                  07485000 T  0060:0000:1
             ELSE BEGIN                                                             07486000 T  0060:0004:0
               STEPIT; BEXP; EMITB(BBC,BUMPL,TL) END                                07487000 T  0060:0005:2
        END DOSTMT;                                                                 07488000 T  0060:0008:0
```

```
                                                                                                                60 IS   11 LONG, NEXT SEG    3

                COMMENT  WHILESTMT COMPILES THE WHILE STATEMENT;                                                    07489000 T  0003!048410
                PROCEDURE WHILESTMT;                                                                                07490000 T  0003!048410
PRT(617) = WHILESTMT
                        BEGIN INTEGER BACK,FRONT;                                                                   07491000 T  0003!048410
                                                                                                          START OF SEGMENT **********   61
STACK(F+2) = BACK
STACK(F+3) = FRONT
                            FOULED ← L;                                                                             07491500 T  0061!000010
                                                                                                                    07492000 T  0061!000011
                            STEPIT; BACK ← L; BEXP; FRONT ← BUMPL;                                                  07493000 T  0061!000011
                            IF ELCLASS ≠ DOV THEN ERR(132) ELSE                                                     07494000 T  0061!000410
                              BEGIN STEPIT; STMT; EMITB(BBW,BUMPL,BACK);                                            07495000 T  0061!000610
                                 CONSTANTCLEAN; EMITB(BFC,FRONT,L) END END WHILESTMT;                               07496000 T  0061!001010
                                                                                                                61 IS   14 LONG, NEXT SEG    3


                COMMENT  GOSTMT COMPILES GO TO STATEMENTS.  GOSTMT LOOKS AT THE                                     07497000 T  0003!048410
                         EXPRESSION.  IF IT IS SIMPLE ENOUGH WE GO DIRECTLY,                                        07498000 T  0003!048410
                         OTHERWISE A CALL ON THE MCP IS GENERATED IN ORDER TO GET                                   07499000 T  0003!048410
                         STORAGE RETURNED.  SEE DEXP AND GENGO;                                                     07500000 T  0003!048410
                    PROCEDURE GOSTMT;                                                                               07501000 T  0003!048410
PRT(620) = GOSTMT
                    BEGIN                                                                                           07502000 T  0003!048410
                        REAL ELBW;                                                                                  07503000 T  0003!048410
                                                                                                          START OF SEGMENT **********   62
STACK(F+2) = ELBW
                        LABEL GOMCP,EXIT;                                                                           07504000 T  0062!000010
                        IF STEPI = TOV THEN STEPIT;                                                                 07505000 T  0062!000010
                        IF ELCLASS = LABELID THEN TB1 ← TRUE                                                        07506000 T  0062!000210
                        ELSE IF ELCLASS = SWITCHID THEN TB1 ← FALSE                                                 07507000 T  0062!000313
                        ELSE BEGIN IF ELCLASS = POLISHV THEN                                                        07511000 T  0062!000610
                                BEGIN POLISHER(1); EMITO(BFW) END                                                   07512000 T  0062!000713
                                ELSE ERR(501);                                                                      07513000 T  0062!000913
                                GO TO EXIT                                                                          07514000 T  0062!001112
                            END;                                                                                    07515000 T  0062!001113
                        IF NOT LOCAL(ELBAT[I]) THEN                                                                 07516000 T  0062!001113
                            BEGIN                                                                                   07516100 T  0062!001211
                                IF TB1 THEN                                                                         07516200 T  0062!001312
                                    BEGIN EMITV(GNAT(ELBAT[I]));                                                    07516300 T  0062!001313
                                        EMITO(BFW);                                                                 07516400 T  0062!001513
                                        STEPIT;                                                                     07516500 T  0062!001610
                                        GO TO EXIT END;                                                             07516600 T  0062!001611
                            BEGIN ERR(501); GO TO EXIT END;                                                         07517000 T  0062!001712
                            END;                                                                                    07517500 T  0062!001811
                        IF TB1 THEN BEGIN GOGEN(ELBAT[I],BFW); STEPIT;                                              07518000 T  0062!001811
                            CONSTANTCLEAN; GO EXIT END                                                              07519000 T  0062!002112
                        ELSE BEGIN                                                                                  07520000 T  0062!002210
                            ELBW ← ELBAT[I];                                                                        07521000 T  0062!002211
                                                                                                                    07522000 T  0062!002313
                            BANA;                                                                                   07523000 T  0062!002313
```

```
                    EMITO(DUP);                                                  07524000 T  0062:002:0
                    EMITO(ADD);                                                  07525000 T  0062:002:1
                         EMITO(BFW);                                             07526000 T  0062:002:3
                    GT3 ← TAKE(GT4←GIT(ELBW))+GT4;                               07527000 T  0062:002:0
                    FOR GT4 ← GT4+1 STEP 1 UNTIL GT3 DO                          07528000 T  0062:002:2
                         GOGEN(TAKE(GT4),BFW);                                   07529000 T  0062:003:2
               END;                                                              07530000 T  0062:003:2
EXIT: END GOSTMT;                                                                07531000 T  0062:003:2
                                                                          62 IS   39 LONG, NEXT SEG    3


PROCEDURE GOGEN(LABELBAT,BRANCHTYPE);                                            07535000 T  0003:048:0
               VALUE LABELBAT,BRANCHTYPE;                                        07536000 T  0003:048:0
               REAL LABELBAT,BRANCHTYPE;                                         07537000 T  0003:048:0
               BEGIN                                                             07538000 T  0003:048:0
                 IF BOOLEAN(GT1←TAKE(GT2←GIT(LABELBAT))).[1:1]                   07539000 T  0003:048:0
                   THEN EMITB(BRANCHTYPE,BUMPL,GT1,[36:12])                      07540000 T  0003:048:3
   COMMENT  LABELR SETS THE SIGN OF THE ADDITIONAL INFO FOR A LABEL              07541000 T  0003:049:0
               NEGATIVE WHEN THE LABEL IS ENCOUNTERED.  SO THIS MEANS            07542000 T  0003:049:1
               THAT WE NOW KNOW WHERE TO GO;                                     07543000 T  0003:049:1
                    ELSE BEGIN EMIT(GT1); EMIT(BRANCHTYPE);                      07544000 T  0003:049:1
                    PUT(GT1&L[36:36:12],GT2) END END GOGEN;                      07545000 T  0003:049:2
```

```
               COMMENT SIMPGO IS USED ONLY BY THE IF STMT ROUTINE.  IT DETERMINES IF   07546000 T  0003:049:2
                       A STATEMENT IS A SIMPLE GO TO STATEMENT;                        07547000 T  0003:049:2
               BOOLEAN PROCEDURE SIMPGO;                                               07548000 T  0003:049:2
                       BEGIN LABEL EXIT;                                               07549000 T  0003:049:2
                                                                                START OF SEGMENT **********   63
                            IF ELCLASS = GOV                                           07550000 T  0063:000:0
                              THEN BEGIN                                               07551000 T  0063:000:0
                                 IF STEPI = TOV THEN STEPIT;                           07552000 T  0063:000:1:2
                                 IF ELCLASS = LABELID THEN                             07553000 T  0063:000:3:2
                                 IF LOCAL(ELBAT[I]) THEN                               07554000 T  0063:000:4:0
                                    BEGIN SIMPGO ← TRUE; GO EXIT END;                  07555000 T  0063:000:5:3
                                      I ← I-1; ELCLASS ← GOV END;                      07556000 T  0063:000:7:2
                EXIT:  END SIMPGO;                                                     07557000 T  0063:000:9:2
                                                                                 63 IS   13 LONG, NEXT SEG    3


               COMMENT IFSTMT COMPILES IF STATEMENTS.  SPECIAL CARE IS TAKEN TO        07558000 T  0003:049:5:2
                       OPTIMIZE CODE IN THE NEIGHBORHOOD OF THE JUMPS.  TO SOME        07559000 T  0003:049:5:2
                       EXTENT SUPPERFULOUS BRANCHING IS AVOIDED;                       07560000 T  0003:049:5:2
               PROCEDURE IFSTMT;                                                       07561000 T  0003:049:5:2
                       BEGIN REAL T1,T2; LABEL EXIT;                                   07562000 T  0003:049:5:2
                                                                                START OF SEGMENT **********   64
STACK(F+2) = T1
STACK(F+3) = T2
                            IFCLAUSE;                                                  07563000 T  0064:000:0
                            IF SIMPGO                                                  07564000 T  0064:000:1
                              THEN BEGIN                                               07565000 T  0064:000:1
                                 T1 ← ELBAT[I];                                        07566000 T  0064:000:1:3
```

```
                    IF STEPI = ELSEV                                                                07567000 T  0064:0002:1
                          THEN BEGIN                                                                07568000 T  0064:0003:2
                             STEPI;                                                                 07569000 T  0064:0004:0
                             IF SIMPGO                                                              07570000 T  0064:0004:1
                               THEN BEGIN                                                           07571000 T  0064:0004:1
                                 GOGEN(ELBAT[I],BFC); GOGEN(T1,BFW);                                07572000 T  0064:0005:3
                    STEPI; GO TO EXIT END ELSE BEGIN EMITLNG;GOGEN(T1,BFC);                         07573000 T  0064:0007:3
                         STMT ; GO TO EXIT  END END ;                                               07574000 T  0064:0010:1
                         EMITLNG; GOGEN(T1,BFC);                                                    07575000 T  0064:0011:3
                         GO EXIT END;                                                               07576000 T  0064:0013:2
                    T1 ← BUMPL; STMT;                                                               07577000 T  0064:0013:3
                    IF ELCLASS ≠ ELSEV THEN                                                         07578000 T  0064:0016:0
                      BEGIN IF L-T1>1023 THEN ADJUST; EMITB(BFC,T1,L);                              07579000 T  0064:0016:1
                                 GO EXIT END;                                                       07579100 T  0064:0020:1
                    STEPI;                                                                          07580000 T  0064:0021:2
                    IF SIMPGO                                                                       07581000 T  0064:0021:3
                      THEN BEGIN                                                                    07582000 T  0064:0021:3
                         T2 ← L; L ←T1-2;GOGEN(ELBAT[I],BFC); L ← T2;                               07583000 T  0064:0022:1
                         STEPI; GO EXIT END;                                                        07584000 T  0064:0026:1
                    T2 ← BUMPL; CONSTANTCLEAN;                                                      07585000 T  0064:0027:3
                    IF L-T1>1023 THEN ADJUST; EMITB(BFC,T1,L); STMT;                                07585100 T  0064:0030:0
                    IF L-T2>1023 THEN ADJUST; EMITB(BFW,T2,L);                                      07586000 T  0064:0034:0
EXIT;  END  IFSTMT;                                                                                 07587000 T  0064:0037:3
                                                                                          64 IS   41 LONG. NEXT SEG    3

                COMMENT  LABELR HANDLES LABELED STATEMENTS.  IT PUTS L INTO THE                     07588000 T  0003:0495:2
                         ADDITIONAL INFO AND MAKES ITS SIGN NEGATIVE.  IT COMPILES                  07589000 T  0003:0495:2
                         AT THE SAME TIME ALL THE PREVIOUS FORWARD REFERENCES SET                   07590000 T  0003:0495:2
                         UP FOR IT BY GOGEN.  (THE ADDITIONAL INFO LINKS TO A LIST                  07591000 T  0003:0495:2
                         IN THE CODE ARRAY OF ALL FORWARD REFERENCES);                              07592000 T  0003:0495:2
              PROCEDURE LABELR;                                                                     07593000 T  0003:0495:2
PRT(621) = LABELR
                    BEGIN LABEL EXIT, ROUND;                                                        07594000 T  0003:0495:2
                                                                                      START OF SEGMENT **********   65
            DEFINE FLBATWORD=RR9#,LINK=GT2#,INDEX=GT3#,ADDITIONAL                                   07595000 T  0065:0000:0
                         =GT4#,NEXTLINK=GT5#;                                                       07596000 T  0065:0000:0
                    REAL OLDL;                                                                      07596500 T  0065:0000:0
STACK(F+2) = OLDL
                    DO BEGIN OLDL ← L;                                                              07597000 T  0065:0000:0
                         IF STEPI ≠ COLON THEN                                                      07597500 T  0065:0000:1
                           BEGIN ERR(133); GO TO EXIT END;                                          07598000 T  0065:0001:3
                    XMARK(LBLREF); % THIS WILL SORT AHEAD OF DECLARATION %110-                       07598100 C  0065:0003:3
                                   % WHEN WE GET AROUND TO THE XREF.     %110-                    07598200 C  0065:0007:3
                         IF NOT LOCAL(ELBATWORD ← ELBAT[I-1])                                       07599000 T  0065:0007:3
                           THEN BEGIN FLAG(134); GO TO ROUND END;                                   07600000 T  0065:0008:1
                         IF STEPI = COLON THEN                                                      07600100 T  0065:0011:3
                           BEGIN I ← I-1; ADJUST END ELSE                                           07600200 T  0065:0012:1
                         IF ELCLASS = LITNO THEN L ← 4×C ELSE                                       07600300 T  0065:0015:2
                         IF ELCLASS=ASTRISK THEN                                                    07600400 T  0065:0018:0
                           BEGIN IF MODE ≠ 0 OR ASTOG THEN                                          07600410 T  0065:0019:2
                                   FLAG(505);                                                       07600420 T  0065:0021:2
                                 ASTOG ← TRUE;                                                      07600430 T  0065:0022:0
                                 L ← 4×PRTI;                                                        07600440 T  0065:0023:2
                               END ELSE                                                             07600450 T  0065:0024:0
```

```
            I ← I-2;                                                     07600500 T  0065:0024:0
            IF STEPI ≠ COLON THEN                                        07600600 T  0065:0026:0
                BEGIN ERR(133); GO TO EXIT END;                          07600700 T  0065:0027:2
            IF L < OLDL THEN                                             07600800 T  0065:0028:1
                BEGIN FLAG(504); GO TO ROUND END;                        07600900 T  0065:0029:3
        GT1 ← TABLE(I+1);                                                07600950 T  0065:0031:2
            LINK ← (ADDITIONAL ← TAKE(INDEX ← GIT(ELBATWORD)))           07601000 T  0065:0033:2
                .[36:12];                                                07602000 T  0065:0034:1
            IF ADDITIONAL < 0 THEN                                       07603000 T  0065:0036:0
                BEGIN FLAG(135); GO TO ROUND END;                        07604000 T  0065:0037:2
            FOULED ← L;                                                  07604010 T  0065:0038:1
            IF TABLE(I+1) = COLON THEN                                   07604020 T  0065:0039:3
                BEGIN                                                    07604030 T  0065:0041:2
                    IF LINK≠0 THEN BEGIN OLDL ← L;                       07604040 T  0065:0041:3
                    DO BEGIN  NEXTLINK ← GET(LINK);                      07604050 T  0065:0043:3
                              L ← LINK;                                  07604060 T  0065:0045:2
                              IF OLDL.[36:10]-L.[36:10]≥128              07604067 T  0065:0046:0
                              THEN FLAG(50) ELSE                         07604068 T  0065:0047:3
                            EMIT(OLDL-LINK&0[46:46:2]+                   07604070 T  0065:0049:3
                                  0&NEXTLINK[46:46:2]+3072);             07604080 T  0065:0052:0
                              L ← L-1;                                   07604085 T  0065:0054:0
                       END UNTIL LINK←LINK-NEXTLINK DIV 4=L;             07604090 T  0065:0055:3
                    L ← OLDL; END; STEPIT;                               07604100 T  0065:0058:0
                    DO IF STEPI ≤ STRNGCON AND ELCLASS ≥                 07604110 T  0065:0059:3
                         NONLITNO THEN EMITWORD(C)                       07604120 T  0065:0061:2
                         ELSE BEGIN ERR(500); I ← I-1 END                07604130 T  0065:0062:1
                    UNTIL STEPI ≠ COMMA;                                 07604140 T  0065:0067:2
                    I ← I-1 ;                                            07604150 T  0065:0068:1
                END ELSE                                                 07604160 T  0065:0069:3
            WHILE LINK ≠ 0                                               07605000 T  0065:0069:3
              DO BEGIN                                                   07606000 T  0065:0071:2
                 NEXTLINK ← GET(LINK-2);                                 07607000 T  0065:0072:0
                 IF L-LINK>1023 THEN ADJUST;                             07607100 T  0065:0074:0
                EMITB(GET(LINK-1),LINK,L);                               07608000 T  0065:0076:0
                 LINK ← NEXTLINK END;                                    07609000 T  0065:0078:1
            PUT(-ADDITIONAL&L[36:36:12],INDEX);                          07610000 T  0065:0079:3
ROUND:       ERRORTOG ← TRUE END UNTIL STEPI ≠ LABELID;                  07645000 T  0065:0082:0
  EXIT: END LABELR;                                                      07646000 T  0065:0084:0
                                                                    65 IS    88 LONG, NEXT SEG    3


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%   07646010 C  0003:0495:2
%*********************************************************************%   07646015 C  0003:0495:2
%*                                                                   *%   07646020 C  0003:0495:2
%*                              C A S E                              *%   07646025 C  0003:0495:2
%*                              = = = =                              *%   07646030 C  0003:0495:2
%*                                                                   *%   07646035 C  0003:0495:2
%*                         S T A T E M E N T                         *%   07646040 C  0003:0495:2
%*                         = = = = = = = = =                         *%   07646045 C  0003:0495:2
%*                                                                   *%   07646050 C  0003:0495:2
%*                                                                   *%   07646055 C  0003:0495:2
%* THIS PROCEDURE HANDLES THE CASE STATEMENT. THE SYNTAX FOR THE CASE *%   07646060 C  0003:0495:2
%* STATMENT IS:                                                      *%   07646065 C  0003:0495:2
%*                                                                   *%   07646070 C  0003:0495:2
%*      CASE <AEXP> OF BEGIN <COMPOUND TAIL>                         *%   07646075 C  0003:0495:2
```

```
%*
%* THE CODE GENERATED FOR THIS STATEMENT IS:
%*
%*                      <AEXP>
%*                      LITC  ?
%*                      BFW         BRANCH TO BRANCH TABLE
%*                      STMT 0
%*                      LITC  ?
%*                      BFW         BRANCH TO RESUME
%*                      STMT 1
%*                      LITC  ?
%*                      BFW         BRANCH TO RESUME
%*                      .
%*                      .
%*                      .
%*     BRANCH TABLE:
%*                      DUP
%*                      ADD
%*                      BFW         BRANCH INTO TABLE
%*                      LITC  ?
%*                      BBW         BRANCH TO STATEMENT 0
%*                      LITC  ?
%*                      BBW         BRANCH TO STATEMENT 1
%*                      .
%*                      .
%*                      .
%*                      LITC  ?
%*                      BBW         BRANCH TO STATEMENT N
%*     RESUME:
%*
%*
%* NOTICE THAT:
%*
%*  1)  THE  CASE INDEX IS NOT INTEGERIZED. IF YOU HAVE ANYTHING THAT
%*      MIGHT CAUSE THE  INDEX  TO  BE  UNNORMALIZED  YOU WILL HAVE TO
%*      INTEGERIZE IT YOURSELF.
%*
%*  2) NO RANGE CHECKING IS DONE ON THE  INDEX.  IF  YOU USE A NUMBER
%*      OUT OF RANGE THEN YOU WILL FALL OFF THE END OF THE WORLD.
%*
%*  3) IF A STATEMENT IN THE CASE COMPOUND TAIL IS NULL,  NO  CODE IS
%*      GENERATED  FOR  THE  STATEMENT.  INSTEAD,  THE BRANCH TABLE GOES
%*      DIRECTLY TO RESUME.
%*
%*  4) IF A STATEMENT  IN  THE  CASE COMPOUND TAIL IS A SIMPLE GO TO.
%*      THE STATEMENT DOES NOT HAVE A BRANCH TO RESUME AFTER IT.
%*
%*************************************************************************
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
                    PROCEDURE CASESTATEMENT;                          %115-
PRT(622) = CASESTATEMENT
                    BEGIN                                             %115-
                      ARRAY CASEADDRESS[0:99];    % THIS ARRAY HOLDS THE RELATIVE   %115-
                                                                  START OF SEGMENT **********   66
STACK(F+2) = CASEADDRESS
                                                  % ADDRESS OF THE BEGINNING OF EACH   %115-
                                                  % CASE.                              %115-

*%        07646080 C  0003:049512
*%        07646085 C  0003:049512
*%        07646090 C  0003:049512
*%        07646095 C  0003:049512
*%        07646100 C  0003:049512
*%        07646105 C  0003:049512
*%        07646110 C  0003:049512
*%        07646115 C  0003:049512
*%        07646120 C  0003:049512
*%        07646125 C  0003:049512
*%        07646130 C  0003:049512
*%        07646135 C  0003:049512
*%        07646140 C  0003:049512
*%        07646145 C  0003:049512
*%        07646150 C  0003:049512
*%        07646155 C  0003:049512
*%        07646160 C  0003:049512
*%        07646165 C  0003:049512
*%        07646170 C  0003:049512
*%        07646175 C  0003:049512
*%        07646180 C  0003:049512
*%        07646185 C  0003:049512
*%        07646190 C  0003:049512
*%        07646195 C  0003:049512
*%        07646200 C  0003:049512
*%        07646205 C  0003:049512
*%        07646210 C  0003:049512
*%        07646215 C  0003:049512
*%        07646220 C  0003:049512
*%        07646225 C  0003:049512
*%        07646230 C  0003:049512
*%        07646235 C  0003:049512
*%        07646240 C  0003:049512
*%        07646245 C  0003:049512
*%        07646250 C  0003:049512
*%        07646255 C  0003:049512
*%        07646260 C  0003:049512
*%        07646265 C  0003:049512
*%        07646270 C  0003:049512
*%        07646275 C  0003:049512
*%        07646280 C  0003:049512
*%        07646285 C  0003:049512
*%        07646290 C  0003:049512
*%        07646295 C  0003:049512
*%        07646300 C  0003:049512
*%        07646305 C  0003:049512
*%        07646310 C  0003:049512
*%        07646315 C  0003:049512
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
          07646320 C  0003:049512
          07646325 C  0003:049512
          07646330 C  0003:049512
          07646335 C  0003:049512
          07646340 C  0066:000113
          07646345 C  0066:000113
```

```
                    INTEGER
                      LINK,                          % HOLDS RELATIVE ADDRESS OF FIRST OF %115-  07646350 C  0066:000113
STACK(F+3) = LINK                                    % BRANCHES THAT MUST BE FIXED UP TO  %115-  07646355 C  0066:000113
                                                     % BRANCH TO THE RESUME POINT.        %115-  07646360 C  0066:000113
                      ADR,                           % HOLDS RELATIVE ADDRESS OF THE BRANCH      07646365 C  0066:000113
STACK(F+4) = ADR                                     % AROUND THE CASE STATEMENTS TO THE  %115-  07646370 C  0066:000113
                                                     % BRANCH TABLE. THIS BRANCH GETS FIXED      07646375 C  0066:000113
                                                     % UP WHEN WE FIND WHERE THE BRANCH TABLE    07646380 C  0066:000113
                                                     % GOES.                             %115-  07646385 C  0066:000113
                      N,                             % COUNT OF NUMBER OF CASE STATEMENT  %115-  07646390 C  0066:000113
STACK(F+5) = N                                       % ENCOUNTERED.                       %115-  07646395 C  0066:000113
                      ENDOFIT,                       % ADDRESS OF RESUME POINT            %115-  07646400 C  0066:000113
STACK(F+6) = ENDOFIT                                 % TEMPORARY                          %115-  07646405 C  0066:000113
                      J,                             % TEMPORARY                          %115-  07646410 C  0066:000113
STACK(F+7) = J                                       %                                   %115-  07646415 C  0066:000113
                      K;                             % TRUE IF WE ARE COMPILING A SIMPLE  %115-  07646420 C  0066:000113
STACK(F+10) = K                                      % GO TO.                             %115-  07646425 C  0066:000113
                                                                                         %115-  07646430 C  0066:000113
                    BOOLEAN                                                              %115-  07646435 C  0066:000113
                      GOTOG;                                                             %115-  07646436 C  0066:000113
STACK(F+11) = GOTOG                                                                      %115-  07646440 C  0066:000113
                                                                                         %115-  07646445 C  0066:000113
                    LABEL XIT;                                                           %115-  07646450 C  0066:000113
                  %                                                                      %115-  07646455 C  0066:000210
                  %                                                                      %115-  07646456 C  0066:000211
                      STEPIT;    % STEP OVER "CASE"                                     %115-  07646457 C  0066:000313
                      AEXP;      % GENERATE CODE FOR CASE INDEX                          %115-  07646458 C  0066:000410
                      IF STEPI ≠ BEGINV THEN % NOTICE WE JUST JUMPED OVER "OF"          %115-  07646459 C  0066:000513
                        BEGIN                                                            %115-  07646460 C  0066:000513
                          ERR(70); GO TO XIT;                                            %115-  07646465 C  0066:000610
                        END;                                                             %115-  07646470 C  0066:000712
                    EMIT(0);  % GENERATE DUMMY BRANCH TO BRANCH TABLE. WILL FIX          %115-  07646475 C  0066:000713
                    EMITO(BFW);  %    IT UP LATER.                                       %115-  07646480 C  0066:000913
                    ADR := L;                                                            %115-  07646485 C  0066:000913
                    WHILE STEPI ≠ ENDV DO                                                %115-  07646490 C  0066:001010
                      BEGIN   % PROCESSING CASE STATEMENTS                               %115-  07646495 C  0066:001112
                        ERRORTOG := TRUE;                                                %115-  07646500 C  0066:001210
                        IF ELCLASS = SEMICOLON THEN % NULL STATEMENT, NO CODE NEEDED      07646505 C  0066:001211
                          N := N + 1  % THIS LEAVES A ZERO IN CASEADDRESS[N]              %115-  07646510 C  0066:001312
                        ELSE                                                             %115-  07646515 C  0066:001411
                          BEGIN                                                          %115-  07646520 C  0066:001513
                            CASEADDRESS[N] := L; % REMEMBER BEGINNING ADDRESS OF         %115-  07646525 C  0066:001611
                            N := N + 1;          % THIS CASE.                            %115-  07646530 C  0066:002010
                            IF (GOTOG := SIMPGO) THEN                                    %115-  07646535 C  0066:002011
                              ELBAT[I:=I-1] := ELCLASS := GOV; % REMEMBER IF SIMPLE      %115-  07646540 C  0066:002112
                            STMT; % PROCESS THE STATEMENT                                %115-  07646545 C  0066:002210
                            IF ELCLASS = SEMICOLON THEN                                  %115-  07646550 C  0066:002211
                              IF NOT GOTOG THEN % GENERATE DUMMY BRANCH TO RESUME        %115-  07646555 C  0066:002313
                                BEGIN                                                    %115-  07646560 C  0066:002410
                                  EMIT(LINK);                                            %115-  07646565 C  0066:002512
                                  EMITO(BFW);                                            %115-  07646570 C  0066:002512
                                  LINK := L;                                             %115-  07646575 C  0066:002512
                                END                                                      %115-
                              ELSE % SIMPLE GO TO, NO CODE TO GENERATE                   %115-
                            ELSE                                                         %115-
```

```
            IF ELCLASS ≠ ENDV THEN                                          %115-          07646580 C  0066:0025:3
              ERR(71);                                                      %115-          07646585 C  0066:0026:1
        END;                                                                %115-          07646590 C  0066:0028:0
    END OF WHILE LOOP;                                                      %115-          07646595 C  0066:0028:0
  ENDTOG := TRUE;      % SKIP OVER COMMENT AFTER END                       %115-          07646601 C  0066:0028:1
  DO STOPDEFINE := TRUE UNTIL                                               %115-          07646602 C  0066:0029:2
    STEP1 ≤ ENDV AND ELCLASS ≥ UNTILV OR NOT ENDTOG;                        %115-          07646603 C  0066:0030:1
  ENDTOG := FALSE;                                                          %115-          07646604 C  0066:0034:0
  EMITB(BFW,ADR,L);     % FIX UP BRANCH TO BRANCH TABLE                     %115-          07646605 C  0066:0034:1
  EMIT0(DUP);           % GENERATE CODE TO MULTIPLY INDEX                   %115-          07646610 C  0066:0036:0
  EMIT0(ADD);           % BY TWO AND BRANCH INTO BRANCH TABLE               %115-          07646615 C  0066:0036:1
  EMIT0(BFW);                                                               %115-          07646620 C  0066:0037:3
  ENDOFIT := L + 2×N; % CALCULATE WHERE RESUME IS                           %115-          07646625 C  0066:0038:0
  WHILE (J:=J+1) ≤ N DO  % GENERATE THE BRANCH TABLE                        %115-          07646630 C  0066:0040:0
    EMITB(BBW,L:=L+2,IF (K:=CASEADDRESS[J-1]) = 0 THEN                      %115-          07646635 C  0066:0042:0
                          ENDOFIT ELSE K);                                  %115-          07646640 C  0066:0046:0
  J := LINK; % TO MAKE THE LOOP WORK                                        %115-          07646645 C  0066:0048:0
  WHILE (LINK:=J) ≠ 0 DO                                                    %115-          07646650 C  0066:0049:2
    BEGIN  % FIXING UP BRANCHES TO RESUME                                   %115-          07646655 C  0066:0050:1
      J := GET(LINK-2);  % LOCATION OF NEXT BRANCH TO FIX                   %115-          07646660 C  0066:0050:1
      EMITB(BFW,LINK,L);                                                    %115-          07646665 C  0066:0052:1
    END;                                                                    %115-          07646670 C  0066:0053:3
XIT:                                                                        %115-          07646674 C  0066:0054:0
END OF CASE STATEMENT;                                                      %115-          07646675 C  0066:0055:2
                                                                                      66 IS    60 LONG, NEXT SEG    3
```

```
            PROCEDURE FILLSTMT(SIZE); VALUE SIZE; INTEGER SIZE;                                07647000 T  0003:0495:2
PRT(623) = FILLSTMT
                BEGIN                                                                          07647500 T  0003:0495:2
                COMMENT  "COCT" PERFORMS THE OCTAL CONVERT FOR THE FILL STATEMENT.             07648000 T  0003:0495:2
                         IF THERE ARE ANY NON-OCTAL DIGITS, THIS PROCEDURE RETURNS             07648500 T  0003:0495:2
                         A ZERO AND THEN THE 3 LOW-ORDER BITS OF THE BAD DIGIT ARE             07649000 T  0003:0495:2
                         RESET AND IGNORED AND ERROR NUMBER 303 IS PRINTED. "COCT"             07649500 T  0003:0495:2
                         ALLOWS FLAG BITS TO BE SET, WHEREAS "OCTIZE" DOES NOT.                07650000 T  0003:0495:2
                           N       NUMBER OF CHARACTERS TO BE CONVERTED.                       07650500 T  0003:0495:2
                           SKBIT   NUMBER OF BITS TO SKIP BEFORE STARTING CONVERSION.          07651000 T  0003:0495:2
                                   THIS IS BECAUSE THE NO. OF CHARS. MAY BE LESS THAN          07651500 T  0003:0495:2
                                   8 AND IT MUST BE RIGHT-JUSTIFIED IN CD(CODEFILE).           07652000 T  0003:0495:2
                           ACC     ADDRESS OF THE ACCUM WHERE ALPHA INFO IS KEPT.              07652500 T  0003:0495:2
                ;                                                                              07653000 T  0003:0495:2
                   REAL STREAM PROCEDURE COCT(N,SKBIT,ACC,CD);VALUE N,SKBIT;                   07653500 T  0003:0495:2
                                                                                  START OF SEGMENT **********   67
PRT(624) = COCT
                   BEGIN                                                                       07654000 T  0067:0000:0
                   S1:=ACC; S1:=S1+6;  DI:=CD;  DS:=8 LIT"00000000";                           07654500 T  0067:0000:0
                   DI:=CD ; SKIP SKBIT DB;TALLY:=1;                                            07655000 T  0067:0002:0
                   N(IF SC>"7"THEN TALLY:=0; SKIP 3 SB;                                        07655500 T  0067:0003:2
                     3(IF SB   THEN DS:=1 SET ELSE SKIP 1 DB; SKIP 1 SB));                     07656000 T  0067:0005:2
                   COCT:=TALLY                                                                 07656500 T  0067:0007:2
                   END COCT;                                                                   07657000 T  0067:0007:3



                   REAL T2;                                                                    07657500 T  0067:0008:1
```

```
STACK(F+2) = T2
                    LABEL L1;
                    STREAM PROCEDURE ZEERO(D);                                          07658000 T  0067:0008:1
                                                                                        07658500 T  0067:0008:1
PRT(625) = ZEERO
                         BEGIN                                                          07659000 T  0067:0008:1
                         D1:=D:DS:=8 LIT"00000000";                                     07659500 T  0067:0009:2
                         S1:=D:31(32(DS:=WDS)); DS:=30 WDS;                             07660000 T  0067:0010:1
                         END ZEERO;                                                     07660500 T  0067:0012:0



                         XMARK(ASSIGNREF); % FILL STATEMENT                    %110-    07660600 C  0067:0012:1
                         STREAMTOG:=BOOLEAN(2);                                         07661000 T  0067:0017:2
                         SEGMENTSTART(TRUE);                                            07661500 T  0067:0018:0
                         IF STEPI#ASSIGNOP THEN ZEERO(CODE(1))                          07662000 T  0067:0018:1
                    ELSE      BEGIN                                                     07662500 T  0067:0023:2
                              FOR T2:=1 STEP 1 UNTIL SIZE DO                            07663000 T  0067:0024:0
                                   BEGIN                                                07663500 T  0067:0025:2
                                   IF STEPI>IDMAX THEN                                  07664000 T  0067:0025:2
                                        BEGIN                                           07664500 T  0067:0026:0
                                        IF ELCLASS#LITNO AND ELCLASS#NONLITNO THEN      07665000 T  0067:0026:1
                                          IF ELCLASS#STRNGCON THEN                      07665500 T  0067:0028:0
                                            IF ELCLASS=ADOP AND                         07666000 T  0067:0029:3
                                              (STEPI=NONLITNO OR ELCLASS=LITNO) THEN    07666500 T  0067:0030:1
                                                C:=C & ELBAT[I-1][1:21:1]               07667000 T  0067:0033:2
                                        ELSE      BEGIN  ERROR(302); GO TO L1  END;     07667500 T  0067:0035:3
                                        IF ELCLASS=STRNGCON AND COUNT=8 THEN            07668000 T  0067:0037:3
                                          MOVECHARACTERS(8,ACCUM[1],3,CODE(T2),0)       07668500 T  0067:0039:3
                                        ELSE MOVE(1,C,CODE(T2))                         07669000 T  0067:0044:0
                                        END                                             07669500 T  0067:0048:1
                                   ELSE IF COUNT≤19 AND ACCUM[1].[18:18]="OCT" THEN     07670000 T  0067:0049:2
                                        BEGIN                                           07670500 T  0067:0052:0
                                        IF OCT(COUNT-3,48-(COUNT-3)×3,ACCUM[1],         07671000 T  0067:0052:1
                                          CODE(T2))=0 THEN FLAG(303)                    07671500 T  0067:0056:1
                                        END                                             07672000 T  0067:0061:2
                                        ELSE BEGIN  ERROR(302); GO TO L1  END ;         07672500 T  0067:0061:3
                                        IF STEPI#COMMA THEN GO TO L1                    07673000 T  0067:0064:0
                                        END;                                            07673500 T  0067:0065:2
                                   ERROR(54);                                           07674000 T  0067:0068:0
                                   END;                                                 07674500 T  0067:0068:1
                    L1:                                                                 07675000 T  0067:0068:1
                         RIGHT(SIZE×4);                                                 07675500 T  0067:0069:2
                         STREAMTOG:=FALSE;                                              07676000 T  0067:0070:0
                         SEGMENT(SIZE,0);                                               07676500 T  0067:0071:2
                         PROGDESCBLDR(ADDRSF,TRUE,SIZE,DDES);                           07677000 T  0067:0072:0
                         END FILLSTMT;                                                  07677500 T  0067:0073:3
                                                                                  67 IS   76 LONG, NEXT SEG    3



                    PROCEDURE STMT;                                                     07711000 T  0003:0495:2
                              BEGIN LABEL                                               07712000 T  0003:0495:2
                                                                                START OF SEGMENT **********   68
                                   L1, L2, L3, L4, L5, L6, L7, L8, L9, L10,             07713000 T  0068:0000:0
                                   L11, L12, L13, L14, L15, L16, L17, L18, L19, L20,    07714000 T  0068:0000:0
```

```
                L21, L22, L23, L24, L25, L26, L27, L28, L29, L30,                07715000 T  0068:000:0
                L31, L32, L33, L34, L35, L36, L37, L38, L39, L40,                07716000 T  0068:000:0
                L41, L42, L43, L44, L45, L46, L47, L48, L49, L50,                07717000 T  0068:000:0
                L51, L52, L53, L54;                                              07718000 T  0068:000:0
                SWITCH S :=                                                      07719000 T  0068:000:0
                L1,  L2,  L3,  L4,  L5,  L6,  L7,  L8,  L9,  L10,                07720000 T  0068:002:1
                L11, L12, L13, L14, L15, L16, L17, L18, L19, L20,                07721000 T  0068:002:1
                L21, L22, L23, L24, L25, L26, L27, L28, L29, L30,                07722000 T  0068:002:1
                L31, L32, L33, L34, L35, L36, L37, L38, L39, L40,                07723000 T  0068:002:1
                L41, L42, L43, L44, L45, L46, L47, L48, L49, L50,                07724000 T  0068:002:1
                L51, L52, L53, L54;                                              07725000 T  0068:002:1
                LABEL AGAIN,EXIT;                                                07726000 T  0068:003:0
                STACKCT := 0;                                                    07726990 T  0068:003:0
AGAIN;          GO TO S[ELCLASS];                                                07727000 T  0068:003:3
                IF ELCLASS = COLON THEN                                          07727010 T  0068:003:0
                     BEGIN STEPIT; GT1 := L;                                     07727020 T  0068:003:1
                          IF ELCLASS = COLON THEN                                07727030 T  0068:003:1
                               BEGIN ADJUST; I := I-1 END                        07727040 T  0068:003:2
                          ELSE IF ELCLASS = LITNO THEN L := 4*C                  07727050 T  0068:003:3
                          ELSE I := I-1;                                         07727060 T  0068:004:3
                          IF L < GT1 OR STEPI # COLON THEN                       07727070 T  0068:004:0
                               BEGIN ERR(504); GO TO EXIT END;                   07727080 T  0068:004:0
                          STEPIT;                                                07727090 T  0068:004:0
                          GO TO AGAIN;                                           07727100 T  0068:004:1
                     END;                                                        07727110 T  0068:004:2
                IF ELCLASS = 0 THEN FLAG(100); FLAG(145);                        07728000 T  0068:004:2
L1:L2:L3:L4:L5:L6:L9:L11:L13:L14:L15:L16:L17:L20:L21:L25:L28:L29:L24:             07729000 T  0068:005:3
L33:L34:L35:L36:L37:L39:                                                         07730000 T  0068:005:0
                ERR(144); GO TO EXIT;                                            07732000 T  0068:005:0
   L7:L8:                                                                        07732000 T  0068:005:2
                SUBHAND(TRUE); GO TO EXIT;                                       07733000 T  0068:005:0
   L10:L18:L19:                                                                  07734000 T  0068:005:2
                PROCSTMT(TRUE); GO TO EXIT;                                      07735000 T  0068:005:0
   L12:                                                                          07736000 T  0068:005:2
                STRMPROCSTMT; GO TO EXIT;                                        07737000 T  0068:005:0
   L22:L23:L26:L27:L30:L31:                                                      07738000 T  0068:005:2
                VARIABLE(FS); GO TO EXIT;                                        07739000 T  0068:005:2
   L32:                                                                          07740000 T  0068:006:0
                LABELR; GO TO AGAIN;                                             07741000 T  0068:006:2
   L38:                                                                          07742000 T  0068:006:0
                POLISHER(0); GO TO EXIT;                                         07743000 T  0068:006:0
   L40:                                                                          07744000 T  0068:006:2
                IF ELBAT[I].ADDRESS = STREAMV THEN                               07745000 T  0068:006:0
                     BEGIN INLINE; GO TO EXIT END;                               07746000 T  0068:006:3
                FLAG(146);                                                       07747000 T  0068:006:2
                IF TABLE(I-2) = ENDV AND MODE > 0 THEN                           07748000 T  0068:006:3
                     BEGIN I := I-2; ELCLASS := ENDV; GO TO EXIT END;            07749000 T  0068:007:1
                I := I-1; ERRORTOG := TRUE; BLOCK(FALSE);                        07750000 T  0068:007:3
                ELCLASS := TABLE(I:=I-1); GO TO EXIT;                            07751000 T  0068:007:0
   L42:                                                                          07752000 T  0068:007:2
                DBLSTMT; GO TO EXIT;                                             07753000 T  0068:007:2
   L43:                                                                          07754000 T  0068:008:0
                FORSTMT; GO TO EXIT;                                             07755000 T  0068:008:0
   L44:                                                                          07756000 T  0068:008:2
                WHILESTMT; GO TO EXIT;                                           07757000 T  0068:008:2
   L45:                                                                          07758000 T  0068:008:0
                DOSTMT; GO TO EXIT;                                              07759000 T  0068:008:0
```

```
   L49:
                CASESTATEMENT; GO TO EXIT;                                                %115-          07759100 C  0068:0083:2
                                                                                          %115-          07759200 C  0068:0083:2
   L51:                                                                                                  07760000 T  0068:0084:0
                IFSTMT; GO TO EXIT;                                                                      07761000 T  0068:0084:0
   L52:                                                                                                  07762000 T  0068:0085:2
                GOSTMT; GO TO EXIT;                                                                      07763000 T  0068:0085:2
   L53:                                                                                                  07764000 T  0068:0086:0
                IOSTMT; GO TO EXIT;                                                                      07765000 T  0068:0086:0
   L54:                                                                                                  07766000 T  0068:0087:2
                IF STEPI = DECLARATORS THEN                                                              07767000 T  0068:0087:2
           BEGIN                                                                                         07768000 T  0068:0088:0
                 IF ELBAT[I].ADDRESS = STREAMV THEN IF STEPI =                    % 6                  07768100 T  0068:0088:1
                   LEFTPAREN THEN                                                 % 6                  07768110 T  0068:0091:2
           BEGIN                                                                  % 6                  07768120 T  0068:0091:3
                 ELCLASS←TABLE(I←I-1) ;                                                                  07768130 T  0068:0092:0
                 COMPOUNDTAIL ;                                                                          07768140 T  0068:0094:0
                 GO TO EXIT ;                                                                            07768160 T  0068:0094:1
           END   ELSE I ← I - 1;                                                  % 6                  07768170 T  0068:0095:2
                 I ← I - 1;                                                     % 6                  07768180 T  0068:0097:2
                 BLOCK(FALSE);     END ELSE COMPOUNDTAIL;                                                07768200 T  0068:0098:0
L46:L47:L48:L50:                                                                                         07769000 T  0068:0100:0
L41:                                                                                      %115-          07770000 P  0068:0100:0
EXIT: END STMT;                                                                                          07771000 T  0068:0100:0
                                                                                                  68 IS  101 LONG. NEXT SEG    3
```

```
                PROCEDURE IOSTMT;                                                                        07991000 T  0003:0495:2
                        IF STEPI # LITNO OR (GT1←ELBAT[I].ADDRESS)>15 THEN ERR(98)ELSE                   07993000 T  0003:0495:2
                        BEGIN EMIT(ELBAT[I-1].ADDRESS&GT1[41:47:1]&GT1[36:44:3]);                        07994000 T  0003:0495:2
                              STEPIT                                                                     07995000 T  0003:0500:1
                        END SCOPE STATEMENT;                                                             07996000 T  0003:0505:2
                                                                                                         07997000 T  0003:0505:2



                PROCEDURE FORSTMT;                                                                       08008000 T  0003:0505:3
                        BEGIN                                                                            08009000 T  0003:0505:3
                         OWN REAL B,STMTSTART,REGO,RETURNSTORE,ADDRES,V,VRET,                            08010000 T  0003:0505:3
                                                                                          START OF SEGMENT **********   69
PRT(626) = B
PRT(627) = STMTSTART
PRT(630) = REGO
PRT(631) = RETURNSTORE
PRT(632) = ADDRES
PRT(633) = V
PRT(634) = VRET
                                 BRET;                                                                   08011000 T  0069:0000:0
PRT(635) = BRET
                        OWN BOOLEAN SIGNA,SIGNB,SIGNC, INT,                                              08012000 T  0069:0000:0
PRT(636) = SIGNA
PRT(637) = SIGNB
PRT(640) = SIGNC
PRT(641) = INT
                                 CONSTANA,CONSTANB,CONSTANC;                                             08013000 T  0069:0000:0
PRT(642) = CONSTANA
```

```
PRT(643) = CONSTANB
PRT(644) = CONSTANC
                        DEFINE SIMPLEB = SIGNC#,          FORMALV = SIGNA#,          08014000 T  0069:000:0
                               SIMPLEV = CONSTANA#, A = V#, Q = REGO#,              08015000 T  0069:000:0
                               OPDC = TRUE#, DESC = FALSE#, K = BRET#;              08016000 T  0069:000:0
                        LABEL EXIT;                                                 08017000 T  0069:000:0
                COMMENT PLUG EMITS EITHER AN OPERAND CALL ON A VARIABLE OR A CALL ON A  08018000 T  0069:000:0
                        CONSTANT DEPENDING ON THE REQUIREMENTS;                     08019000 T  0069:000:0
                PROCEDURE PLUG(C,A); VALUE C,A; REAL A; BOOLEAN C;                  08020000 T  0069:000:0
PRT(645) = PLUG
                        IF C THEN EMITNUM(A) ELSE EMITV(A,ADDRESS);                 08021000 T  0069:000:0



                COMMENT SIMPLE DETERMINES IF AN ARITHMETIC EXPRESSION IS + OR - A    08022000 T  0069:000:3
                        CONSTANT OR A SIMPLE VARIABLE.  IT MAKES A THROUGH REPORT    08023000 T  0069:000:3
                        ON ITS ACTIVITY.  IT ALSO MAKES PROVISION FOR THE RESCAN     08024000 T  0069:000:3
                        OF ELBAT (THIS IS THE ACTION WITH K = SEE CODE IN THE        08025000 T  0069:000:3
                        TABLE ROUTINE FOR FURTHER DETAILS);                         08026000 T  0069:000:3
                BOOLEAN PROCEDURE SIMPLE(B,A,S); BOOLEAN B,S; REAL A;               08027000 T  0069:000:3
PRT(646) = SIMPLE
                        BEGIN                                                       08028000 T  0069:000:3
                            S ← IF STEPI ≠ ADOP THEN FALSE ELSE ELBAT[I].ADDRESS    08029000 T  0069:000:3
                                        = SUB;                                      08030000 T  0069:000:6:1
                            IF ELCLASS = ADOP THEN STEPIT;                          08031000 T  0069:000:8:1
                            IF ELCLASS ≥ NONLITNO AND ELCLASS ≤ STRNGCON            08032000 T  0069:001:0:0
                              THEN BEGIN K ← K+1; SIMPLE ← TRUE;                    08033000 T  0069:001:1:2
                                 ELBAT[I] ← 0&COMMENTV[2:41:7]&K[16:37:11];         08034000 T  0069:001:4:1
                                 INFO[0,K] ← A ← C; B ← TRUE END                    08035000 T  0069:001:7:3
                            ELSE BEGIN                                              08036000 T  0069:002:1:3
                              B ← FALSE; A ← ELBAT[I];                              08037000 T  0069:002:2:0
                              SIMPLE ← REALID ≤ ELCLASS AND ELCLASS ≤ INTID END;    08038000 T  0069:002:4:0
                        STEPIT END SIMPLE;                                          08039000 T  0069:002:6:1



                COMMENT TEST EMITS THE STEP-UNTIL ELEMENT TEST;                     08040000 T  0069:002:9:2
                PROCEDURE TEST;                                                     08041000 T  0069:002:9:2
PRT(647) = TEST
                        BEGIN                                                       08042000 T  0069:002:9:2
                            IF NOT CONSTANB THEN                                    08043000 T  0069:002:9:2
                              BEGIN EMITO(SUB); IF SIMPLEB THEN EMITV(B,ADDRESS)    08044000 T  0069:003:0:1
                                 ELSE BEGIN                                         08045000 T  0069:003:2:1
                                   EMITL(2+L-BRET);                                 08046000 T  0069:003:4:0
                                   EMITB(BBW,BUMPL,B);                              08047000 T  0069:003:6:0
                                 END;                                               08048000 T  0069:003:8:0
                                 EMITO(MUL); EMIT(0) END;                           08049000 T  0069:003:8:0
                                 EMITO(IF SIGNB THEN GEQ ELSE LEQ); EMIT (0); L←L-1 08050000 T  0069:003:9:3
                        END TEST;                                                   08051000 T  0069:004:3:2



                BOOLEAN PROCEDURE SIMPI(ALL); VALUE ALL; REAL ALL;                  08052000 T  0069:004:4:0
PRT(650) = SIMPI
```

```
                        BEGIN                                                                    08053000 T  0069:004:0
                             CHECKER(VRET+ALL);                                                  08054000 T  0069:004:0
                             ADDRES ← ALL.ADDRESS;                                               08055000 T  0069:004:0
                             FORMALV ← ALL.[9:2] = 2;                                            08056000 T  0069:004:3
                             IF T ← ALL.CLASS > INTARRAYID OR T < BOOID OR                       08057000 T  0069:004:2
                                GT1 ← (T-BOOID) MOD 4 < 1 THEN                                   08058000 T  0069:005:0
                                  ERR(REAL(T ≠ 0) × 51 + 100);                                   08059000 T  0069:005:1
                             INT ← GT1 = 2;                                                      08060000 T  0069:005:2
                             SIMPI ← T ≤ INTID END SIMPI;                                        08061000 T  0069:005:1


                   COMMENT  STORE EMITS THE CODE FOR THE STORE INTO THE FOR INDEX;               08062000 T  0069:006:0
                   PROCEDURE STORE(S); VALUE S; BOOLEAN S;                                     08063000 T  0069:006:0
PRT(651) = STORE
                        BEGIN                                                                    08064000 T  0069:006:0
                          IF FORMALV THEN BEGIN EMITO(XCH); S ← FALSE END                        08065000 T  0069:006:0
                           ELSE BEGIN                                                            08066000 T  0069:006:2
                             EMITL(ADDRES);                                                      08067000 T  0069:006:3
                             IF ADDRES  > 1023 THEN EMITO(PRTE) END;                             08068000 T  0069:006:1
                              T  ← (REAL(S)+1)×16;                                               08069000 T  0069:006:1
                             EMITO((IF INT THEN T+512 ELSE 4×T)+4) END STORE;                    08070000 T  0069:007:0

                   COMMENT CALL EFFECTS A CALL ON THE INDEX;                                     08071000 T  0069:007:0
                   PROCEDURE CALL(S); VALUE S; BOOLEAN S;                                      08072000 T  0069:007:0
PRT(652) = CALL
                        BEGIN                                                                    08073000 T  0069:007:0
                         IF SIMPLEV                                                              08074000 T  0069:007:0
                           THEN IF S THEN EMITV(ADDRES) ELSE EMITN(ADDRES)                       08075000 T  0069:007:2
                           ELSE BEGIN                                                            08076000 T  0069:007:0
                             EMITL(2+L-VRET);                                                    08077000 T  0069:007:2
                             EMITB(BBW,BUMPL,V);                                                 08078000 T  0069:008:1
                             IF S THEN EMITO(LOD) END END CALL;                                  08079000 T  0069:008:2


                   PROCEDURE FORLIST(NUMLE); VALUE NUMLE; BOOLEAN NUMLE;                         08080000 T  0069:008:1
PRT(653) = FORLIST
                        BEGIN                                                                    08081000 T  0069:008:1
                   PROCEDURE FIX(STORE,BACK,FORWART,START);                                      08082000 T  0069:008:1
                                                                                          START OF SEGMENT **********   70
PRT(654) = FIX
                        VALUE STORE,BACK,FORWART,START;                                          08083000 T  0070:000:0
                        REAL  STORE,BACK,FORWART,START;                                          08084000 T  0070:000:0
                        BEGIN                                                                    08085000 T  0070:000:0
                             EMITB(GET(FORWART-1),FORWART,START);                                08086000 T  0070:000:0
                             IF RETURNSTORE ≠ 0                                                  08087000 T  0070:000:2
                               THEN BEGIN                                                        08088000 T  0070:000:2
                                  L ← STORE; EMITNUM(B-BACK);                                    08089000 T  0070:000:3
                                  EMITPAIR(RETURNSTORE,STD) END END FIX;                         08090000 T  0070:000:5
```

```
                        INTEGER BACKFIX, FORWARDBRANCH, FOOT, STOREFIX;                    08091000 T  0070:0006:1
STACK(F+2) = BACKFIX
STACK(F+3) = FORWARDBRANCH
STACK(F+4) = FOOT
STACK(F+5) = STOREFIX
                        LABEL BRNCH,EXIT;                                                  08092000 T  0070:0006:1
                        STOREFIX ← L; Q ← REAL(MODE=0)+3;                                  08093000 T  0070:0006:1
                        FOR K ← 1 STEP 1 UNTIL Q DO EMITO(NOP);                            08094000 T  0070:0009:3
                        IF NUMLE                                                           08095000 T  0070:0014:0
                         THEN BEGIN                                                        08096000 T  0070:0014:0
                              BACKFIX ← L;                                                 08097000 T  0070:0014:1
                              IF FORMALV THEN CALL(DESC) END                               08098000 T  0070:0015:3
                         ELSE BACKFIX ← V + REAL(SIMPLEV)-1;                               08099000 T  0070:0017:2
                                                                                           08100000 T  0070:0019:2
                        AEXP;                                                              08101000 T  0070:0019:2
                  COMMENT PICK UP FIRST ARITHMETIC EXPRESSION;                             08102000 T  0070:0019:3
                        IF  ELCLASS = STEPV                                                08103000 T  0070:0019:3
                           THEN BEGIN                                                      08104000 T  0070:0020:0
                  COMMENT HERE WE HAVE A STEP ELEMENT;                                     08105000 T  0070:0021:2
                              BACKFIX ← BUMPL;                                             08106000 T  0070:0021:2
                  COMMENT LEAVE ROOM FOR FORWARD JUMP;                                     08107000 T  0070:0022:1
                              IF FORMALV THEN CALL(DESC); CALL(OPDC);                      08108000 T  0070:0022:1
                  COMMENT FETCH INDEX;                                                     08109000 T  0070:0025:2
                              IF I > 70 THEN BEGIN NXTELBT ← 1; I ← 0 END                  08110000 T  0070:0025:2
                                  ELSE REGO ← I;                                           08111000 T  0070:0027:3
                              IF SIMPLEB ← SIMPLE(CONSTANB,B,SIGNB) AND                    08112000 T  0070:0029:2
                               (ELCLASS = UNTILV OR ELCLASS = WHILEV)                      08113000 T  0070:0030:0
                                THEN BEGIN                                                 08114000 T  0070:0031:2
                  COMMENT WE HAVE A SIMPLE STEP FUNCTION;                                  08115000 T  0070:0033:2
                                PLUG(CONSTANB ,B);                                         08116000 T  0070:0033:2
                              END ELSE BEGIN                                               08117000 T  0070:0034:0
                  COMMENT THE STEP FUNCTION IS NOT SIMPLE;  WE CONSTRUCT A                 08118000 T  0070:0034:1
                              SUBROUTINE;                                                  08119000 T  0070:0034:1
                                  I ← IF I < 4 THEN 0 ELSE REGO; STEPIT;                   08120000 T  0070:0034:1
                                  SIGNB ← CONSTANB ← FALSE;                                08121000 T  0070:0038:0
                                  EMIT(0); B ← L;                                          08122000 T  0070:0039:2
                                  AEXP; EMITO(XCH);                                        08123000 T  0070:0040:1
                                  BRET ← L;                                                08124000 T  0070:0042:0
                                  EMITO(BFW) END;                                          08125000 T  0070:0042:1
                              EMITO(REAL(SIGNB)×32+ADD);                                   08126000 T  0070:0043:3
                              EMITB(BFW,BACKFIX,L);                                        08127000 T  0070:0045:2
                              IF ELCLASS = UNTILV                                          08128000 T  0070:0046:1
                                THEN BEGIN  COMMENT  STEP-UNTIL ELEMENT;                   08129000 T  0070:0046:1
                                   STORE(TRUE); IF FORMALV THEN CALL(OPDC);                08130000 T  0070:0047:3
                                   STEPIT; AEXP; TEST END                                  08131000 T  0070:0050:0
                                ELSE BEGIN  COMMENT  STEP-WHILE ELEMENT;                   08132000 T  0070:0051:3
                                   IF ELCLASS ≠ WHILEV THEN                                08133000 T  0070:0052:0
                                     BEGIN ERR(153); GO TO EXIT END;                       08134000 T  0070:0052:1
                                   STEPIT; STORE(FALSE); BEXP END END                      08135000 T  0070:0054:1
                           ELSE BEGIN                                                      08136000 T  0070:0056:0
                  COMMENT WE DO NOT HAVE A STEP ELEMENT;                                   08137000 T  0070:0056:1
                              STORE(FALSE);                                                08138000 T  0070:0056:1
                              IF ELCLASS = WHILEV                                          08139000 T  0070:0057:3
                                THEN BEGIN                                                 08140000 T  0070:0057:3
                  COMMENT WE HAVE A WHILE ELEMENT;                                         08141000 T  0070:0058:1
                                   STEPIT; BEXP END                                        08142000 T  0070:0058:1
                                ELSE BEGIN                                                 08143000 T  0070:0059:3
```

```
          COMMENT   ONE EXPRESSION ELEMENT;                                    08144000 T  0070:006:0
                        IF ELCLASS ≠ COMMA THEN BEGIN                          08145000 T  0070:006:0
                            EMITB(BFW,BUMPL,L+2); BACKFIX ← L END              08146000 T  0070:006:3
                         ELSE BACKFIX ← L + 2;                                 08147000 T  0070:006:2
                        L ← L+1; EMIT(BFW); GO TO BRNCH END END;               08148000 T  0070:006:1
          COMMENT   THIS IS THE COMMON POINT;                                  08149000 T  0070:006:2
                    IF ELCLASS = COMMA THEN EMITLNG; L ← L+1;                  08150000 T  0070:006:2
                    EMIT(BFC);                                                 08151000 T  0070:007:0
BRNCH;              FORWARDBRANCH ← L; DIALA ← DIALB ← 0;                      08152000 T  0070:007:2
                    IF ELCLASS = COMMA                                         08153000 T  0070:007:2
                      THEN BEGIN                                               08154000 T  0070:007:2
                         STEPIT;                                               08155000 T  0070:007:0
                         FORLIST(TRUE);                                        08156000 T  0070:007:1
                         FIX(STOREFIX,BACKFIX,FORWARDBRANCH,STMTSTART) END     08157000 T  0070:007:3
                      ELSE BEGIN                                               08158000 T  0070:007:2
                         IF ELCLASS ≠ DOV                                      08159000 T  0070:007:3
                             THEN BEGIN ERR(154); REGO←L; GO EXIT END;         08160000 T  0070:007:3
                         STEPIT;                                               08161000 T  0070:008:1
                         IF NUMLE THEN FOOT := GETSPACE(FALSE,-1); % TEMP.     08162000 T  0070:008:2
                    STMT;                                                      08163000 T  0070:008:3
                                                                               08164000 T  0070:008:0
                         IF NUMLE THEN BEGIN                                   08165000 T  0070:008:0
                           EMITV(RETURNSTORE ← FOOT); EMITO(BBW) END           08166000 T  0070:008:2
                         ELSE BEGIN                                            08167000 T  0070:008:2
                           EMITB(BBW,BUMPL,BACKFIX); RETURNSTORE ← 0 END;      08168000 T  0070:008:3
                         STMTSTART ← FORWARDBRANCH; B ← L;                     08169000 T  0070:009:1
                         CONSTANTCLEAN; REGO ← L;                              08170000 T  0070:009:0
                         FIX(STOREFIX,BACKFIX,FORWARDBRANCH,L) END;            08171000 T  0070:009:2
EXIT;               END FORLIST;                                               08172000 T  0070:009:1
```

70 IS 100 LONG, NEXT SEG 69

```
                    REAL T1,T2,T3,T4;                                          08173000 T  0069:008:1
STACK(F+2) = T1
STACK(F+3) = T2
STACK(F+4) = T3
STACK(F+5) = T4
                        NXTELBT ← 1; I ← 0;                                    08174000 T  0069:008:1
                        STEPIT;                                                08175000 T  0069:008:1
                        IF SIMPI(VRET←ELBAT[I])                                08176000 T  0069:008:2
                          THEN BEGIN                                           08177000 T  0069:008:3
                             IF STEPI ≠ ASSIGNOP THEN BEGIN ERR(152); GO EXIT END;  08178000 T  0069:008:2
                             XMARK(ASSIGNREF); % FOR STATEMENT                 %110-  08178100 C  0069:009:3
                             T1 ← L; IF FORMALV THEN EMITN(ADDRES);            08179000 T  0069:009:0
                             K ← 0;                                            08180000 T  0069:009:0
                             IF SIMPLE(CONSTANA,A,SIGNA) THEN                  08181000 T  0069:009:2
                             IF ELCLASS = STEPV THEN                           08182000 T  0069:010:0
                             IF SIMPLE(CONSTANB,B,SIGNB) THEN                  08183000 T  0069:010:3
                             IF ELCLASS = UNTILV THEN                          08184000 T  0069:010:2
                             IF SIMPLE(CONSTANC,Q,SIGNC) THEN                  08185000 T  0069:010:1
                             IF ELCLASS = DOV THEN                             08186000 T  0069:010:0
                               BEGIN                                           08187000 T  0069:010:3
                                  PLUG(CONSTANA,A);                            08188000 T  0069:010:0
                                  IF SIGNA THEN EMITO(CHS);                    08189000 T  0069:010:2
                                  RETURNSTORE  ← BUMPL; ADJUST; CONSTANTCLEAN; 08190000 T  0069:011:1
                                  STMTSTART ← L;                               08191000 T  0069:011:2
```

```
                STEP;T;
                T1 ← ((((4096 × RETURNSTORE+STMTSTART)×2+
                     REAL(CONSTANB))×2+
                     REAL(CONSTANC))×2+
                     REAL(SIGNB))×2+
                     REAL(SIGNC);
                T2 ← VRET;
                T3 ← B;
                T4 ← Q;
                STMT;
                SIGNC ← BOOLEAN(T1.[47:1]);
                SIGNB ← BOOLEAN(T1.[46:1]);
                CONSTANC ← BOOLEAN(T1.[45:1]);
                CONSTANB ← BOOLEAN(T1.[44:1]);
                STMTSTART ← T1.[32:12];
                RETURNSTORE ← T1.[20:12];
                VRET ← T2;
                B ← T3;
                Q ← T4;
                SIMPLEV← SIMPI(VRET);
                          IF FORMALV THEN EMITN(ADDRES); EMITV(ADDRES);
                          PLUG(CONSTANB,B);
                          EMITO(IF SIGNB THEN SUB ELSE ADD);
                          EMITB(BFW,RETURNSTORE,L);
                          STORE(TRUE);
                          IF FORMALV THEN CALL(OPDC);
                          PLUG(CONSTANC,Q);
                          IF SIGNC THEN EMITO(CHS);
                        SIMPLEB ← TRUE; TEST; EMITLNG;
                        EMITB(BBC,BUMPL,STMTSTART);
                        GO TO EXIT END;
                     I ← 2; K ← 0;
                     SIMPLEV ← SIMPI(VRET);
                     V ← T1 END
                  ELSE BEGIN
                     EMIT(0); V ← L; SIMPLEV ← FALSE; FORMALV ← TRUE;
                     VARIABLE(FR); EMITO(XCH); VRET ← L; EMITO(BFW);
                     IF ELCLASS≠ASSIGNOP THEN BEGIN ERR(152); GO EXIT END;
                     END;
                  STEPIT; FORLIST(FALSE); L ← REG0;
EXIT:  K ← 0 END FORSTMT;
```

```
08192000 T  0069:0114:0
08193000 T  0069:0114:1
08194000 T  0069:0116:0
08195000 T  0069:0117:2
08196000 T  0069:0118:0
08197000 T  0069:0119:2
08198000 T  0069:0120:0
08199000 T  0069:0121:2
08200000 T  0069:0121:3
08201000 T  0069:0122:1
08202000 T  0069:0123:2
08203000 T  0069:0124:0
08204000 T  0069:0125:3
08205000 T  0069:0126:1
08206000 T  0069:0128:0
08207000 T  0069:0129:2
08208000 T  0069:0130:1
08209000 T  0069:0131:2
08210000 T  0069:0132:0
08211000 T  0069:0132:1
08212000 T  0069:0134:0
08213000 T  0069:0136:0
08214000 T  0069:0137:2
08215000 T  0069:0139:3
08216000 T  0069:0140:1
08217000 T  0069:0141:3
08218000 T  0069:0143:2
08219000 T  0069:0144:0
08220000 T  0069:0145:3
08221000 T  0069:0147:2
08222000 T  0069:0149:3
08223000 T  0069:0151:2
08224000 T  0069:0152:1
08225000 T  0069:0153:3
08226000 T  0069:0154:1
08227000 T  0069:0155:2
08228000 T  0069:0158:0
08229000 T  0069:0161:2
08230000 T  0069:0163:3
08231000 T  0069:0163:3
08232000 T  0069:0165:3
69 IS  169 LONG, NEXT SEG    3
```

```
                REAL PROCEDURE REED;
PRT(655) = REED
                    BEGIN
                    LABEL EOF; INTEGER I,J,K;
STACK(F+3) = I
STACK(F+4) = J
STACK(F+5) = K
                    STREAM PROCEDURE MOVE(N,F,T); VALUE N,T;
PRT(656) = MOVE
                         BEGIN  SI:=F; DI:=T; DS:=N WDS  END MOVE;
```

```
08999000 T  0003:0505:3
08999025 T  0003:0505:3
08999050 T  0003:0505:3
START OF SEGMENT **********   71
08999075 T  0071:0000:0
08999100 T  0071:0000:0
```

```
                    J:=-1;
                    READ(CODISK[NO])[EOF];
PRT(657) = EOF
                    REED:=I:=FETCH(MKABS(CODISK(1)));
                    K:=MKABS(CODE(0))-1;
                    WHILE I:=J>30 DO
                         BEGIN
                         MOVE(30,CODISK(0),K); K:=K+30; J:=J+30;
                         READ(CODISK);
                         END;
                    MOVE(I:=J,CODISK(0),K);
                    READ(CODISK)[EOF];
               EOF:
               END REED;



               PROCEDURE RIGHT(L); VALUE L; INTEGER L;
                    BEGIN
                    INTEGER I,J;

STACK(F+2) = I
STACK(F+3) = J
                    I:=(L+7) DIV 4;
                    MOVE(1,I,CODISK(0));
                    MOVE(29,CODE(0),CODISK(1));
                    WRITE(CODISK);
                    J:=29;
                    WHILE I:=J>0 DO
                         BEGIN
                         MOVE(30,CODE(J),CODISK(0));
                         WRITE(CODISK);
                         J:=J+30;
                         END;
                    END RIGHT;



               COMMENT THE PROGRAM ROUTINE DOES THE INITIALIZATION AND THE WRAPUP
                    FOR THE REST OF THE COMPILER.  THE MAIN PROGRAM OF THE COMPILER
                    IS SIMPLY A CALL ON THE PROGRAM ROUTINE;
               PROCEDURE PROGRAM;
PRT(660) = PROGRAM
                    BEGIN
               STREAM PROCEDURE MDESC(WD,TOLOC);VALUE WD;
PRT(661) = MDESC
                 BEGIN DI←LOC WD; DS←  SET;SI← LOC WD; DI←TOLOC;DS←WDS END;
```

```
                                                  08999125 T  0071:000:2
                                                  08999150 T  0071:000:2
                                                  08999175 T  0071:000:0
                                                  08999200 T  0071:001:0
                                                  08999225 T  0071:001:2
                                                  08999250 T  0071:002:1
                                                  08999275 T  0071:002:1
                                                  08999300 T  0071:002:2
                                                  08999325 T  0071:003:2
                                                  08999350 T  0071:003:3
                                                  08999375 T  0071:003:0
                                                  08999400 T  0071:004:2
                                                  08999425 T  0071:004:0
                                            71 IS    47 LONG, NEXT SEG    3



                                                  08999450 T  0003:050:3
                                                  08999475 T  0003:050:3
                                                  08999500 T  0003:050:3
                                          START OF SEGMENT **********   72

                                                  08999525 T  0072:000:0
                                                  08999550 T  0072:000:3
                                                  08999575 T  0072:000:3
                                                  08999600 T  0072:001:1
                                                  08999625 T  0072:001:1
                                                  08999650 T  0072:001:2
                                                  08999675 T  0072:001:3
                                                  08999700 T  0072:001:3
                                                  08999725 T  0072:002:1
                                                  08999750 T  0072:003:1
                                                  08999775 T  0072:003:3
                                                  08999800 T  0072:003:0
                                            72 IS    35 LONG, NEXT SEG    3



                                                  09000000 T  0003:050:3
                                                  09001000 T  0003:050:3
                                                  09002000 T  0003:050:3
                                                  09003000 T  0003:050:3
                                                  09004000 T  0003:050:3
                                                  09005000 T  0003:050:3
                                          START OF SEGMENT **********   73

                                                  09006000 T  0073:000:0
```

```
                DEFINE START'INTRSC=426#;                                                                   09024000 T  0073'0001'3
                LABEL L1;                                                                                   09025000 T  0073'0001'3
                LISTOG'=LISTER'=BOOLEAN(1-ERRORCOUNT.[46:1]);                                               09028000 T  0073'0001'3
COMMENT  LISTOG IS NOT SET BY DEFAULT ON TIMESHARING;                                                       09028010 T  0073'0005'2
                NOHEADING '= TRUE;                                                                          09028050 T  0073'0005'2
                ERRORCOUNT '= 0;                                                                            09028900 T  0073'0006'0
                ERRMAX'=999; % MAY BE CHANGED IN DOLLARCARD.                                                09028910 T  0073'0006'1
                BASENUM'=10000; ADDVALUE'=1000; NEWBASE'=TRUE;                                              09028920 T  0073'0007'3
COMMENT  DEFAULT VALUES FOR "$SEQ" OPTION;                                                                  09028930 T  0073'0009'3
                LASTUSED '= 4; % FOR 'NITIALIZATION.                                                        09029000 T  0073'0009'3
                NEXT'NFO ← LAST'NFO ← LASTSEQROW×256+LASTSEQUENCE+1;                                        09033000 T  0073'0010'1
                PUTNBUMP(0);                                                                                09034000 T  0073'0013'2
                GT1 ← -"        ";                                                                          09034100 T  0073'0014'0
                MDESC(GT1,INFO[LASTSEQROW,LASTSEQUENCE]);                                                   09034200 T  0073'0015'2
                BLANKET(0,INFO[LASTSEQROW,LASTSEQUENCE]); % FOR "$ CHECK",                                  09034500 T  0073'0017'2
                READACARD; % INITIALIZATION OF NCR,FCR, AND LCR, AND                                        09035000 T  0073'0019'2
                           % READS FIRST CARD INTO CARD BUFFER.                                            09036000 T  0073'0019'3
                LASTUSED '= 1; % ASSUMES CARD ONLY UNTIL TOLD DIFFERENTLY.                                  09037000 T  0073'0019'3
                NXTELBT ← 1;                                                                                09038000 T  0073'0020'0
                PRTI←PRTIMAX←PRTBASE;                                                                       09039000 T  0073'0021'2
                MRCLEAN ← TRUE;                                                                             09040000 T  0073'0022'0
COMMENT  START FILLING TABLES NEEDED TO COMPILE A PROGRAM;                                                  09040100 T  0073'0023'2
                FILL TEN[*] WITH                                                                            09041000 T  0073'0023'2
   OCT'7711'10463422054,     OCT'761332600326467,     OCT'751621340414205,                                  09042000 T  0073'0023'3
                                                                                    START OF SEGMENT *********   74
   OCT'742165630517247,      OCT'732623176643120,     OCT'723370036413744,                                  09043000 T  0073'0024'1
   OCT'714266046116735,      OCT'705343457542525,     OCT'676634373473252,                                  09044000 T  0073'0024'1
   OCT'651040347241213,      OCT'641250441111455,     OCT'631522551333770,                                  09045000 T  0073'0024'1
   OCT'622047303622767,      OCT'612461164567564,     OCT'603175421725521,                                  09046000 T  0073'0024'1
   OCT'574034726313046,      OCT'565044113775657,     OCT'556255136775233,                                  09047000 T  0073'0024'1
   OCT'547730366574502,      OCT'521171646433362,     OCT'511430220142257,                                  09048000 T  0073'0024'1
   OCT'501736264172732,      OCT'472325741231521,     OCT'463013331500045,                                  09049000 T  0073'0024'1
   OCT'453616220020057,      OCT'444561664024072,     OCT'435716241031111,                                  09050000 T  0073'0024'1
   OCT'427301711237333,      OCT'401116227350722,     OCT'371341675243107,                                  09051000 T  0073'0024'1
   OCT'361632254513731,      OCT'352200727636717,     OCT'342641115606502,                                  09052000 T  0073'0024'1
   OCT'333411341150223,      OCT'324313631402270,     OCT'315376577702746,                                  09053000 T  0073'0024'1
   OCT'306676337663537,      OCT'261045602764047,     OCT'251257143561061,                                  09054000 T  0073'0024'1
   OCT'241532774515275,      OCT'232061573640554,     OCT'222476132610706,                                  09055000 T  0073'0024'1
   OCT'213215561353071,      OCT'204061115645707,     OCT'175075341217270,                                  09056000 T  0073'0024'1
   OCT'166314631463146,      OCT'141000000000000,     OCT'131200000000000,                                  09057000 T  0073'0024'1
   OCT'121440000000000,      OCT'111750000000000,     OCT'102342000000000,                                  09058000 T  0073'0024'1
   OCT'073032400000000,      OCT'063641100000000,     OCT'054611320000000,                                  09059000 T  0073'0024'1
   OCT'045753604000000,      OCT'037346545000000,     OCT'011124027620000,                                  09060000 T  0073'0024'1
    OCT'001351035564000,      OCT'001164324512100,      OCT'002221411634520,                                 09061000 T  0073'0024'1
    OCT'003265714203644,      OCT'004343277244615,      OCT'005434157115760,                                 09062000 T  0073'0024'1
    OCT'006543212741354,      OCT'007674055531647,      OCT'011105307106022,                                 09063000 T  0073'0024'1
   OCT'012126570727426,      OCT'013154327115334,     OCT'014207414740623,                                  09064000 T  0073'0024'1
   OCT'015251320130770,      OCT'016323604157166,     OCT'017410545213024,                                  09065000 T  0073'0024'1
   OCT'020512676455631,      OCT'021635456171177,     OCT'023100477162743,                                  09066000 T  0073'0024'1
   OCT'024120617017534,      OCT'025144762623464,     OCT'026176157370401,                                  09067000 T  0073'0024'1
   OCT'027235613266501,      OCT'030305156144221,     OCT'031366411575266,                                  09068000 T  0073'0024'1
   OCT'032464114134543,      OCT'033601137163674,     OCT'034741367020653,                                  09069000 T  0073'0024'1
   OCT'036113166462502,      OCT'037136024177223,     OCT'040165431237070,                                  09070000 T  0073'0024'1
   OCT'041222737506706,      OCT'042267527430470,     OCT'043345455336606,                                  09071000 T  0073'0024'1
   OCT'044436770626347,      OCT'045546566774041,     OCT'046700324573052,                                  09072000 T  0073'0024'1
```

```
    0CT0501060411731665,    0CT0511274514320242,    0CT0521553637404312,          09073000 T  0073:002#:1
    0CT0532106607305375,    0CT0542530351166674,    0CT0553256443424453,          09074000 T  0073:002#:1
    0CT0564132154331566,    0CT0575160607420123,    0CT0606414751324150,          09075000 T  0073:002#:1
    0CT0621012014361120,    0CT0631214417455344,    0CT0641457523370635,          09076000 T  0073:002#:1
    0CT0651773450267005,    0CT0662372362344606,    0CT0673071057035747,          09077000 T  0073:002#:1
    0CT0703707272645341,    0CT0714671151416632,    0CT0726047403722400,          09078000 T  0073:002#:1
    0CT0737461304707100,    0CT0751137556607072,    0CT0761367512350710,          09079000 T  0073:002#:1
    0CT0771665435043072;                                                          09080000 T  0073:002#:1
                                                                                74 IS  115 LONG, NEXT SEG   73
COMMENT  THIS IS THE FILL FOR THE SECOND ROW OF INFO:                             09081000 T  0073:002#:1
              THE FIRST ITEMS ARE STREAM RESERVED WORDS,                          09082000 T  0073:002#:1
              THEN ORDINARY RESERVED WORDS,                                       09083000 T  0073:002#:1
              THEN INTRINSIC FUNCTIONS:                                           09084000 T  0073:002#:1
 FILL INFO[1:*] WITH                                                              09085000 T  0073:002#:1
    0CT0670000600000002, "2SI000",                          %256                  09086000 T  0073:0025:3
                                                                              START OF SEGMENT **********   75
    0CT0700001040000002, "2DI000",                          %258                  09087000 T  0073:0026:1
    0CT0710001460000002, "2CI000",                          %260                  09088000 T  0073:0026:1
    0CT0720001630000002, "5TALLY",                          %262                  09089000 T  0073:0026:1
    0CT0730000530000002, "2DS000",                          %264                  09090000 T  0073:0026:1
    0CT0740000150000002, "4SKIP0",                          %266                  09091000 T  0073:0026:1
    0CT0750001620000002, "4JUMP0",                          %268                  09092000 T  0073:0026:1
    0CT0760000740000002, "2DB000",                          %270                  09093000 T  0073:0026:1
    0CT0770000500000002, "2SB000",                          %272                  09094000 T  0073:0026:1
    0CT1010000730000002, "2SC000",                          %274                  09095000 T  0073:0026:1
    0CT1020001160000002, "3LOC00",                          %276                  09096000 T  0073:0026:1
    0CT1030001170000002, "2DC000",                          %278                  09097000 T  0073:0026:1
    0CT1040001430000002, "5LOCAL",                          %280                  09098000 T  0073:0026:1
    0CT1050000340000002, "3LIT00",                          %282                  09099000 T  0073:0026:1
    0CT1060001036400002, "3SET00",                          %284                  09100000 T  0073:0026:1
    0CT1060001066500002, "5RESET",                          %286                  09101000 T  0073:0026:1
    0CT1060001020500002, "3WDS00",                          %288                  09102000 T  0073:0026:1
    0CT1060001357700002, "3CHR00",                          %290                  09103000 T  0073:0026:1
    0CT1060001057300002, "3ADD00",                          %292                  09104000 T  0073:0026:1
    0CT1060001617200002, "3SUB00",                          %294                  09105000 T  0073:0026:1
    0CT1060000727600002, "3ZON00",                          %296                  09106000 T  0073:0026:1
    0CT1060000417500002, "3NUM00",                          %298                  09107000 T  0073:0026:1
    0CT1060000766700002, "3OCT00",                          %300                  09108000 T  0073:0026:1
    0CT1060000176600002, "3DEC00",                          %302                  09109000 T  0073:0026:1
    0CT1004000260000003, "6TOGGL", "E0000000",              %304                  09110000 T  0073:0026:1
    0CT0130311060000002, "3ABS00",                          %307                  09110001 T  0073:0026:1
    0CT1360441030000002, "3AND00",                          %309                  09112000 T  0073:0026:1
    0CT0500000170000002, "5ARRAY",                          %311                  09112100 T  0073:0026:1
    0CT0660000000000002, "5BEGIN",                          %313                  09112200 T  0073:0026:1
    0CT0500000040000003, "7BOOLF", "AN000000",              %315                  09112300 T  0073:0026:1
    0CT1070000000000003, "7COMME", "NT000000",              %318                  09112400 T  0073:0026:1
    0CT0500000230000003, "6DEFIN", "E0000000",              %321                  09112500 T  0073:0026:1
    0CT1410446000000002, "3DIV00",                          %324                  09112600 T  0073:0026:1
    0CT0550000000000002, "2D0000",                          %326                  09112700 T  0073:0026:1
    0CT0520000000000003, "6DOUBL", "E0000000",              %328                  09112800 T  0073:0026:1
    0CT0570000000000002, "4ELSE0",                          %331                  09112900 T  0073:0026:1
    0CT0600000000000002, "3END00",                          %333                  09113000 T  0073:0026:1
    0CT1340442030000002, "3EQV00",                          %335                  09113100 T  0073:0026:1
    0CT0410000000000002, "5FALSE",                          %337                  09113200 T  0073:0026:1
    0CT0130310030000002, "4FLAG0",                          %339                  09113300 T  0073:0026:1
    0CT0530000000000002, "3FOR00",                          %341                  09113400 T  0073:0026:1
    0CT1100000000000003, "7FORWA", "RD      ",              %343                  09113500 T  0073:0026:1
    0CT0640000000000002, "2G0000",                          %346                  09113600 T  0073:0026:1
```

```
0CT0130316060320002, "4HUNTO",
0CT0630000000000002, "2IFOOO",
0CT0500000040000002, "4REALO",
0CT0500000050000003, "7INTEG", "ERO000000",
0CT0500000070000002, "5LABEL",
0CT0360002000020003, "6MEMOR", "Y      ",
0CT1410456000000002, "3MODOO",
0CT0500000140000003, "7MONIT", "OR     ",
0CT0130301060000002, "4NARSO",
0CT0500000200000002, "4NAMEO",
0CT0130304030000002, "5NFLAG",
0CT1320300230000002, "3NOTOO",
0CT1350440430000002, "2ORGOO",
0CT0500000020000002, "4SAVEO",
0CT0500000010000002, "3OWNOO",
0CT0460000000000003, "6POLIS", "H      ",
0CT0500000160000003, "9PROCE", "DURE   ",
0CT0130300000160011, "4SIGNO",
          0CT2025,   COMMENT  DUP   ;
          0CT0000,   COMMENT  LITC 0;
          0CT0425,   COMMENT  NEQ   ;
          0CT1025,   COMMENT  XCH   ;
          0CT0155,   COMMENT  DIA  1;
          0CT0161,   COMMENT  DIB  1;
          0CT0165,   COMMENT  TRB  1;
0CT1110000000000002, "4STEPO",
0CT0500000220000003, "6STREA", "M      ",
0CT0500000110000003, "#SUBRO", "UTINE  ",
0CT0500000150000003, "6SWITC", "H      ",
0CT1120000000000002, "4THENO",
0CT1130000000000002, "2TOOOO",
0CT0410000010000002, "4TRUEO",
0CT0560000000000002, "5UNTIL",
0CT1140000000000002, "5VALUE",
0CT0540000000000002, "5WHILE",
0CT1310440200000002, "3ADDOO",
0CT1310240270000002, "3BRTOO",
0CT1310453050000002, "3CCXOO",
0CT1310442500000002, "3CDCOO",
0CT1310457050000002, "3CFXOO",
0CT1310302060000002, "3CHSOO",
0CT1310440500000002, "3COCOO",
0CT1310242020000002, "3COMOO",
0CT1310302060000002, "3CSBOO",
0CT1310240120000002, "3DELOO",
0CT1260100550000002, "3DIAOO",
0CT1260100610000002, "3DIBOO",
0CT1310344050000002, "3DUPOO",
0CT1310451050000002, "3EQLOO",
0CT1310443050000002, "3FCXOO",
0CT1310447050000002, "3FFXOO",
0CT1310440250000002, "3GEQOO",
0CT1310440450000002, "3GTROO",
0CT1310104420000002, "3HLBOO",
0CT1310104420000002, "3HP2OO",
0CT1310446000000002, "3IDVOO",
0CT1310251020000002, "3IIOOO",
```

```
%348    09113700 T  0073:0026:1
%350    09113800 T  0073:0026:1
%352    09113900 T  0073:0026:1
%354    09114000 T  0073:0026:1
%357    09114100 T  0073:0026:1
%359    09114200 T  0073:0026:1
%362    09114300 T  0073:0026:1
%364    09114400 T  0073:0026:1
%367    09114500 T  0073:0026:1
%369    09114600 T  0073:0026:1
%371    09114700 T  0073:0026:1
%373    09114800 T  0073:0026:1
%375    09114900 T  0073:0026:1
%377    09115000 T  0073:0026:1
%379    09115100 T  0073:0026:1
%381    09115200 T  0073:0026:1
%384    09115300 T  0073:0026:1
%387    09115400 T  0073:0026:1
        09115500 T  0073:0026:1
        09115600 T  0073:0026:1
        09115700 T  0073:0026:1
        09115800 T  0073:0026:1
        09115900 T  0073:0026:1
        09116000 T  0073:0026:1
        09116100 T  0073:0026:1
%396    09116200 T  0073:0026:1
%398    09116300 T  0073:0026:1
%401    09116400 T  0073:0026:1
%404    09116500 T  0073:0026:1
%407    09116600 T  0073:0026:1
%409    09116700 T  0073:0026:1
%411    09116800 T  0073:0026:1
%413    09116900 T  0073:0026:1
%415    09117000 T  0073:0026:1
%417    09117100 T  0073:0026:1
%419    09117200 T  0073:0026:1
%421    09117300 T  0073:0026:1
%423    09117400 T  0073:0026:1
%425    09117500 T  0073:0026:1
%427    09117600 T  0073:0026:1
%429    09117700 T  0073:0026:1
%431    09117800 T  0073:0026:1
%433    09117900 T  0073:0026:1
%435    09118000 T  0073:0026:1
%437    09118100 T  0073:0026:1
%439    09118200 T  0073:0026:1
%441    09118300 T  0073:0026:1
%443    09118400 T  0073:0026:1
%445    09118500 T  0073:0026:1
%447    09118600 T  0073:0026:1
%449    09118700 T  0073:0026:1
%451    09118800 T  0073:0026:1
%453    09118900 T  0073:0026:1
%455    09119000 T  0073:0026:1
%457    09119050 T  0073:0026:1
%459    09119100 T  0073:0026:1
%461    09119200 T  0073:0026:1
```

```
                0CT1310250220000002, "3INA00",                          %463            09119300 T  0073:0026:1
                0CT1310250420000002, "3INB00",                          %465            09119400 T  0073:0026:1
                0CT1310100420000002, "3INI00",                          %467            09119500 T  0073:0026:1
                0CT1400440300000002, "3INX00",                          %469            09119600 T  0073:0026:1
                0CT1310244220000002, "3IOR00",                          %471            09119700 T  0073:0026:1
                0CT1310250220000002, "3IP100",                          %473            09119800 T  0073:0026:1
                0CT1310250420000002, "3IP200",                          %475            09119900 T  0073:0026:1
                0CT1310145060000002, "3IPS00",                          %477            09120000 T  0073:0026:1
                0CT1310410240000002, "3ISD00",                          %479            09120100 T  0073:0026:1
                0CT1310450440000002, "3ISN00",                          %481            09120200 T  0073:0026:1
                0CT1310100420000002, "3IT100",                          %483            09120300 T  0073:0026:1
                0CT1310450250000002, "3LEQ00",                          %485            09120400 T  0073:0026:1
                0CT1310505300000002, "3LLL00",                          %487            09120500 T  0073:0026:1
                0CT1310441030000002, "3LND00",                          %489            09120600 T  0073:0026:1
                0CT1310300230000002, "3LNG00",                          %491            09120700 T  0073:0026:1
                0CT1310304040000002, "3LOD00",                          %493            09120800 T  0073:0026:1
                0CT1310440430000002, "3LOR00",                          %495            09120900 T  0073:0026:1
                0CT1310442030000002, "3LQV00",                          %497            09121000 T  0073:0026:1
                0CT1310450450000002, "3LSS00",                          %499            09121100 T  0073:0026:1
                0CT1310101100000002, "3MKS00",                          %501            09121200 T  0073:0026:1
                0CT1310441000000002, "3MUL00",                          %503            09121300 T  0073:0026:1
                0CT1310441050000002, "3NEQ00",                          %505            09121400 T  0073:0026:1
                0CT1310100130000002, "3NOP00",                          %507            09121500 T  0073:0026:1
                0CT0650006550000003, "6SCOP0", "N.......";              %509            09121600 T  0073:0026:1
                                                                                  75 IS  256 LONG, NEXT SEG   73
            FILL INFO[2,*] WITH                                                       09121650 T  0073:0026:1
                0CT1310300000020004, "3RDF00",                          %512            09121700 T  0073:0027:3
                                                                                  START OF SEGMENT **********  76
                        0CT0000,   COMMENT  LITC 0;                               09121800 T  0073:0028:1
                        0CT2141,   COMMENT  FXS  ;                                09121900 T  0073:0028:1
                0CT1310300000020004, "3RDS00",                          %516            09122000 T  0073:0028:1
                        0CT0004,   COMMENT  LITC 1;                               09122100 T  0073:0028:1
                        0CT2141,   COMMENT  FXS  ;                                09122200 T  0073:0028:1
                0CT1310456000000002, "3RDV00",                          %520            09122300 T  0073:0028:1
                0CT1310304030000002, "3RFB00",                          %522            09122400 T  0073:0028:1
                0CT1310240470000002, "3RN000",                          %524            09122500 T  0073:0028:1
                0CT1310145060000002, "3RRB00",                          %526            09122600 T  0073:0028:1
                0CT1310311060000002, "3RSB00",                          %528            09122700 T  0073:0028:1
                0CT1310242470000002, "3RSP00",                          %530            09122800 T  0073:0028:1
                0CT1310141020000002, "3RTM00",                          %532            09122900 T  0073:0028:1
                0CT1310240470000002, "3RTN00",                          %534            09123000 T  0073:0028:1
                0CT1310141020000002, "3RTR00",                          %536            09123100 T  0073:0028:1
                0CT1310242470000002, "3RTS00",                          %538            09123200 T  0073:0028:1
                0CT1310310030000002, "3SFB00",                          %540            09123300 T  0073:0028:1
                0CT1310442040000002, "3SND00",                          %542            09123400 T  0073:0028:1
                0CT1310301060000002, "3SSB00",                          %544            09123500 T  0073:0028:1
                0CT1310316060000002, "3SSF00",                          %546            09123600 T  0073:0028:1
                0CT1310301060000002, "3SSN00",                          %548            09123700 T  0073:0028:1
                0CT1310311060000002, "3SSP00",                          %550            09123800 T  0073:0028:1
                0CT1310401040000002, "3STD00",                          %552            09123900 T  0073:0028:1
                0CT1310240000020004, "3STF00",                          %554            09124000 T  0073:0028:1
                        0CT0010,   COMMENT  LITC 2;                               09124100 T  0073:0028:1
                        0CT2141,   COMMENT  FXS  ;                                09124200 T  0073:0028:1
                0CT1310442040000002, "3STN00",                          %558            09124300 T  0073:0028:1
                0CT1310240000020004, "3STS00",                          %560            09124400 T  0073:0028:1
                        0CT0014,   COMMENT LITC 3;                                09124500 T  0073:0028:1
                        0CT2141,   COMMENT FXS  ;                                 09124600 T  0073:0028:1
```

```
0CT;3104406000000002, "3SUB00",                %564          09124700 T  0073:0028:1
0CT;3103440600000002, "3TFB00",                %566          09124800 T  0073:0028:1
0CT;2704406500000002, "3TFR00",                %568          09124900 T  0073:0028:1
0CT;3101555060000002, "3TI000",                %570          09125000 T  0073:0028:1
0CT;3103440600000002, "3TOP00",                %572          09125050 T  0073:0028:1
0CT;2704406500000002, "3TRB00",                %574          09125100 T  0073:0028:1
0CT;3003000000000002, "3VFI00",                %576          09125200 T  0073:0028:1
0CT;3105020500000002, "3XCH00",                %578          09125300 T  0073:0028:1
0CT;3101010700000002, "3XIT00",                %580          09125400 T  0073:0028:1
0CT;3101050200000002, "3ZIP00",                %582          09125500 T  0073:0028:1
0CT;3101050200000002, "3ZP100",                %584          09125600 T  0073:0028:1
0CT;2705007500000002, "3CFF00",                %586          09125700 T  0073:0028:1
0CT;2705007500000002, "3FCE00",                %588          09125800 T  0073:0028:1
0CT;2705007100000002, "3CFL00",                %590          09125900 T  0073:0028:1
0CT;2705007100000002, "3FCL00",                %592          09126000 T  0073:0028:1
0CT;3104402100000002, "3DLA00",                %594          09126100 T  0073:0028:1
0CT;3104402100000002, "3ADL00",                %596          09126200 T  0073:0028:1
0CT;3104406100000002, "3DLS00",                %598          09126300 T  0073:0028:1
0CT;3104406100000002, "3SDL00",                %600          09126400 T  0073:0028:1
0CT;3104410100000002, "3DLM00",                %602          09126500 T  0073:0028:1
0CT;3104410100000002, "3MDL00",                %604          09126600 T  0073:0028:1
0CT;3104420100000002, "3DLD00",                %606          09126700 T  0073:0028:1
0CT;3104420100000002, "3DDL00",                %608          09126800 T  0073:0028:1
0CT;0460000000000002, "1P0000",                %610          09126900 T  0073:0028:1
0CT;0360002000020002, "1M0000",                %612          09127000 T  0073:0028:1
0CT;3102400000020004, "3PRL00",                %614          09127100 T  0073:0028:1
          0CT;0111,   COMMENT  PRL;                          09127200 T  0073:0028:1
          0CT;0055,   COMMENT  NOP;                          09127300 T  0073:0028:1
0CT;0650006610000003, "7SCOPO", "FF......",    %618          09127400 T  0073:0028:1
0CT;0030000000040003, "2LBOOO", "[#      ",    %621          09127500 T  0073:0028:1
0CT;0030000000040003, "2RBOOO", "]#      ",    %624          09127600 T  0073:0028:1
0CT;0030000000040003, "3GTR00", ">#      ",    %627          09127700 T  0073:0028:1
0CT;0030000000040003, "3GEQ00", "≥#      ",    %630          09127800 T  0073:0028:1
0CT;0030000000040003, "3EQL00", "=#      ",    %633          09127900 T  0073:0028:1
0CT;0030000000040003, "3NEQ00", "#       ",    %636          09128000 T  0073:0028:1
0CT;0030000000040003, "3LEQ00", "≤#      ",    %639          09128100 T  0073:0028:1
0CT;0030000000040003, "3LSS00", "<#      ",    %642          09128200 T  0073:0028:1
0CT;0030000000040003, "5TIMES", "×#      ",    %645          09128300 T  0073:0028:1
0CT;1310117530000002, "3SCI00",                %648          09128400 T  0073:0028:1
0CT;1310117540000002, "3SAN00",                %650          09128500 T  0073:0028:1
0CT;1310157730000002, "3SCS00",                 %652 %112-   09128600 P  0073:0028:1
0CT;0500000250000002, "5FIELD",                 %654 %115-   09128700 P  0073:0028:1
0CT;0610000000000000, "4CASE0",                 %656 %115-   09128800 P  0073:0028:1
                                                             09128900 T  0073:0028:1
                                                             09129000 T  0073:0028:1
                                                             09129100 T  0073:0028:1
                                                             09129200 T  0073:0028:1
                                                             09129300 T  0073:0028:1
                                                             09129400 T  0073:0028:1
                                                             09129500 T  0073:0028:1
                                                             09129600 T  0073:0028:1
                                                             09129700 T  0073:0028:1
                                                             09129800 T  0073:0028:1
                                                             09129900 T  0073:0028:1
                                                             09130000 T  0073:0028:1
                                                             09130100 T  0073:0028:1
                                                             09130200 T  0073:0028:1
```

```
0;    % END OF INFO FILL.

        FOR GT2←256 STEP GT1.LINK WHILE NOT BOOLEAN(GT1,FORMAL) DO
             PUT((GT1←TAKE(GT2))&GT2[35:35:13],GT2);
        FOR GT1←GT2 STEP GT2.LINK WHILE GT2.LINK≠0 DO
             PUT((GT2←TAKE(GT1))&STACKHEAD[GT3←TAKE(GT1+1),[12:36]
                  MOD 125][35:35:13],STACKHEAD[GT3]←GT1);
COMMENT THIS IS THE FILL FOR THE SPECIAL CHARACTORS;
FILL SPECIAL[*] WITH
 OCT1200000000200000,  COMMENT  #;   OCT0000000000100000,  COMMENT  @;
 OCT0000000000000000,                OCT1160000000120000,  COMMENT  !;
 OCT1370440450002763,  COMMENT  >;   OCT1370440250002662,  COMMENT  ≥;
 OCT1400440200000000,  COMMENT  +;   OCT0000000000000000,
 OCT1220000000060000,  COMMENT  .;   OCT1210000000000000,  COMMENT  [;
 OCT1250000000000000,  COMMENT  &;   OCT0450000000000000,  COMMENT  (;
 OCT1370450450003571,  COMMENT  <;   OCT1330401040000000,  COMMENT  ←;
 OCT1410441000000000,  COMMENT  ×;   OCT0000000000000000,
 OCT0000000000040000,  COMMENT  $;   OCT0470000000000000,  COMMENT  *;
 OCT1400440600000000,  COMMENT  -;   OCT1240000000160000,  COMMENT  );
 OCT0620000000000000,  COMMENT  ,;   OCT1370450250003470,  COMMENT  ≤;
 OCT0000000000000000,                OCT1410442000000000,  COMMENT  /;
```

```
09130300 T  0073:0028:1
09130400 T  0073:0028:1
09130500 T  0073:0028:1
09130600 T  0073:0028:1
09130700 T  0073:0028:1
09130800 T  0073:0028:1
09130900 T  0073:0028:1
09131000 T  0073:0028:1
09131100 T  0073:0028:1
09131200 T  0073:0028:1
09131300 T  0073:0028:1
09131400 T  0073:0028:1
09131500 T  0073:0028:1
09131600 T  0073:0028:1
09131700 T  0073:0028:1
09131800 T  0073:0028:1
09131900 T  0073:0028:1
09132000 T  0073:0028:1
09132100 T  0073:0028:1
09132200 T  0073:0028:1
09132300 T  0073:0028:1
09132400 T  0073:0028:1
09132500 T  0073:0028:1
09132600 T  0073:0028:1
09132700 T  0073:0028:1
09132800 T  0073:0028:1
09132900 T  0073:0028:1
09133000 T  0073:0028:1
09133100 T  0073:0028:1
09133200 T  0073:0028:1
09133300 T  0073:0028:1
09133400 T  0073:0028:1
09133450 T  0073:0028:1
09133500 T  0073:0028:1
09133600 T  0073:0028:1
09133700 T  0073:0028:1
   76 IS  147 LONG, NEXT SEG   73
09133800 T  0073:0028:1
09133900 T  0073:0034:0
09134000 T  0073:0040:0
09134100 T  0073:0045:3
09134200 T  0073:0049:2
09197000 T  0073:0053:2
09198000 T  0073:0053:2
09199000 T  0073:0053:3
START OF SEGMENT **********   77
09200000 T  0073:0054:1
09201000 T  0073:0054:1
09202000 T  0073:0054:1
09203000 T  0073:0054:1
09204000 T  0073:0054:1
09205000 T  0073:0054:1
09206000 T  0073:0054:1
09207000 T  0073:0054:1
09208000 T  0073:0054:1
09209000 T  0073:0054:1
09210000 T  0073:0054:1
```

```
OCT1170000000000000,  COMMENT  ,;   OCT0000000000020000,  COMMENT  %;          09211000 T  0073!0054!1
OCT1370441050002561,  COMMENT  #;   OCT1370451050002460,  COMMENT  =;          09212000 T  0073!0054!1
OCT1230000000000000,  COMMENT  ];   OCT0000000000140000,  COMMENT  ";          09213000 T  0073!0054!1
 0,0;                                                                          09214000 T  0073!0054!1
                                                                        77 IS   32 LONG, NEXT SEG   73
  FILL MACRO[*] WITH                                                          09215000 T  0073!0054!1
       OCT0131,          COMMENT SFS A       00          ;                    09216000 T  0073!0055!2
                                                                      START OF SEGMENT *********   78
       OCT0116,          COMMENT SFD A       01          ;                    09217000 T  0073!0056!1
       OCT0000,          COMMENT SYNTAX ERROR02          ;                    09218000 T  0073!0056!1
       OCT0140,          COMMENT INC A       03          ;                    09219000 T  0073!0056!1
       OCT0130,          COMMENT SRS A       04          ;                    09220000 T  0073!0056!1
       OCT0117,          COMMENT SRD A       05          ;                    09221000 T  0073!0056!1
       OCT0000,          COMMENT SYNTAX ERROR06          ;                    09222000 T  0073!0056!1
       OCT0000,          COMMENT SYNTAX ERROR07          ;                    09223000 T  0073!0056!1
       OCT00310143,      COMMENT CRF A, SFS 008          ;                    09224000 T  0073!0056!1
       OCT00160143,      COMMENT CRF A, SFD 009          ;                    09225000 T  0073!0056!1
       OCT00470143,        COMMENT CRF A,  JFW 0  10       ;                  09226000 T  0073!0056!1
       OCT00400143,      COMMENT CRF A, INC 011          ;                    09227000 T  0073!0056!1
       OCT00300143,      COMMENT CRF A, SRS 012          ;                    09228000 T  0073!0056!1
       OCT00170143,      COMMENT CRF A, SRD 013          ;                    09229000 T  0073!0056!1
       OCT0000,          COMMENT SYNTAX ERROR14          ;                    09230000 T  0073!0056!1
       OCT0000,          COMMENT SYNTAX ERROR15          ;                    09231000 T  0073!0056!1
       OCT0153,          COMMENT RSA A       16          ;                    09232000 T  0073!0056!1
       OCT0104,          COMMENT RDA A       17          ;                    09233000 T  0073!0056!1
       OCT0150,          COMMENT RCA A       18          ;                    09234000 T  0073!0056!1
  OCT0042014300 42,      COMMENT SEC 0, CRF A, SEC 0 19  ;                    09235000 T  0073!0056!1
       OCT0122,          COMMENT SES A       20          ;                    09236000 T  0073!0056!1
       OCT0106,          COMMENT SED A       21          ;                    09237000 T  0073!0056!1
       OCT0000,          COMMENT SYNTAX ERROR22          ;                    09238000 T  0073!0056!1
       OCT0000,          COMMENT SYNTAX ERROR23          ;                    09239000 T  0073!0056!1
       OCT0056,          COMMENT TSA 0       24          ;                    09240000 T  0073!0056!1
       OCT0000,          COMMENT SYNTAX ERROR25          ;                    09241000 T  0073!0056!1
       OCT0000,          COMMENT SYNTAX ERROR26          ;                    09242000 T  0073!0056!1
       OCT0000,          COMMENT SYNTAX ERROR27          ;                    09243000 T  0073!0056!1
       OCT0000,          COMMENT SYNTAX ERROR28          ;                    09244000 T  0073!0056!1
       OCT0007,          COMMENT TDA 0       29          ;                    09245000 T  0073!0056!1
       OCT0000,          COMMENT SYNTAX ERROR30          ;                    09246000 T  0073!0056!1
       OCT0000,          COMMENT SYNTAX ERROR31          ;                    09247000 T  0073!0056!1
       OCT0115,          COMMENT SSA A       32          ;                    09248000 T  0073!0056!1
       OCT0114,          COMMENT SDA A       33          ;                    09249000 T  0073!0056!1
       OCT0154,          COMMENT SCA A       34          ;                    09250000 T  0073!0056!1
       OCT0141;          COMMENT STC A       35          ;                    09251000 T  0073!0056!1
                                                                        78 IS   36 LONG, NEXT SEG   73
  FILL OPTIONS[*] WITH "5CHECK",0,          % 0,1                             09251208 T  0073!0056!1
                                                                      START OF SEGMENT *********   79
                       "6DEBUG",0,          % 2,3                             09251212 T  0073!0058!0
                       "4DECK0",0,          % 4,5                             09251214 T  0073!0058!0
                       "6FORMA",0,          % 6,7                             09251216 T  0073!0058!0
                       "9INTRI",0,          % 8,9                             09251218 T  0073!0058!0
                       "5LISTA",0,          % 10,11                           09251220 T  0073!0058!0
                       "4LISTO",0,          % 12,13                           09251224 T  0073!0058!0
                       "5LISTP",0,          % 14,15                           09251228 T  0073!0058!0
                       "3MCPOO",0,          % 16,17                           09251230 T  0073!0058!0
                       "4TAPEO",0,          % 18,19                           09251232 T  0073!0058!0
                       "4NESTO",0,          % 20,21                           09251234 T  0073!0058!0
                       "3NEWOO",0,          % 22,23                           09251236 T  0073!0058!0
```

```
                "7NEWIN",0,          % 24,25                                    09251240 T  0073:0058:0
                "40MITO",0,          % 26,27                                    09251244 T  0073:0058:0
                "1$0000",0,          % 28,29                                    09251248 T  0073:0058:0
                "3PRTOO",0,          % 30,31                                    09251252 T  0073:0058:0
                "5PUNCH",0,          % 32,33                                    09251256 T  0073:0058:0
                "5PURGE",0,          % 34,35                                    09251260 T  0073:0058:0
                "4SEGSO",0,          % 36,37                                    09251264 T  0073:0058:0
                "3SEQOO",0,          % 38,39                                    09251268 T  0073:0058:0
                "6SEQER",0,          % 40,41                                    09251272 T  0073:0058:0
                "6SINGL",0,          % 42,43                                    09251276 T  0073:0058:0
                "5STUFF",0,          % 44,45                                    09251278 T  0073:0058:0
                "4VOIDO",0,          % 46,47                                    09251280 T  0073:0058:0
                "5VOIDT",0,          % 48,49                                    09251284 T  0073:0058:0
                "4BENDO",0,          % 50, 51          %108-                    09251285 C  0073:0058:0
                "4XREFO",0,          % 52, 53          %108-                    09251286 C  0073:0058:0
0;                                                                              09251288 T  0073:0058:0
                                                                        79 IS   55 LONG, NEXT SEG   73
 DO UNTIL STEP1 = BEGINV;                                                       09252000 T  0073:0058:0
 GT1 ←"       ";                                                                09253000 T  0073:0060:1
 INTOG ← INTOG AND TRUE; %                                                      09253050 T  0073:0061:3
 DISKADR ← IF INTOG THEN INTRINSICADR ELSE 2;                                   09253100 T  0073:0064:0
 MDESC(GT1,INFO[LASTSEQROW,LASTSEQUENCE]);                                      09253500 T  0073:0067:3
 MDESC(GT1,INFO[LASTSEQROW,LASTSEQUENCE-1]);                                    09254000 T  0073:0069:3
 MDESC(GT1,INFO[LASTSEQROW,LASTSEQUENCE-2]);                                    09255000 T  0073:0072:0
STMT;                                                                           09275000 T  0073:0074:1
 LOCK(STUFF);                                                                   09281000 T  0073:0075:2
 CLOSE(CARD,RELEASE);                                                           09281500 T  0073:0076:1
 IF LASTUSED ≠ 1 THEN CLOSE(TAPE,RELEASE);                                      09282000 T  0073:0078:1
 IF NEWTOG THEN LOCK(NEWTAPE,*);                                                09282500 T  0073:0081:3
 IF T←((L+3)DIV 4) + CORADR > 4080 THEN FLAG(040);                              09282600 T  0073:0084:1
 IF NOT NOHEADING THEN % PRINT THESE THINGS IF ANY                              09362000 T  0073:0088:1
                  BEGIN              % LISTING HAS BEEN DONE.                   09363000 T  0073:0089:2
          STREAM PROCEDURE PAN(T,FIEL,NER,LSQ); VALUE NER,T;                    09364000 T  0073:0089:3
PRT(662) = *SEGMENT DESCRIPTOR*
                                                                  START OF SEGMENT **********   80
PRT(663) = PAN
                BEGIN DI ← FIEL; 44(DS←2LIT" ");                                09365000 T  0080:0000:0
                SI ← LSQ; DS ← WDS; SI ←FIEL; DS ← 3 WDS;                       09366000 T  0080:0001:2
                DI ← FIEL; DS← 28 LIT"NUMBER OF ERRORS DETECTED = ";            09367000 T  0080:0002:0
                SI ← LOC NER; DS ← 3 DEC; DS ← 22 LIT ".  COMPILATION TIME = ";  09368000 T  0080:0006:0
                SI ← LOC T; DS ← 4 DEC; DS ← 9 LIT " SECONDS."; END;            09369000 T  0080:0009:3


          STREAM PROCEDURE PEN(FIL,PRTSIZ,BASE,CORE,DISK);                      09370000 T  0080:0012:0
PRT(664) = PEN
                    VALUE PRTSIZ,BASE,CORE,DISK;                                09371000 T  0080:0012:0
                BEGIN DI←FIL; DS ← 9 LIT"PRT SIZE="; SI←LOC PRTSIZ;             09372000 T  0080:0012:0
                      DS ← 3 DEC; DS←14 LIT" BASE ADDRESS=";                    09373000 T  0080:0014:0
                      SI←LOC BASE; DS←4 DEC; DS←10 LIT" CORE REQ=";             09374000 T  0080:0016:0
                      SI←LOC CORE; DS←4 DEC; DS←10 LIT" DISK REQ=";             09375000 T  0080:0018:0
                      SI←LOC DISK; DS←5 DEC; DS←61 LIT " ";                     09376000 T  0080:0020:0
                END  PEN;                                                       09377000 T  0080:0028:1


          STREAM PROCEDURE FINALAX(LINE,N,SEQ); VALUE N;                        09378000 T  0080:0029:2
```

```
PRT(665) = FINALAX
                    BEGIN DI ← LINE; 15(DS ← 8 LIT " ");                                    09379000 T  0080:0029:2
                         DI ← LINE; DS ← 31 LIT "NUMBER OF ACCIDENTAL ENTRIES = ";          09380000 T  0080:0031:2
                         SI ← LOC N; DS ← 3 DEC; DI ← DI+8;                                 09381000 T  0080:0035:3
                         SI ← SEQ; SI ← SI-16; DS ← 8 CHR;                                  09382000 T  0080:0036:0
                    END;                                                                    09383000 T  0080:0037:2



                    IF AXNUM ≠0 THEN                                                        09384000 T  0080:0037:2
                         BEGIN                                                              09384050 T  0080:0038:1
                         FINALAX(LIN[0],AXNUM,INFO[LASTSEQROW,LASTSEQUENCE]);               09384100 T  0080:0039:2
                         WRITELINE;                                                         09384500 T  0080:0041:3
                         END;                                                               09384600 T  0080:0052:0
                    SCRAM := (TIME(1)-TIME1)/60;                                            09385000 T  0080:0052:0
                    PAN(SCRAM,LIN[0],ERRORCOUNT,INFO[LASTSEQROW,LASTSEQUENCE-1])            09386000 T  0080:0054:0
                        ;                                                                   09386500 T  0080:0057:2
                        WRITELINE;                                                          09387000 T  0080:0057:3
                             PEN(LIN[0],PRTIMAX,T1:=(L+3)DIV 4,T1:=CORADR+T,                09388000 T  0080:0067:3
                      ((T+29)DIV 30+DISKADR)×30);                                           09389000 T  0080:0071:3
                             WRITELINE;                                                     09389500 T  0080:0074:0
                             LOCK(LINE,RELEASE); END;                                       09390000 T  0080:0084:1
PRT(666) = *SEGMENT DESCRIPTOR*
                                                                                  80 IS   87 LONG, NEXT SEG   73
                    IF ERRORCOUNT ≠ 0 THEN I←0/0 ELSE                                       09391000 T  0073:0092:0
                        BEGIN                                                               09392000 T  0073:0094:1
                        ARRAY SAVINFO[0:31,0:255],                                          09392300 T  0073:0095:2
PRT(667) = *SEGMENT DESCRIPTOR*
                                                                                 START OF SEGMENT **********   81
STACK(F+2) = SAVINFO
                        INFO[0:200,0:255]; % FOR LARGE MCP"S,                               09392500 T  0081:0003:2
STACK(F+3) = INFO
                                    INTEGER SAVNDX,NONSAVNDX,N;                             09393000 T  0081:0005:3
STACK(F+4) = SAVNDX
STACK(F+5) = NONSAVNDX
STACK(F+6) = N
                        INTEGER Q,J,K,M;                                                    09393010 T  0081:0005:3
STACK(F+7) = Q
STACK(F+10) = J
STACK(F+11) = K
STACK(F+12) = M
                        BOOLEAN TSSTOG; REAL T;                                             09393020 T  0081:0005:3
STACK(F+13) = TSSTOG
STACK(F+14) = T
                        REAL PROCEDURE PUSHER(GRINCH,GOT,XMAS); VALUE XMAS; REAL XMAS;      09393050 T  0081:0005:3
PRT(670) = PUSHER
                                            ARRAY GOT[0]; ARRAY GRINCH [0,0];               09393060 T  0081:0005:3
                             BEGIN                                                          09393070 T  0081:0005:3
                             REAL WHO,WHAT;                                                 09393080 T  0081:0005:3
                                                                                 START OF SEGMENT **********   82
STACK(F+3) = WHO
STACK(F+4) = WHAT
                             DEFINE LINKR = [32:8]#;                                        09393090 T  0082:0000:0
                    %                                                                       09393100 T  0082:0000:0
                             IF WHO:=XMAS,LINKC ≤ 225 THEN                                  09393110 T  0082:0000:0
                                  BEGIN                                                     09393120 T  0082:0001:3
```

```
                    MOVE(30,GRINCH[XMAS,LINKR,WHO],GOT[0]);                                  09393130 T  0082:0002:0
                    PUSHER:=XMAS + 30;                                                       09393140 T  0082:0005:2
                    END                                                                      09393150 T  0082:0006:1
               ELSE BEGIN                                                                    09393160 T  0082:0006:1
                    MOVE(WHAT:=256-WHO,GRINCH[XMAS,LINKR,WHO],GOT[0]);                       09393170 T  0082:0007:2
                    XMAS:=XMAS + WHAT;                                                       09393180 T  0082:0011:2
                    MOVE(WHO:=30-WHAT, GRINCH[XMAS,LINKR,0], GOT[WHAT]);                     09393190 T  0082:0012:0
                    PUSHER:=XMAS + WHO;                                                      09393200 T  0082:0016:0
                    END;                                                                     09393220 T  0082:0017:3
               END PUSHER;                                                                   09393230 T  0082:0017:3
                                                                                          82 IS   20 LONG, NEXT SEG   81


          PROCEDURE PUSHEE(GRINCH,N,B,Y); VALUE N,B,Y; REAL N,B,Y;                           09393240 T  0081:0005:3
PRT(671) = PUSHEE
               ARRAY GRINCH[0,0];                                                            09393250 T  0081:0005:3
               BEGIN                                                                         09393260 T  0081:0005:3
               REAL I,J,X;                                                                   09393270 T  0081:0005:3
                                                                                       START OF SEGMENT **********   83
STACK(F+2) = I
STACK(F+3) = J
STACK(F+4) = X
               DEFINE LINKR = [32:8]#;                                                       09393280 T  0083:0000:0
               J:=Y;                                                                         09393290 T  0083:0000:0
               I:=B + N;                                                                     09393300 T  0083:0000:1
               WHILE B < I DO                                                                09393310 T  0083:0002:0
                    BEGIN                                                                    09393320 T  0083:0003:2
                    IF Y:=B,LINKC ≤ 225 THEN                                                 09393330 T  0083:0003:2
                    BEGIN                                                                    09393340 T  0083:0005:2
                    MOVE(30,CODE(J),GRINCH[B,LINKR,Y]);                                      09393350 T  0083:0005:3
                    J:=J + 30;                                                               09393360 T  0083:0011:2
                    B:=B + 30;                                                               09393370 T  0083:0012:0
                    END                                                                      09393380 T  0083:0013:3
               ELSE BEGIN                                                                    09393390 T  0083:0013:3
                    MOVE(X:=256-Y,CODE(J),GRINCH[B,LINKR,Y]);                                09393400 T  0083:0014:0
                    B:=B + X;                                                                09393410 T  0083:0020:1
                    J:=J + X;                                                                09393420 T  0083:0021:3
                    MOVE(Y:=30-X,CODE(J),GRINCH[B,LINKR,0]);                                 09393430 T  0083:0023:2
                    B:=B + Y;                                                                09393440 T  0083:0029:3
                    J:=J + Y;                                                                09393450 T  0083:0030:1
                    END;                                                                     09393460 T  0083:0032:0
               END;                                                                          09393470 T  0083:0032:0
          END PUSHEE;                                                                        09393480 T  0083:0032:1
                                                                                          83 IS   35 LONG, NEXT SEG   81


     STREAM PROCEDURE FIXHDR(F,N); VALUE N;                                                  09393700 T  0081:0005:3
PRT(672) = FIXHDR
          BEGIN SI←F; SI←SI-24; DI←LOC F; DS←WDS;                                          09393710 T  0081:0005:3
               SI←F; 14(SI←SI+8); DI←LOC F; DS←WDS;                                        09393720 T  0081:0007:2
               DI←F; DI←DI+38; SI← LOC N;                                                  09393730 T  0081:0008:1
               SI←SI+7; DS←CHR;                                                            09393740 T  0081:0009:2
          END FIXHDR;                                                                      09393750 T  0081:0009:3
```

```
          LABEL EOF;                                                                                    09394000 T  0081:001:0
     IF NOT INTOG THEN                                                                                  09394100 T  0081:001:0
BEGIN                                                                                                   09394200 T  0081:001:2
          L←(L+3)DIV 4; COMMENT L←NUM. OF WORDS IN OUTER BLOCK;                                         09395000 T  0081:001:3
          FILL SAVINFO[0,*] WITH                                                                        09395100 T  0081:001:2
               0CT77000000000000015,                                                                    09395200 T  0081:001:0
                                                                                                  START OF SEGMENT **********   84
               0CT0253010477527705,                                                                     09395300 T  0081:001:2
               0CT0051000000000000,                                                                     09395400 T  0081:001:2
               0CT0441070001000062;                                                                     09395500 T  0081:001:2
                                                                                                   84 IS    4 LONG, NEXT SEG =  81
          Q ← -1;                                                                                       09395700 T  0081:001:2
     PUSHEE(SAVINFO,L,4,5);                                                                             09396000 T  0081:001:0
     SAVNDX:=L;                                                                                         09397000 T  0081:001:1
END;                                                                                                    09397100 T  0081:001:2
          REWIND(COD;SK);                                                                               09398000 T  0081:001:2
 DO BEGIN IF   REED=0 THEN GO TO EOF;                                                                   09399000 T  0081:002:2
          N←FETCH(MKABS(CODE(0)))=1;                                                                    09400000 T  0081:002:1
          IF BOOLEAN(FETCH(MKABS(CODE(1)))) THEN                                                        09401000 T  0081:002:1
BEGIN                                                                                                   09402000 T  0081:003:3
     PUSHEE(SAVINFO,N,SAVNDX,1);                                                                        09402100 T  0081:003:0
     SAVNDX:=SAVNDX +N;                                                                                 09403000 T  0081:003:0
END ELSE BEGIN                                                                                          09404000 T  0081:003:3
     IF DECKTOG THEN                                                                                    09405000 T  0081:003:0
               STACKHEAD[Q←Q+1] ← 1024×NONSAVNDX+N;                                                     09405500 T  0081:003:1
               PUSHEE(INFO,N,NONSAVNDX,1);                                                              09406000 T  0081:004:1
               NONSAVNDX:=((NONSAVNDX + N + 29)DIV 30)×30;                                              09407000 T  0081:004:1
          END;                                                                                          09408000 T  0081:004:3
          END UNTIL FALSE;                                                                              09412000 T  0081:004:3
     FOF:      N←(SAVNDX+29) DIV 30; COMMENT NUMBER OF DISK SEGMENTS                                    09413000 T  0081:004:0
               OCCUPIED BY SAVE PROCEDURES AND ARRAYS;                                                  09414000 T  0081:005:1
     IF INTOG AND NOT DECKTOG THEN                                                                      09414010 T  0081:005:1
BEGIN    % INTRINSIC FUNCTION OPTION                                                                    09414020 T  0081:005:1
     FOR J:=USEROPINX STEP 2 UNTIL OPARSIZE DO % IS TIMESHARING SET                                     09414022 T  0081:005:2
     IF OPTIONS[J] = "@TIMES" THEN                                                                      09414024 T  0081:005:2
     BEGIN TSSTOG:=BOOLEAN(OPTIONS[J+1]); J:=OPARSIZE END;                                              09414026 T  0081:005:0
     I ← PRTBASE + 1; J ← 0;                                                                            09414030 T  0081:006:2
     DO IF GT1 ← PRT[I] ≠0 THEN                                                                         09414040 T  0081:006:2
     BEGIN                                                                                              09414050 T  0081:006:2
          J ← J+1;                                                                                      09414060 T  0081:006:3
           SAVINFO[J,LINKR,J,LINKC] ←                                                                   09414070 T  0081:006:1
            0&GT1[8:8:10]                                                                               09414080 T  0081:006:2
            &GT1[33:18:15];                                                                             09414090 T  0081:007:0
     END UNTIL I:=I +1 ≥ PRTIMAX;                                                                       09414100 T  0081:007:2
      SAVINFO[0,0] ← J;      % # OF INTRINSICS                                                          09414110 T  0081:007:3
     SAVNDX ← MAXINTRINSIC;                                                                             09414120 T  0081:007:3
END ELSE BEGIN                                                                                          09414130 T  0081:007:0
               I←PRTBASE; DO IF GT1←PRT[I]≠0 THEN                                                       09415000 T  0081:007:8:0
          BEGIN IF GT1.[1:5]≠LDES THEN                                                                  09415500 T  0081:008:1:2
          BEGIN IF (GT1←GT1&(GT1.[33:15]+L)[33:33:15]).[6:2]≠3 THEN                                     09416000 T  0081:008:2:1
                  GT1←GT1&(GT1.[18:15]+N)[18:33:15];                                                    09417000 T  0081:008:6:1
          END;                                                                                          09417500 T  0081:008:9:2
               MDESC(GT1,SAVINFO[I,LINKR,I,LINKC]);                                                     09418000 T  0081:008:9:2
          END ELSE SAVINFO[I,LINKR,I,LINKC]:=0 UNTIL I:=I+1≥PRTIMAX;                                    09419000 T  0081:009:2:0
               MDESC(0&1[2:47:1],SAVINFO[0,PRTBASE-1]);                                                 09419100 T  0081:009:7:2
```

```
        SAVNDX ← 30 × N;                                                                09420000 T  0081:0101:3
END;                                                                                    09420010 T  0081:0102:1
        I ← 0; J ← -1;                                                                  09420020 T  0081:0102:1
                                                                                        09420100 T  0081:0104:1
        IF NOT DECKTOG THEN                                                             09421000 T  0081:0104:1
  BEGIN                                                                                 09421500 T  0081:0105:3
      DO                                                                                09422000 T  0081:0106:0
  BEGIN                                                                                 09423000 T  0081:0106:0
        I:=PUSHER(SAVINFO,ELBAT,I);                                                     09424000 T  0081:0106:0
        J:=J + 1;                                                                       09425000 T  0081:0109:2
        WRITE(DISK,30,ELBAT[*]);                                                        09425900 T  0081:0110:0
  END UNTIL I ≥ SAVNDX;                                                                 09426000 T  0081:0114:1
        I:=0;                                                                           09427000 T  0081:0115:3
        WHILE I < NONSAVNDX DO                                                          09427100 T  0081:0116:1
  BEGIN                                                                                 09427200 T  0081:0118:0
        I:=PUSHER(INFO,ELBAT,I);                                                        09427500 T  0081:0118:0
        J:=J + 1;                                                                       09428000 T  0081:0121:2
        WRITE(DISK,30,ELBAT[*]);                                                        09429000 T  0081:0122:1
  END;                                                                                  09430000 T  0081:0126:1
        N←IF INTOG THEN IF TSSTOG THEN                                                  09430050 T  0081:0127:2
            TSSINTYPE ELSE DCINTYPE ELSE MCPTYPE;                                       09430060 T  0081:0128:1
        FIXHDR(DISK,N);                                                                 09430075 T  0081:0131:3
        LOCK(DISK,*);                                                                   09430100 T  0081:0132:1
  END  ELSE                                                                             09431000 T  0081:0134:1
  BEGIN ELBAT[0]←0; I←16;                                                               09432000 T  0081:0134:1
        DO BEGIN MOVE(8,SAVINFO[I,LINKR,I,LINKC],ELBAT[1]);                             09433000 T  0081:0137:2
            ELBAT[9]←B2D(I+96)&1[11:47:1]&(I+96)[23:35:1];                              09434000 T  0081:0140:1
            WRITE(DECK,10,ELBAT[*]);                                                    09435000 T  0081:0145:2
        END UNTIL I←I+8≥SAVNDX;                                                         09436000 T  0081:0149:3
    FILL ELBAT[*] WITH 0,                                                               09437000 T  0081:0151:3
                                                                                START OF SEGMENT **********    85
        0CT7500000000000012,                                                        09438000 T  0081:0153:3
        0CT0004535530611765,                                                        09439000 T  0081:0153:3
        0CT7006000404210435,                                                        09440000 T  0081:0153:3
        0CT7700000000000015,                                                        09441000 T  0081:0153:3
        0CT0253010477527705,                                                        09442000 T  0081:0153:3
        0CT0051000004410046,                                                        09443000 T  0081:0153:3
        0CT0441070001000062,                                                        09444000 T  0081:0153:3
        0CT0040413100000000,                                                        09445000 T  0081:0153:3
        0CT0001000000001101;                                                        09446000 T  0081:0153:3
                                                                                  85 IS   10 LONG, NEXT SEG   81
    WRITE(DECK,10,ELBAT[*]);                                                        09447000 T  0081:0153:3
    ELBAT[0] ←0&REAL(DECKTOG)[1:19:17];                                             09447010 T  0081:0157:3
    FOR I ← 0 STEP 1 UNTIL Q DO                                                     09447020 T  0081:0160:1
        BEGIN K ← STACKHEAD[I],[23:15];                                             09447030 T  0081:0162:0
            M ← STACKHEAD[I],[38:10];                                               09447040 T  0081:0163:3
            FOR J ← 0 STEP 8 UNTIL M DO BEGIN                                       09447050 T  0081:0165:2
                MOVE(8,INFO[(J+K),LINKR,(J+K),LINKC],                               09447060 T  0081:0166:0
                    ELBAT [1]);                                                     09447070 T  0081:0169:3
                ELBAT[9] ← B2D(J)&"310"[1:31:17];                                   09447080 T  0081:0170:1
                WRITE(DECK,10,ELBAT[*]) END;                                        09447090 T  0081:0173:2
        END;                                                                        09447100 T  0081:0179:3
END END END PROGRAM;                                                                09448000 T  0081:0182:0
PRT(673) = *SEGMENT DESCRIPTOR*
                                                                                  81 IS  186 LONG, NEXT SEG   73
                                                                                  73 IS  101 LONG, NEXT SEG    3
```

```
                COMMENT THIS SECTION CONTAINS GENERATORS USED BY THE BLOCK ROUTINE;                10000000 T  0003:0505:3
                PROCEDURE DEFINEGEN(MACRO,J); VALUE MACRO,J; BOOLEAN MACRO; REAL J;               10228000 T  0003:0505:3
PRT(674) = DEFINEGEN
                    BEGIN
                       OWN INTEGER CHARCOUNT, REMCOUNT;                                           10229000 T  0003:0505:3
                                                                                                  10230000 T  0003:0505:3
                                                                                        START OF SEGMENT **********   86
PRT(675) = CHARCOUNT
PRT(676) = REMCOUNT
                COMMENT CHARCOUNT CONTAINS NUMBER OFCHARACTORS OF THE DEFINE THAT WE              10231000 T  0086:0000:0
                        HAVE PUT INTO INFO,  REMCOUNT CONTAINS NUMBER OF CHARACT-                 10232000 T  0086:0000:0
                        ORS REMAINING IN THIS ROW OF INFO;                                        10233000 T  0086:0000:0
                PROCEDURE PUTOGETHER(CHAR); REAL CHAR;                                            10234000 T  0086:0000:0
PRT(677) = PUTOGETHER
                    BEGIN                                                                         10235000 T  0086:0000:0
                STREAM PROCEDURE PACKINFO(INFO,ISKIP,COUNT,ASKIP,ACCUM);                          10236000 T  0086:0000:0
                                                                                        START OF SEGMENT **********   87
PRT(700) = PACKINFO
                        VALUE ISKIP,COUNT,ASKIP;                                                  10237000 T  0087:0000:0
                        BEGIN DI ← INFO; DI ← DI+ISKIP;                                           10238000 T  0087:0000:0
                              SI ← ACCUM;SI ← SI+ASKIP; SI ← SI+3;                                10239000 T  0087:0000:1
                              DS ← COUNT CHR END PACKINFO;                                        10240000 T  0087:0001:3



                        INTEGER COUNT,SKIPCOUNT;                                                  10241000 T  0087:0002:1
STACK(F+2) = COUNT
STACK(F+3) = SKIPCOUNT
                      IF (COUNT ← CHAR.[12:6]) + CHARCOUNT > 2047                                 10242000 T  0087:0002:1
                        THEN BEGIN FLAG(142);  TB1← TRUE  END                                     10243000 T  0087:0004:1
                        ELSE BEGIN                                                                10244000 T  0087:0007:2
                           IF COUNT > REMCOUNT                                                    10245000 T  0087:0009:2
                             THEN BEGIN                                                           10246000 T  0087:0009:2
                                SKIPCOUNT ← COUNT-(COUNT←REMCOUNT);                               10247000 T  0087:0010:0
                                REMCOUNT ← 2047 END                                               10248000 T  0087:0012:0
                             ELSE REMCOUNT ← REMCOUNT-COUNT;                                      10249000 T  0087:0012:1
                           GT1 ← CHARCOUNT DIV 8 + NEXTINFO;                                      10250000 T  0087:0016:0
                           PACKINFO(INFO[GT1.LINKR,GT1.LINKC],CHARCOUNT.[45:3],                   10251000 T  0087:0018:0
                           COUNT,0,CHAR);                                                         10252000 T  0087:0021:2
                           IF SKIPCOUNT ≠ 0 THEN                                                  10253000 T  0087:0022:0
                               PACKINFO(INFO[NEXTINFO.LINKR+1,0],0,SKIPCOUNT,                     10254000 T  0087:0023:2
                                  COUNT,CHAR);                                                    10255000 T  0087:0026:1
                           CHARCOUNT ← CHARCOUNT+SKIPCOUNT+COUNT END                              10256000 T  0087:0027:2
                    END PUTOGETHER;                                                               10257000 T  0087:0029:2
                                                                                        87 IS    32 LONG, NEXT SEG   86



                STREAM PROCEDURE SCAN(D,S,Q,N,J); VALUE J,N,Q;                                    10257100 T  0086:0000:0
PRT(701) = SCAN
                    BEGIN DI←D;DI←DI+11;SI←S;SI←SI+3;                                             10257200 T  0086:0000:0
                         IF N SC=DC THEN                                                          10257300 T  0086:0001:2
                         IF SC>"0" THEN                                                           10257400 T  0086:0001:3
                         BEGIN DI←LOC J; DI←DI+7;                                                 10257500 T  0086:0002:1
```

```
                                IF SC≤DC THEN                                                    10257600 T  0086:0003:3
                                BEGIN J←SI;DI←J;SI←LOC Q;SI←SI+6;DS←CHR;                          10257700 T  0086:0004:0
                                     DI←S;DI←DI+2;DS←CHR;                                        10257800 T  0086:0006:0
                           END  END  END;                                                        10257900 T  0086:0006:1


                              INTEGER LASTRESULT;                                                10258000 T  0086:0007:2
STACK(F+2) = LASTRESULT
                                REAL K,N,ELCLASS;                                                10258100 T  0086:0007:2
STACK(F+3) = K
STACK(F+4) = N
STACK(F+5) = ELCLASS
                                DEFINE I=NXTELBT#;                                               10258200 T  0086:0007:2
                                LABEL FINAL,PACKIN;                                              10258300 T  0086:0007:2
                                LABEL BACK,SKSC,EXIT;                                            10259000 T  0086:0007:2
                                TB1← FALSE;                                                      10260000 T  0086:0007:2
                                CHARCOUNT←(NEXTINFO-LASTINFO)×8;                                 10261000 T  0086:0007:3
                                DEFINECTR ← 1; LASTRESULT ← 2;                                   10262000 T  0086:0009:3
                                REMCOUNT ← (256 - NEXTINFO MOD 256) × 8;                         10263000 T  0086:0011:2
                                NEXTINFO←LASTINFO;                                               10263100 T  0086:0013:2
                                IF J≠0 THEN N←TAKE(LASTINFO+1),[12:6];                           10263110 T  0086:0014:0
                                K←0;                                                             10263200 T  0086:0017:3
                     BACK:      STOPDEFINE←TRUE;                                                 10263300 T  0086:0018:0
                                ELCLASS←TABLE(NXTELBT);                                          10263400 T  0086:0019:3
                     SKSC:      NXTELBT←NXTELBT-1;                                               10263500 T  0086:0021:2
                                IF MACRO THEN                                                    10263600 T  0086:0022:0
                                BEGIN IF ELCLASS=COMMA THEN                                      10263700 T  0086:0022:1
                                IF K=0 THEN                                                      10263800 T  0086:0023:3
                     FINAL:     BEGIN PUTOGETHER("1#0000"); GO TO EXIT END                       10263900 T  0086:0025:2
                                ELSE GO PACKIN;                                                  10264000 T  0086:0029:2
                                IF ELCLASS=LEFTPAREN OR ELCLASS=LFTBRKET THEN                    10264100 T  0086:0029:2
                                BEGIN K←K+1; GO TO PACKIN END;                                   10264200 T  0086:0030:1
                                IF ELCLASS=RTPAREN OR ELCLASS=RTBRKET THEN                       10264300 T  0086:0033:2
                                IF K←K-1<0 THEN GO FINAL ELSE GO PACKIN;                         10264400 T  0086:0034:1
                                IF ELCLASS=SEMICOLON THEN                                        10264410 T  0086:0038:0
                                     BEGIN FLAG(142); GO TO FINAL END ELSE GO PACKIN             10264420 T  0086:0038:1
                           END;                                                                  10264500 T  0086:0040:1
                                IF J≠0 THEN                                                      10264600 T  0086:0040:1
                                IF ACCUM[1].[12:6]-1=N THEN                                      10264700 T  0086:0041:2
                                     SCAN(INFO[LASTINFO, LINKR ,LASTINFO, LINKC],                10264800 T  0086:0043:3
                                          ACCUM[1],N+770,N,J);                                   10264900 T  0086:0046:1
                     PACKIN:                                                                     10264910 T  0086:0048:1
                                IF RESULT = 4                                                    10265000 T  0086:0049:2
                                  THEN BEGIN                                                     10266000 T  0086:0049:2
                        COMMENT  INSERT " MARKS - 2130706432 IS DECIMAL FOR 1"0000;              10267000 T  0086:0050:0
                                   PUTOGETHER(2130706432);                                       10268000 T  0086:0050:0
                                   PUTOGETHER(ACCUM[1]);                                         10269000 T  0086:0051:2
                                   PUTOGETHER(2130706432) END                                    10270000 T  0086:0052:0
                                ELSE BEGIN                                                       10271000 T  0086:0052:1
                                  IF BOOLEAN(RESULT) AND BOOLEAN(LASTRESULT)                     10272000 T  0086:0055:2
                                   THEN PUTOGETHER("1 0000"); COMMENT INSERT BLANK;              10273000 T  0086:0055:2
                                  PUTOGETHER(ACCUM[1]) END;                                      10274000 T  0086:0057:2
                                IF TB1 THEN GO TO EXIT;                                          10275000 T  0086:0058:0
                                LASTRESULT ← RESULT;                                             10276000 T  0086:0059:2
                                IF MACRO THEN GO BACK;                                           10276500 T  0086:0059:3
                                IF ELCLASS=DECLARATORS AND ELBAT[I].ADDRESS = DEFINEV            10277000 T  0086:0060:1
```

```
                        THEN BEGIN DEFINECTR ← DEFINECTR+1; GO BACK END;
                     IF ELCLASS ≠ CROSSHATCH THEN GO BACK;
                     IF DEFINECTR ≠ 1
                       THEN BEGIN STOPDEFINE ← TRUE;
                          IF ELCLASS←TABLE(I)≠COMMA THEN
                          DEFINECTR←DEFINECTR-1; GO SKSC END;
EXIT;          DEFINECTR← 0;
               NEXTINFO ←(CHARCOUNT+7) DIV 8←NEXTINFO;
          END  DEFINEGEN;



          PROCEDURE DBLSTMT;
               BEGIN
               REAL S,T;

STACK(F+2) = S
STACK(F+3) = T
               LABEL L1,L2,L3,EXIT;
               S←0;
               IF STEPI≠LEFTPAREN THEN ERR(281)
               ELSE
          L1;  BEGIN
                    IF STEPI=COMMA THEN
                    BEGIN
                    DPTOG←TRUE;
                    IF STEPI=ADOP THEN STEPIT;
                    EMITNUM(NLO);
                    EMITNUM(IF ELBAT[I-1].ADDRESS =SUB THEN -NHI ELSE NHI);
                    DPTOG←FALSE;
                    STEPIT;
                         GO TO L2;
                    END;
                    IF TABLE(I+1)=COMMA THEN
                         BEGIN
                         IF ELCLASS=ADOP  OR ELCLASS=MULOP THEN
                              BEGIN
                              EMITO(ELBAT[I].ADDRESS+1);
                              IF S←S-1≤0 THEN FLAG(282); STEPIT;
                              GO TO L3;
                              END;
                         IF ELCLASS=ASSIGNOP THEN
                              BEGIN
                              IF S≠1 THEN FLAG(283); S←0;    STEPIT;
                              DO
                              BEGIN
                          IF ELCLASS  ≠COMMA THEN BEGIN ERR(284);GO EXIT END;
                              STEPIT;
                              IF ELCLASS≤INTID AND ELCLASS≥REALID THEN
                              BEGIN  EMITN(ELBAT[I].ADDRESS); STEPIT END
                                                  ELSE VARIABLE(FL);
                                             EMITO(STD) END UNTIL S←S+1=2 ;
                              IF ELCLASS≠RTPAREN THEN ERR(285) ELSE STEPIT;
                              GO TO EXIT;
                              END;
```

```
                                                  10278000 T  0086:0062:1
                                                  10279000 T  0086:0067:2
                                                  10280000 T  0086:0068:0
                                                  10281000 T  0086:0068:1
                                                  10282000 T  0086:0070:0
                                                  10283000 T  0086:0072:0
                                                  10284000 T  0086:0074:0
                                                  10285000 T  0086:0075:3
                                                  10286000 T  0086:0078:0
                                            86 IS   81 LONG, NEXT SEG    3



                                                  12002000 T  0003:0505:3
                                                  12003000 T  0003:0505:3
                                                  12004000 T  0003:0505:3
                                          START OF SEGMENT **********   88

                                                  12005000 T  0088:0000:0
                                                  12006000 T  0088:0000:0
                                                  12007000 T  0088:0000:1
                                                  12008000 T  0088:0002:1
                                                  12009000 T  0088:0003:2
                                                  12010000 T  0088:0004:0
                                                  12011000 T  0088:0005:2
                                                  12012000 T  0088:0005:3
                                                  12013000 T  0088:0006:0
                                                  12014000 T  0088:0008:0
                                                  12015000 T  0088:0009:2
                                                  12016000 T  0088:0013:2
                                                  12017000 T  0088:0014:0
                                                  12018000 T  0088:0014:1
                                                  12019000 T  0088:0015:2
                                                  12020000 T  0088:0015:2
                                                  12021000 T  0088:0016:1
                                                  12022000 T  0088:0017:2
                                                  12023000 T  0088:0019:2
                                                  12024000 T  0088:0019:3
                                                  12025000 T  0088:0021:3
                                                  12026000 T  0088:0025:2
                                                  12027000 T  0088:0025:3
                                                  12028000 T  0088:0025:3
                                                  12029000 T  0088:0026:0
                                                  12030000 T  0088:0026:1
                                                  12031000 T  0088:0030:0
                                                  12032000 T  0088:0030:0
                                                  12033000 T  0088:0030:0
                                                  12034000 T  0088:0032:1
                                                  12035000 T  0088:0033:2
                                                  12036000 T  0088:0034:1
                                                  12037000 T  0088:0037:2
                                                  12038000 T  0088:0038:1
                                                  12039000 T  0088:0041:3
                                                  12040000 T  0088:0044:1
                                                  12041000 T  0088:0045:2
```

```
                    IF ELCLASS≤INTID AND ELCLASS≥BOOID THEN                                  12042000 T  0088:0045:2
                        BEGIN                                                                12043000 T  0088:0046:1
                        CHECKER(T←ELBAT[I]);                                                 12044000 T  0088:0047:2
                        STEPIT;STEPIT;                                                       12045000 T  0088:0048:1
                        AEXP;                                                                12046000 T  0088:0049:3
                        EMITV(T,ADDRESS);                                                    12047000 T  0088:0050:0
                        GO TO L2;                                                            12048000 T  0088:0051:3
                        END;                                                                 12049000 T  0088:0052:0
                    END    ;                                                                 12050000 T  0088:0052:0
                        AEXP;                                                                12051000 T  0088:0052:0
                        IF ELCLASS≠COMMA THEN BEGIN ERR(284);GO EXIT                         12052000 T  0088:0052:1
                                                                              END;           12053000 T  0088:0055:2
                        STEPIT; AEXP; EMITO(XCH);                                            12054000 T  0088:0055:2
L2:         S←S+1;                                                                           12055000 T  0088:0056:1
L3:         IF ELCLASS≠COMMA THEN BEGIN ERR(284);GO TO EXIT END;                             12056000 T  0088:0058:0
            GO TO L1;                                                                        12057000 T  0088:0061:3
EXIT:END                                                                                     12058000 T  0088:0062:0
    END  DBLSTMT;                                                                            12059000 T  0088:0062:0
                                                                                      88 IS    65 LONG, NEXT SEG    3


REAL PROCEDURE FIXDEFINEINFO(T); VALUE T; REAL T;                                            12101000 T  0003:0505:3
    BEGIN REAL K,S,P,J,EL;                                                                   12102000 T  0003:0505:3
                                                                                      START OF SEGMENT **********   89

STACK(F+3) = K
STACK(F+4) = S
STACK(F+5) = P
STACK(F+6) = J
STACK(F+7) = EL
                    STREAM PROCEDURE SET(S,D,K,F); VALUE K,E;                                12103000 T  0089:0000:0
PRT(702) = SET
        BEGIN SI←S;SI←SI+11;DI←D;DI←DI+3;DS←K CHR;                                           12104000 T  0089:0000:0
            SI←LOC E; SI←SI+6; DS←2 CHR;                                                     12105000 T  0089:0001:3
        END;                                                                                 12106000 T  0089:0002:0


            MACROID←TRUE;                                                                    12107000 T  0089:0002:1
            P←(FIXDEFINEINFO←T).ADDRESS;                                                     12108000 T  0089:0003:3
            K←COUNT;                                                                         12109000 T  0089:0005:3
            S←SCRAM;                                                                         12110000 T  0089:0006:0
            STREAMTOG←TRUE & STREAMTOG[1:3:45] ;                                             12110100 T  0089:0007:2
            STOPDEFINE←TRUE;                                                                 12111000 T  0089:0008:1
            EL←TABLE(NXTELBT);                                                               12112000 T  0089:0009:3
            NXTELBT←NXTELBT+1;                                                               12113000 T  0089:0010:1
            IF EL≠LEFTPAREN AND EL≠LFTBRKET THEN                                             12114000 T  0089:0012:0
                FLAG(141)                                                                    12115000 T  0089:0013:3
            ELSE DO BEGIN J←J+1;                                                             12116000 T  0089:0014:1
                SET(INFO[T,LINKR,T,LINKC],ACCUM[1],K,64×J+12);                               12117000 T  0089:0017:2
                ACCUM[1].[12:6]←K+2;                                                         12118000 T  0089:0022:0
                ACCUM[0]←0;                                                                  12119000 T  0089:0025:2
                ACCUM[0].CLASS←DEFINEDID;                                                    12120000 T  0089:0026:0
                COUNT←K+2;                                                                   12121000 T  0089:0028:1
                SCRAM←ACCUM[1] MOD 125;                                                      12122000 T  0089:0030:0
```

```
                    E;
                    DEFINEGEN(TRUE,0);
                END UNTIL EL←ELBAT[NXTELBT].CLASS≠COMMA;
                IF EL≠RTPAREN AND EL≠RTBRKET OR J≠P THEN FLAG(141);
                MACROID←FALSE;
                STREAMTOG←STREAMTOG.[1:45] ;
            END;
```

```
                PROCEDURE SCATTERELBAT;
                    BEGIN
                        REAL T;
STACK(F+2) = T
                        T ← ELBAT[I];
                        KLASSF ← T.CLASS;
                        FORMALF ← BOOLEAN(T.FORMAL);
                        VONF ← BOOLEAN(T.VO);
                        LEVELF ← T.LVL;
                        ADDRSF ← T.ADDRESS;
                        INCRF ← T.INCR;
                        LINKF ← T.LINK;
                    END  SCATTERELBAT;
```

```
                PROCEDURE CHKSOB;
PRT(703) = CHKSOB
                    IF GTA1[J←J-1]≠0 THEN FLAG(23);
```

```
                DEFINE
                        ADDC=532480#,
                        SUBC=1581056#,
                        EMITSTORE=EMITPAIR#;
                 PROCEDURE PURGE(STOPPER);
                          VALUE STOPPER;
                          REAL STOPPER;
                 BEGIN
                   INTEGER POINTER;
STACK(F+2) = POINTER
                     LABEL RECOV; DEFINE ELCLASS = KLASSF#;
                     REAL J,N,OCR,TL,ADD;
STACK(F+3) = J
STACK(F+4) = N
STACK(F+5) = OCR
STACK(F+6) = TL
STACK(F+7) = ADD
                       POINTER←LASTINFO;
```

```
                                                    12123000 T  0089:003!3
                                                    12124000 T  0089:003210
                                                    12125000 T  0089:003312
                                                    12126000 T  0089:003513
                                                    12127000 T  0089:003913
                                                    12127100 T  0089:004010
                                                    12128000 T  0089:004210
                                                 89 IS   46 LONG, NEXT SEG    3
```

```
                                                    13197000 T  0003:050513
                                                    13198000 T  0003:050513
                                                    13199000 T  0003:050513
                                                 START OF SEGMENT **********   90
                                                    13200000 T  0090:000010
                                                    13201000 T  0090:000112
                                                    13202000 T  0090:000210
                                                    13203000 T  0090:000313
                                                    13204000 T  0090:000411
                                                    13205000 T  0090:000610
                                                    13206000 T  0090:000712
                                                    13207000 T  0090:000811
                                                    13208000 T  0090:000913
                                                 90 IS   12 LONG, NEXT SEG    3
```

```
                                                    13209000 T  0003:050513
                                                    13210000 T  0003:050513
```

```
                                                    13211000 T  0003:050913
                                                    13212000 T  0003:050913
                                                    13213000 T  0003:050913
                                                    13214000 T  0003:050913
                                                    13215000 T  0003:050913
                                                    13216000 T  0003:050913
                                                    13217000 T  0003:050913
                                                    13218000 T  0003:050913
                                                    13219000 T  0003:050913
                                                 START OF SEGMENT **********   91
                                                    13220000 T  0091:000010
                                                    13221000 T  0091:000010
```

```
                                                    13222000 T  0091:000010
```

```
WHILE POINTER ≥ STOPPER                                                                  13223000 T  0091:0000:1
DO                                                                                       13224000 T  0091:0001:2
  BEGIN                                                                                  13225000 T  0091:0002:0
        IF ELCLASS←(GT1←TAKE(POINTER)).CLASS=NONLITNO                                    13226000 T  0091:0002:0
        THEN BEGIN                                                                       13227000 T  0091:0004:1
        NCII←NCII-1;                                                                     13228000 T  0091:0005:3
             EMITNUM(TAKE(POINTER+1));                                                   13229000 T  0091:0006:1
             EMITSTORE(MAXSTACK,STD);                                                    13230000 T  0091:0008:1
             MAXSTACK←(G←MAXSTACK)+1;                                                    13231000 T  0091:0009:3
             J←L; L←GT1.LINK;                                                            13232000 T  0091:0011:2
             DO                                                                          13233000 T  0091:0013:2
               BEGIN                                                                     13234000 T  0091:0014:0
                  GT4←GET(L);                                                            13235000 T  0091:0014:0
                  EMITV(G)                                                               13236000 T  0091:0015:2
               END                                                                       13237000 T  0091:0015:3
             UNTIL (L←GT4)=4095;                                                         13238000 T  0091:0016:0
             L←J;                                                                        13239000 T  0091:0017:3
             POINTER←POINTER-GT1.INCR                                                    13240000 T  0091:0018:1
             END                                                                         13241000 T  0091:0018:1
        ELSE                                                                             13242000 T  0091:0020:0
        BEGIN                                                                            13243000 T  0091:0020:0
        IF NOT BOOLEAN(GT1,FORMAL)                                                       13244000 T  0091:0022:0
          THEN BEGIN                                                                     13245000 T  0091:0022:0
             IF ELCLASS = LABELID                                                        13246000 T  0091:0023:3
               THEN BEGIN                                                                13247000 T  0091:0023:3
                 ADD ← GT1.ADDRESS;                                                      13248000 T  0091:0024:1
                 IF NOT BOOLEAN(OCR←TAKE(GIT(POINTER))).[1:1]                            13249000 T  0091:0026:0
                    THEN IF OCR.[36:12] ≠ 0 OR ADD ≠ 0                                   13250000 T  0091:0028:0
                       THEN BEGIN GT1 ← 160; GO TO RECOV END;                            13251000 T  0091:0030:1
     IF ADD ≠ 0 THEN                                                                     13252000 T  0091:0033:2
PROGDESCBLDR(ADD,TRUE,OCR,[36:10],LDES) END                                              13252500 T  0091:0033:3
ELSE IF FALSE                                                                            13253000 T  0091:0036:0
       THEN BEGIN                                                                        13254000 T  0091:0037:2
          IF TAKE(POINTER+1) < 0                                                         13255000 T  0091:0037:3
            THEN BEGIN GT1 ← 162; GO TO RECOV END;                                       13256000 T  0091:0038:1
          OCR ←(J ←  TAKE(GIT(POINTER))),[24:12];                                        13257000 T  0091:0041:2
          N ← GET( (J←J.[36:12])+4); TL ← L;                                             13258000 T  0091:0043:3
          IF ADD ← GT1.ADDRESS ≠ 0                                                       13259000 T  0091:0047:2
           THEN BEGIN                                                                    13260000 T  0091:0048:1
                                                                                         13261000 T  0091:0049:3
             IF OCR ≠ 0                                                                  13262000 T  0091:0049:3
THEN BEGIN L←OCR=2; CALLSWITCH(POINTER); EMITO(BFW);END;                                 13263000 T  0091:0049:3
             L←J+11; EMITL(15); EMITO(RTS);                                              13264000 T  0091:0053:3
             FOR J ← 4 STEP 4 UNTIL N                                                    13265000 T  0091:0056:0
               DO BEGIN                                                                  13266000 T  0091:0056:0
                 EMITL(GNAT(GET(L)×4096+GET(L+1)));                                      13267000 T  0091:0057:2
                 EMITO(RTS) END END                                                      13268000 T  0091:0060:1
           ELSE BEGIN                                                                    13269000 T  0091:0063:3
             L ← J+13;                                                                   13270000 T  0091:0066:0
             FOR J ← 4 STEP 4 UNTIL N                                                    13271000 T  0091:0067:2
               DO BEGIN                                                                  13272000 T  0091:0067:2
                 GT1 ← GET(L)×4096+GET(L+1);                                             13273000 T  0091:0068:0
                 GOGEN(GT1,BFW) END;END;                                                 13274000 T  0091:0071:2
       L ← TL END                                                                        13277000 T  0091:0074:1
    ELSE IF ELCLASS ≥ PROCID AND ELCLASS ≤ INTPROCID                                     13278000 T  0091:0075:2
     THEN IF TAKE(POINTER+1) < 0                                                         13279000 T  0091:0078:0
       THEN BEGIN GT1 ← 161;                                                             13280000 T  0091:0080:1
RECOV:   MOVE(9,INFO[POINTER.LINKR,POINTER.LINKC],ACCUM);                                13281000 T  0091:0082:0
```

```
                        Q ← ACCUM[1]; FLAG(GT1); ERRORTOG ← TRUE END                                    13282000 T  0091:008610
                END;                                                                                    13283000 T  0091:008912
            XREFDUMP(POINTER); % DUMP XREF INFO                                          %110-          13283500 C  0091:008912
            GT2←TAKE(POINTER+1);                                                                        13284000 T  0091:009112
            GT3←GT2.PURPT;                                                                              13285000 T  0091:009211
         STACKHEAD[(0&GT2[12:12:36])MOD 125]←TAKE(POINTER).LINK;                                        13286000 T  0091:009410
            POINTER←POINTER-GT3                                                                         13287000 T  0091:009713
                END                                                                                     13288000 T  0091:009810
          END ;                                                                                         13289000 T  0091:009912
       LASTINFO←POINTER;                                                                                13290000 T  0091:009913
       NEXTINFO←STOPPER                                                                                 13291000 T  0091:010010
     END;                                                                                               13292000 T  0091:010010
                                                                                                  91 IS  105 LONG, NEXT SEG    3



    PROCEDURE F;                                                                                        13293000 T  0003:050913
    COMMENT                                                                                             13294000 T  0003:050913
           F IS THE PROCEDURE WHICH PLACES AN ENTRY IN INFO AND                                         13295000 T  0003:050913
           HOOKS IT INTO STACKHEAD. THE PREVIOUS STACKHEAD LINK                                         13296000 T  0003:050913
           IS SAVED IN THE LINK OF THE ELBAT WORD IN THE NEW ENTRY                                      13297000 T  0003:050913
           E PREVENTS AN ENTRY FROM OVERFLOWING A ROW,STARTING AT THE                                   13298000 T  0003:050913
           BEGINNING OF THE NEXT ROW IF NECESSARY                                          ;            13299000 T  0003:050913
     BEGIN                                                                                              13300000 T  0003:050913
        REAL WORDCOUNT,RINX;                                                                            13301000 T  0003:050913
                                                                                                 START OF SEGMENT **********   92
STACK(F+2) = WORDCOUNT
STACK(F+3) = RINX
                IF RINX←(NEXTINFO+WORDCOUNT←(COUNT+18)DIV 8     ).LINKR ≠                               13302000 T  0092:000010
                     NEXTINFO.LINKR                                                                     13303000 T  0092:000312
           THEN BEGIN PUT(125&(RINX×256-NEXTINFO)[27:40:8],NEXTINFO);                                   13304000 T  0092:000312
                NEXTINFO←256×RINX END;                                                                  13305000 T  0092:000713
                IF SPECTOG THEN                                                                         13305100 T  0092:000912
                IF NOT MACROID THEN                                                                     13305200 T  0092:000912
                     UNHOOK;                                                                            13305300 T  0092:001010
                                                                                                        13306000 T  0092:001112
                ACCUM[0].INCR←WORDCOUNT;                                                                13307000 T  0092:001112
                IF NOT INLINETOG OR MACROID THEN BEGIN                                                  13307500 T  0092:001313
                 ACCUM[0].LINK ←STACKHEAD[SCRAM];STACKHEAD[SCRAM]←NEXTINFO;                             13308000 T  0092:001512
                                END;                                                                    13308500 T  0092:001912
                 ACCUM[1].PURPT←NEXTINFO-LASTINFO;                                                      13309000 T  0092:001912
           MOVE(WORDCOUNT,ACCUM,INFO[NEXTINFO.LINKR,NEXTINFO.LINKC]);                                   13310000 T  0092:002210
                IF XREF THEN % MAKE DECLARATION REFERENCE                                %110-          13310050 C  0092:002513
           %     IF (ACCUM[0].CLASS ≠ DEFINEDID OR NOT                                   %110-          13310075 C  0092:002611
           %         BOOLEAN(ACCUM[0].FORMAL)) THEN % NOT DEFINE PARAMETER               %110-          13310080 C  0092:002611
                 BEGIN                                                                   %110-          13310100 C  0092:002611
                   XREFINFO[NEXTINFO] !=                                                 %110-          13310200 C  0092:002712
                     IF SPECTOG THEN                                                     %110-          13310300 C  0092:002913
                       XREFINFO[ELBAT[I]]                                                %110-          13310350 C  0092:003010
                     ELSE                                                                %110-          13310400 C  0092:003313
                       ((XLUN != XLUN + 1) & SGNO SEGNOF);                               %110-          13310450 C  0092:003410
                   IF SPECTOG THEN % JUST GO BACK AND FIX UP XREF ENTRY                  %110-          13310500 C  0092:003712
                     XMARK(DECLREF)                                                      %110-          13310525 C  0092:003712
                   ELSE                                                                  %110-          13310550 C  0092:004210
                     XREFIT(NEXTINFO,CARDNUMBER,IF PTOG AND NOT STREAMTOG                %110-          13310575 C  0092:004210
                             THEN NORMALREF ELSE DECLREF);                               %110-          13310580 C  0092:004210
```

```
          END
%         ELSE % DEFINE PARAMETERS - DONT CROSS REF.                    %110-
%           XREFINFO[NEXTINFO] := 0                                     %110-
          ELSE                                                          %110-
            IF DEFINING.[1:1]  THEN % WE ARE DOING XREFING              %110-
              XREFINFO[NEXTINFO] := 0;                                  %110-
            LASTINFO:=NEXTINFO;
            NEXTINFO:=NEXTINFO+WORDCOUNT
 END;
```

```
                                                                                13310600 C  00921004712
                                                                                13310700 C  00921004712
                                                                                13310750 C  00921004712
                                                                                13310800 C  00921004712
                                                                                13310900 C  00921004712
                                                                                13310950 C  00921004811
                                                                                13311000 T  00921005211
                                                                                13312000 T  00921005312
                                                                                13313000 T  00921005313
                                                                      92 IS   57 LONG. NEXT SEG    3
```

```
PROCEDURE ENTRY(TYPE);
          VALUE TYPE;
          REAL TYPE;
COMMENT
       ENTRY ASSUMES THAT I IS POINTING AT AN IDENTIFIER WHICH
       IS BEING DECLARED AND MAKES UP THE ELBAT ENTRY FOR IT
       ACCORD TO TYPE          .IF THE ENTRY IS AN ARRAY AND NOT
       A SPECIFICATION THEN A DESCRIPTOR IS PLACED IN THE STACK
       FOR THE UPCOMING COMMUNICATE TO GET STORAGE FOR THE ARRAY(S)    ;
 BEGIN
     J:=0;I:=I-1;
     DO
       BEGIN
            STOPDEFINE :=TRUE; STEPIT; SCATTERELBAT;
            IF FORMALF:=SPECTOG
               THEN
                BEGIN
                 IF ELCLASS#SPCRET
                    THEN FLAG(002);
                         BUP:=BUP+1
;     KLASSF:=TYPE;MAKEUPACCUM; E;J:=J+1;
                END
               ELSE
                BEGIN
                 IF ELCLASS>IDMAX AND ELCLASS≤MULOP
               THEN IF ELCLASS= POLYSHV THEN ELCLASS:=TYPE ELSE FLAG(3);
                 IF LEVELF=LEVEL
                   THEN FLAG(001);
                 VONF:=P2;
                 FORMALF:=PTOG;
            IF XREF AND NOT SPECTOG THEN % ERASE PREVIOUS XREF ENTRY.
               XREFPT:=XREFPT-REAL(ELBAT[I]#0); % GET RID OF LAST CREF
       KLASSF:=TYPE; MAKEUPACCUM;E; J:=J+1;
       IF ((FORMALF:=PTOG) OR(STREAMTOG AND NOT STOPGSP)) AND NOT P2
         THEN ADDRSF:=PJ:=PJ+1
         ELSE IF STOPGSP
                THEN ADDRSF:=0
                     ELSE ADDRSF:=GETSPACE(P2,LASTINFO+1);
         PUT(TAKE(LASTINFO)& ADDRSF[16:37:11],LASTINFO);
       END END

        UNTIL STEPI#COMMA OR STOPENTRY; GTA1[0]:=J
 END;
```

```
                                                                                13314000 T  00031050913
                                                                                13315000 T  00031050913
                                                                                13316000 T  00031050913
                                                                                13317000 T  00031050913
                                                                                13318000 T  00031050913
                                                                                13319000 T  00031050913
                                                                                13320000 T  00031050913
                                                                                13321000 T  00031050913
                                                                                13322000 T  00031050913
                                                                                13323000 T  00031050913
                                                                                13324000 T  00031050913
                                                                                13325000 T  00031051210
                                                                                13326000 T  00031051210
                                                                                13327000 T  00031051210
                                                                                13328000 T  00031051313
                                                                                13329000 T  00031051313
                                                                                13330000 T  00031051411
                                                                                13331000 T  00031051512
                                                                                13332000 T  00031051512
                                                                                13333000 T  00031051712
                                                                                13333500 T  00031051712
                                                                                13334000 T  00031052112
                                                                                13335000 T  00031052112
                                                                                13336000 T  00031052112
                                                                                13337000 P  00031052113
                                                                                13338000 T  00031052211
                                                                                13339000 T  00031052712
                                                                                13340000 T  00031052713
                                                                                13341000 T  00031052912
                                                                                13341100 T  00031053010
                                                                                13341200 C  00031053011
                                                                                13341300 C  00031053210
                                                                                13342000 T  00031053411
                                                                                13343000 T  00031053713
                                                                                13344000 T  00031053913
                                                                                13345000 T  00031054112
                                                                                13346000 T  00031054312
                                                                                13347000 T  00031054410
                                                                                13348000 T  00031054712
                                                                                13349000 T  00031054913
                                                                                13350000 T  00031054913
                                                                                13351000 T  00031054913
                                                                                13352000 T  00031055210
```

```
                  PROCEDURE UNHOOK;                                                                    13353000 T  0003:055:2
                COMMENT                                                                                13354000 T  0003:055:2
                   UNHOOK ASSUMES THAT THE WORD IN ELBAT[I] POINTS TO A PSUEDO ENTRY                   13355000 T  0003:055:2
                   FOR APARAMETER,ITS JOB IS TO UNHOOK THAT FALSE ENTRY SO THAT                        13356000 T  0003:055:2
                   E WILL WORK AS NORMAL.                                                    ;         13357000 T  0003:055:2
                  BEGIN                                                                                13358000 T  0003:055:2
                   REAL LINKT,A,LINKP;                                                                 13359000 T  0003:055:2
                                                                                          START OF SEGMENT **********   93
STACK(F+2) = LINKT
STACK(F+3) = A
STACK(F+4) = LINKP
                   LABEL L;                                                                            13360000 T  0093:000:0
                         LINKT←STACKHEAD[SCRAM]     ; LINKP←ELBAT[I].LINK;                           13361000 T  0093:000:0
                         IF LINKT=LINKP THEN STACKHEAD[SCRAM]←TAKE(LINKT).LINK                       13362000 T  0093:000:1
                                        ELSE                                                         13363000 T  0093:000:3
                       L: IF A←TAKE(LINKT).LINK=LINKP                                                13364000 T  0093:000:0
                          THEN PUT((TAKE(LINKT))&(TAKE(A))[35:35:13],LINKT)                          13365000 T  0093:000:1
                          ELSE BEGIN LINKT←A; GO TO L END;                                           13366000 T  0093:001:0
                   END;                                                                              13367000 T  0093:001:1
                                                                                          93 IS    17 LONG, NEXT SEG    3

                  PROCEDURE MAKEUPACCUM;                                                               13368000 T  0003:055:2
                    BEGIN                                                                              13369000 T  0003:055:2
                       IF PTOG                                                                         13370000 T  0003:055:2
                         THEN GT1←LEVELF ELSE GT1←LEVEL;                                               13371000 T  0003:055:0
                       ACCUM[0]← ABS(ELBAT[I] & KLASSF[2:41:7] & REAL(FORMALF)[9:47:1]               13372000 T  0003:055:1
                                  & REAL(VONF)[10:47:1] & GT1[11:43:5] &ADDRSF[16:37:11]             13373000 T  0003:055:3
                                  )                                                                  13374000 T  0003:056:0
                    END;                                                                               13375000 T  0003:056:1


                  PROCEDURE ARRAE;                                                                     13376000 T  0003:056:3
PRT(704) = ARRAE
                   BEGIN                                                                               13377000 T  0003:056:3
                       INTEGER SAVEINFO;                                                               13378000 T  0003:056:3
                                                                                          START OF SEGMENT **********   94
STACK(F+2) = SAVEINFO
                   LABEL BETA1;                                                                        13379000 T  0094:000:0
                         TYPEV←REALARRAYID;                                                            13380000 T  0094:000:0
                          IF T1←GTA1[J←J-1]=0 THEN J←J+1                                               13381000 T  0094:000:1
                                              ELSE                                                     13382000 T  0094:000:4
                                                IF T1=OWNV THEN                                        13383000 T  0094:000:5
                                                      BEGIN                                            13384000 T  0094:000:6
                                                           P2←TRUE;IF SPECTOG THEN                     13385000 T  0094:000:6
                                                           FLAG(13)                                    13386000 T  0094:000:7
                                                     END                                               13387000 T  0094:000:8
                                                       ELSE                                            13388000 T  0094:000:9
                                                    TYPEV←REALARRAYID+T1-REALV;                        13389000 T  0094:000:9
```

```
 BETA1: ENTER(TYPEV);
          IF ELCLASS≠LFTBRKET THEN FLAG(16);
          IF STEPI=LITNO THEN
              BEGIN
                 SAVEINFO←ELBAT[I].ADDRESS;
                 IF STEPI≠RTBRKET THEN FLAG(53);
                                          FILLSTMT(SAVEINFO);
SAVEINFO←1;
              END
                 ELSE
             BEGIN IF ELCLASS≠ASTRISK THEN FLAG(56);
                   SAVEINFO←1;
                 WHILE STEPI≠RTBRKET DO
                      BEGIN IF ELCLASS≠COMMA AND
                            STEPI≠ASTRISK THEN FLAG(56);
                            SAVEINFO←SAVEINFO+1
                      END;STEPIT;

END;       PUT(TAKE(LASTINFO)&SAVEINFO[27:40:8],LASTINFO);
J ← 1 ;      GTA1[0] ← 0     ;
IF ELCLASS=COMMA THEN BEGIN STEPIT;GO TO BETA1 END
 END ARRAE;
```

```
 PROCEDURE PUTNBUMP(X);
             VALUE X;
             REAL X;
  BEGIN
        INFO[NEXTINFO,LINKR,NEXTINFO,LINKC]←X;
        NEXTINFO←NEXTINFO+1
  END ;
```

```
 PROCEDURE JUMPCHKX;
COMMENT THIS PROCEDURE IS CALLED AT THE START OF ANY EXECUTABLE CODE
        WHICH THE BLOCK MIGHT EMIT.IT DETERMINES WHETHER ANY JUMPS
        ARROUND NONEXECUTABLE CODE MAY BE WAITING AND WHETHER IT
        IS THE FIRST EXECUTABLE CODE;
IF NOT SPECTOG THEN
BEGIN
   IF AJUMP
       THEN
   BEGIN ADJUST;
         EMITB(BFW,SAVEL,L)
   END ELSE
         IF FIRSTX=4095
           THEN
             BEGIN
               ADJUST;
               FIRSTX←L;
             END;
       AJUMP←FALSE
```

```
13390000 T  0094:0011:2
13391000 T  0094:0012:1
13392000 T  0094:0014:1
13393000 T  0094:0015:3
13394000 T  0094:0016:0
13395000 T  0094:0017:3
13396000 T  0094:0020:0
13397000 T  0094:0020:1
13398000 T  0094:0021:3
13399000 T  0094:0021:3
13400000 T  0094:0021:3
13401000 T  0094:0024:0
13402000 T  0094:0024:1
13403000 T  0094:0026:1
13404000 T  0094:0027:2
13405000 T  0094:0029:3
13406000 T  0094:0030:0
13407000 T  0094:0032:0
13408000 T  0094:0032:0
13408500 T  0094:0034:1
13409000 T  0094:0036:1
13410000 T  0094:0038:1
94 IS   41 LONG, NEXT SEG    3
```

```
13589000 T  0003:0563:3
13590000 T  0003:0563:3
13591000 T  0003:0563:3
13592000 T  0003:0563:3
13593000 T  0003:0563:3
13594000 T  0003:0567:2
13595000 T  0003:0567:2
```

```
13596000 T  0003:0568:1
13597000 T  0003:0568:1
13598000 T  0003:0568:1
13599000 T  0003:0568:1
13600000 T  0003:0568:1
13601000 T  0003:0568:1
13602000 T  0003:0569:3
13603000 T  0003:0570:0
13604000 T  0003:0570:0
13605000 T  0003:0570:0
13606000 T  0003:0571:2
13607000 T  0003:0572:0
13608000 T  0003:0572:1
13609000 T  0003:0573:2
13610000 T  0003:0573:3
13611000 T  0003:0574:0
13612000 T  0003:0574:1
13613000 T  0003:0575:3
13614000 T  0003:0575:3
```

```
END;                                                                                    13615000 T  0003:0575:3


 PROCEDURE JUMPCHKNX;                                                                   13616000 T  0003:0578:0
COMMENT JUMPCHKNX DETERMINES WHETHER ANY EXECUTABLE CODE HAS BEEN                       13617000 T  0003:0578:0
        EMITTED AND IF SO WHETHER IT WAS JUST PREVIOUS TO THE                           13618000 T  0003:0578:0
        NON EXECUTABLE ABOUT TO BE EMITTED.IF BOTH THEN L IS BUMPED                     13619000 T  0003:0578:0
        AND SAVED FOR A LATER BRANCH;                                                   13620000 T  0003:0578:0
IF NOT SPECTOG THEN                                                                     13621000 T  0003:0578:0
BEGIN                                                                                   13622000 T  0003:0578:1
   IF FIRSTX≠4095                                                                       13623000 T  0003:0579:2
     THEN                                                                               13624000 T  0003:0579:2
      BEGIN                                                                             13625000 T  0003:0579:3
       IF NOT AJUMP                                                                     13626000 T  0003:0580:0
         THEN                                                                           13627000 T  0003:0580:0
            SAVEL←BUMPL;                                                                13628000 T  0003:0580:1
      AJUMP←TRUE                                                                        13629000 T  0003:0583:2
      END;ADJUST                                                                        13630000 T  0003:0583:2
END;                                                                                    13631000 T  0003:0583:3


PROCEDURE SEGMENTSTART(SAVECODE);VALUE SAVECODE;BOOLEAN SAVECODE;                       13632000 T  0003:0586:0
    BEGIN                                                                               13632100 T  0003:0586:0
    STREAM PROCEDURE PRINT(SAVECODE,ADR,FIEL); VALUE SAVECODE,ADR;                      13633000 T  0003:0586:0
                                                                        START OF SEGMENT **********   95
PRT(705) = PRINT
                    BEGIN                                                               13634000 T  0095:0000:0
                    LABEL L1;                                                           13635000 T  0095:0000:0
                    DI:=FIEL; DS:=8 LIT" ";                                             13636000 T  0095:0000:0
                    SI:=FIEL; DS:=9 WDS; DI:=DI-3;                                      13637000 T  0095:0001:3
                    SAVECODE(DS:=38 LIT "START OF SAVE SEGMENT; BASE ADDRESS = ";       13638000 T  0095:0002:0
                      JUMP OUT TO L1);                                                  13639000 T  0095:0008:0
                    DS:=38 LIT " START OF REL SEGMENT; DISK ADDRESS = ";                13640000 T  0095:0009:2
               L1:                                                                      13641000 T  0095:0014:0
                    SI:=LOC ADR; DS:=5 DEC;                                             13642000 T  0095:0014:0
                    END PRINT;                                                          13643000 T  0095:0014:1


        MOVE(1,SAVECODE,CODE(0));                                                       13651000 T  0095:0014:1
        IF SAVECODE AND INTOG AND NOT DECKTOG THEN FLAG(57);                            13651100 T  0095:0019:2
        IF LISTER OR SEGSTOG THEN                                                       13652000 T  0095:0022:1
             BEGIN                                                                      13652500 T  0095:0024:0
             PRINT(SAVECODE,IF SAVECODE THEN CORADR ELSE DISKADR,LINE[*]);              13653000 T  0095:0024:1
             IF NOHEADING THEN DATIME; WRITELINE;                                       13653500 T  0095:0027:3
             END;                                                                       13654000 T  0095:0039:2
        END SEGMENTSTART;                                                               13655000 T  0095:0039:2
                                                                        95 IS   40 LONG, NEXT SEG    3


PROCEDURE SEGMENT(SIZE,FR); VALUE SIZE,FR; INTEGER SIZE,FR;                             13657000 T  0003:0586:0
```

```
                    BEGIN
                    STREAM PROCEDURE PRINT(SIZE,FIEL); VALUE SIZE;                                  13660000 T  0003:058:0
                                                                                                    13661000 T  0003:058:0
                                                                                          START OF SEGMENT *********   96
PRT(706) = PRINT
                         BEGIN                                                                      13663000 T  0096:000:0
                         DI:=FIEL; DS:=8 LIT" ";                                                    13665000 T  0096:000:0
                         SI:=FIEL; DS:=14 WDS;                                                      13667000 T  0096:001:3
                         DI:=DI-16; DS:=6 LIT"SIZE= ";                                              13668000 T  0096:002:0
                         SI:=LOC SIZE; DS:=4 DEC; DS:=6 LIT" WORDS"                                 13670000 T  0096:003:2
                         END PRINT;                                                                 13673000 T  0096:004:1


                    STREAM PROCEDURE DOIT(C,A,I,S,F,W); VALUE C,A,F,W;                              13673100 T  0096:005:2
PRT(707) = DOIT
                         BEGIN  LOCAL N;                                                            13673150 T  0096:005:2
                         DI:=S; DS:=8 LIT" "; SI:=S; DS:=9 WDS;                                     13673200 T  0096:005:2
                         DI:=DI-8; SI:=LOC W; DS:=4 DEC;                                            13673250 T  0096:007:2
                         SI:=I; SI:=SI+10;DI:=LOC N; DI:=DI+7; DS:=CHR;                             13673300 T  0096:007:3
                         DI:=S; SI:=LOC F; SI:=SI+7; DS:=CHR; SI:=LOC C;                            13673350 T  0096:009:2
                         DS:=3 DEC; DS:=4 DEC;SI:=I; SI:=SI+11;DS:=N CHR;                           13673400 T  0096:010:0
                         END DOIT;                                                                  13673450 T  0096:011:3


                    IF LISTER OR SEGSTOG THEN                                                       13674000 T  0096:012:0
                         BEGIN                                                                      13674500 T  0096:013:2
                         PRINT(SIZE,LIN[*]);                                                        13675000 T  0096:013:3
                         IF NOHEADING THEN DATIME; WRITELINE;                                       13676000 T  0096:015:2
                         END;                                                                       13677000 T  0096:026:1
                    IF STUFFTOG THEN IF FR>0 THEN IF LEVEL>1 THEN                                   13677100 T  0096:026:1
                         BEGIN                                                                      13677150 T  0096:029:3
                         KLASSF:=TAKE(PROINFO),CLASS;                                               13677200 T  0096:030:0
                        IF FR > 1024 THEN FR←FR-1024;                                               13677250 T  0096:032:0
                         DOIT(KLASSF,FR,INFO[PROINFO,LINKR,PROINFO,LINKC],                          13677300 T  0096:034:1
                              TWXA[0],SAF,SIZF);                                                    13677400 T  0096:037:3
                         WRITE(STUFF,10,TWXA[*]);                                                   13677500 T  0096:039:2
                         END;                                                                       13677600 T  0096:043:2
                    IF SIZE>SEGSIZEMAX THEN SEGSIZEMAX:=SIZE;                                       13678000 T  0096:043:2
                    END SEGMENT;                                                                    13681000 T  0096:045:2
                                                                                          96 IS    47 LONG, NEXT SEG    3


                  STREAM PROCEDURE MOVECODE(EDOC,TEDOC);                                            13683000 T  0003:058:0
PRT(710) = MOVECODE
                   BEGIN LOCAL T1,T2,T3;                                                            13684000 T  0003:058:0
                     SI←EDOC;T1←SI;                                                                 13685000 T  0003:058:0
                     SI←TEDOC;T2←SI;                                                                13686000 T  0003:058:1
                     SI←LOC EDOC;                                                                   13687000 T  0003:058:2
                     SI←SI+3;                                                                       13688000 T  0003:058:2
                     DI←LOC T3;                                                                     13689000 T  0003:058:3
                     DI←DI+5;                                                                       13690000 T  0003:058:3
                     SKIP 3 DB;                                                                     13691000 T  0003:058:0
```

```
15(IF SB THEN DS← 1 SET ELSE DS←1 RESET;SKIP 1 SB);                    13692000 T  0003:0588:0
  SI← LOC EDOC;                                                        13693000 T  0003:0590:0
  DI← LOC T2;                                                          13694000 T  0003:0590:1
  DS← 5 CHR;                                                           13695000 T  0003:0590:1
  3(IF SB THEN DS←1 SET ELSE DS←1 RESET;SKIP 1 SB);                    13696000 T  0003:0591:2
  DI←T3;                                                               13697000 T  0003:0593:2
  SI←LOC T2;                                                           13698000 T  0003:0593:2
  DS←WDS;                                                              13699000 T  0003:0593:3
  DI←LOC T3;                                                           13700000 T  0003:0593:3
  DI←DI+5;                                                             13701000 T  0003:0594:0
  SKIP 3 DB;                                                           13702000 T  0003:0594:0
  SI←LOC TEDOC;                                                        13703000 T  0003:0594:1
  SI←SI+3;                                                             13704000 T  0003:0594:1
15(IF SB THEN DS←1 SET ELSE DS← 1 RESET;SKIP 1 SB);                    13705000 T  0003:0595:2
  SI← LOC TEDOC;                                                       13706000 T  0003:0597:2
  DI← LOC T1;                                                          13707000 T  0003:0597:2
  DS← 5 CHR;                                                           13708000 T  0003:0597:3
 3(IF SB THEN DS←1 SET ELSE DS←1 RESET;SKIP 1 SB);                     13709000 T  0003:0597:3
  DI←T3;                                                               13710000 T  0003:0599:3
  SI←LOC T1;                                                           13711000 T  0003:0600:0
  DS←WDS                                                               13712000 T  0003:0600:0
END;                                                                   13713000 T  0003:0600:1


PROCEDURE ENTER(TYPE);                                                 13714000 T  0003:0600:1
                VALUE TYPE;                                            13715000 T  0003:0600:1
                REAL TYPE;                                             13716000 T  0003:0600:1
        BEGIN                                                          13717000 T  0003:0600:1
                G←GTA1[J←J-1];                                         13718000 T  0003:0600:1
                IF NOT SPECTOG                                         13719000 T  0003:0603:2
                  THEN                                                 13720000 T  0003:0603:2
                    BEGIN                                              13721000 T  0003:0603:3
                      IF NOT P2                                        13722000 T  0003:0604:0
                         THEN IF P2←(G=OWNV)                           13723000 T  0003:0604:0
                                 THEN G←GTA1[J←J-1];                   13724000 T  0003:0605:2
                      IF NOT P3                                        13725000 T  0003:0608:1
                         THEN IF P3←(G=SAVEV)                          13726000 T  0003:0608:1
                                 THEN G←GTA1[J←J-1]                    13727000 T  0003:0610:0
                    END;                                               13728000 T  0003:0612:1
                IF G≠0 THEN FLAG(25) ELSE ENTRY(TYPE)                  13729000 T  0003:0613:3
        END;                                                           13730000 T  0003:0616:0


            PROCEDURE HTTEOAP(GOTSTORAGE,RELAD,STOPPER,PRTAD);         13731000 T  0003:0617:2
PRT(711) = HTTEOAP
                VALUE GOTSTORAGE,RELAD,STOPPER,PRTAD;                  13732000 T  0003:0617:2
                BOOLEAN GOTSTORAGE;                                    13733000 T  0003:0617:2
                REAL RELAD,STOPPER,PRTAD;                              13734000 T  0003:0617:2
         BEGIN                                                         13735000 T  0003:0617:2
                IF FUNCTOG                                             13736000 T  0003:0617:2
                  THEN                                                 13737000 T  0003:0617:2
                    BEGIN                                              13738000 T  0003:0617:2
                        EMITV(513);                                    13739000 T  0003:0617:3
```

```
                                    EMITO(RTN)                                                                  13740000 T  0003:061B:1
                                  END                                                                           13741000 T  0003:061B:1
                                ELSE                                                                            13742000 T  0003:0619:2
                                    EMITO(XIT);                                                                 13743000 T  0003:0619:2
                              CONSTANTCLEAN;                                                                    13744000 T  0003:0620:1
                              PURGE(STOPPER);                                                                   13745000 T  0003:0621:2
                       MOVE(1,CODE(0),Z); PROGDFSCBLDR(PRTAD,BOOLEAN(Z),(L+3)DIV 4,PDES);                       13746000 T  0003:0621:3
                       END HTTEOAP;                                                                             13747000 T  0003:0628:0



                    PROCEDURE INLINE;                                                                           13748000 T  0003:0628:1
                     BEGIN                                                                                      13749000 T  0003:0628:1
                      INTEGER SN,LN,P,LS,J; BOOLEAN MKST;                                                       13750000 T  0003:0628:1
                                                                                                    START OF SEGMENT **********   97
STACK(F+2) = SN
STACK(F+3) = LN
STACK(F+4) = P
STACK(F+5) = LS
STACK(F+6) = J
STACK(F+7) = MKST
                          BOOLEAN FLIPFLOP;                                                                     13750500 T  0097:0000:0
STACK(F+10) = FLIPFLOP
                            INTEGER PN;                                                                         13750600 T  0097:0000:0
STACK(F+11) = PN
                       LABEL L1,L2,L3;                                                                          13751000 T  0097:0000:0
                         PN+1 ;                                                                                 13751100 T  0097:0000:0
                              FLIPFLOP+INLINETOG+TRUE;P+0;MKST+FALSE;LS+L;EMITO(NOP);                           13752000 T  0097:0000:1
                         IF STEPI≠LEFTPAREN THEN FLAG(59);                                                      13753000 T  0097:0005:2
                         IF TABLE(I+1)=COLON THEN BEGIN STEPIT;GO TO L2 END ;                                   13753100 T  0097:0007:2
                    L1:  IF STEPI>IDMAX THEN BEGIN FLAG(465); GO TO L2 END ;                                    13754000 T  0097:0010:1
                           ACCUM[0]+08P[16:37:11]&LOCLID[2:41:7]&SCRAM[35:35:13];                               13755000 T  0097:0013:3
                              F;IF FLIPFLOP THEN BEGIN FLIPFLOP+FALSE;LN+SN+LASTINFO END;                       13755500 T  0097:0018:0
                              IF STEPI=COMMA OR ELCLASS=COLON OR ELCLASS=RTPAREN                                13756000 T  0097:0021:2
                           THEN BEGIN I+I-2;STEPIT END                                                          13757000 T  0097:0023:3
                           ELSE IF ELCLASS≠ASSIGNOP THEN FLAG(60) ELSE STEPIT;                                  13758000 T  0097:0026:1
                           AEXP;                                                                                13759000 T  0097:0030:0
                     L2:   IF ELCLASS=COLON THEN                                                                13760000 T  0097:0030:1
                           BEGIN IF MKST THEN FLAG(99); MKST+TRUE; EMITO(MKS); P+P+2;                           13761000 T  0097:0031:3
                                 IF TABLE(I+1)≠RTPAREN THEN GO TO L1; STEPIT                                    13761100 T  0097:0036:1
                                 ;PN+2;                                                                         13761110 T  0097:0038:1
                           END ELSE P+P+1;                                                                      13761200 T  0097:0040:0
                           IF ELCLASS=COMMA THEN GO TO L1;                                                      13762000 T  0097:0041:3
                           IF ELCLASS≠RTPAREN THEN FLAG(61);                                                    13763000 T  0097:0043:2
                           IF NOT MKST THEN                                                                     13764000 T  0097:0045:2
                              BEGIN J+L;L+LS;EMITO(MKS);L+J END;                                                13765000 T  0097:0045:3
                         IF STEPI ≠ SEMICOLON THEN FLAG(62);                                                    13766000 T  0097:0049:2
                                  EMITO(584);                                                                   13766100 T  0097:0051:2
                                                                                                                13766200 T  0097:0052:0
                                                                                                                13766300 T  0097:0052:0
                                                                                                                13766400 T  0097:0052:0
                                                                                                                13766500 T  0097:0052:0
                          L3:ELBAT[I]+TAKE(SN);SCATTERELBAT;ADDRSF+P-ADDRSF;                                     13767000 T  0097:0052:0
                              PUT(ELBAT[I]&ADDRSF[16:37:11]&STACKHEAD[LINKF][33:33:15],SN);                     13768000 T  0097:0055:3
                                 STACKHEAD[LINKF]+SN;  SN+SN+INCRF;                                             13769000 T  0097:0058:1
                                 IF ADDRSF≠PN THEN GO TO L3 ;                                                   13770000 T  0097:0061:2
                                 INLINETOG+  FALSE;                                                             13770500 T  0097:0062:0
```

```
          PN.NEXTINFO;                                                                  13770600 T  0097:0063:2
          STREAMTOG.TRUE;STREAMWORDS;IF STEPI#BEGINV THEN STREAMSTMT                    13771000 T  0097:0063:3
          ELSE BEGIN STEPIT;COMPOUNDTAIL END;                                           13772000 T  0097:0066:1
          STREAMTOG.FALSE;PURGE(PN);STREAMWORDS;PURGE(LN);EMITL(16);                    13773000 T  0097:0068:1
                                                                                        13773500 T  0097:0072:0
END INLINE;                                                                             13774000 T  0097:0072:0
                                                                                  97 IS   76 LONG, NEXT SEG    3



 COMMENT THIS SECTION CONTAINS THE BLOCK ROUTINE  ;                                     14000000 T  0003:0628:1
PROCEDURE BLOCK(SOP);                                                                   14001000 T  0003:0628:1
          VALUE SOP;                                                                    14002000 T  0003:0628:1
          BOOLEAN SOP;                                                                  14003000 T  0003:0628:1
COMMENT SOP IS TRUE IF THE BLOCK WAS CALLED BY ITSELF THROUGH THE                       14004000 T  0003:0628:1
        PROCEDURE DECLARATION-OTHERWISE IT WAS CALLED BY STATEMENT.                     14005000 T  0003:0628:1
        THE BLOCK ROUTINE IS RESPONSIBLE FOR HANDLING THE BLOCK                         14006000 T  0003:0628:1
        STRUCTURE OF AN ALGOL PROGRAM-SEGMENTING EACH BLOCK,HANDLING                    14007000 T  0003:0628:1
        ALL DECLARATIONS,DOING NECESSARY BOOKKEEPING REGARDING EACH                     14008000 T  0003:0628:1
        BLOCK, AND SUPPLYING THE SCANNER WITH ALL NECESSARY INFORMATION                 14009000 T  0003:0628:1
        ABOUT DECLARED IDENTIFIERS.                                                     14010000 T  0003:0628:1
        IT ALSO WRITES EACH SEGMENT ONTO THE PCT;                                       14011000 T  0003:0628:1
BEGIN                                                                                   14012000 T  0003:0628:1
 LABEL OWNERR,SAVERR,BOOLEANDEC,REALDEC,ALPHADEC,INTEGERDEC,                            14013000 T  0003:0628:1
                                                                                START OF SEGMENT **********   98
    LABELDEC,DUMPDEC,SUBDEC,OUTDEC,INDEC,MONITORDEC,                                    14014000 T  0098:0000:0
    SWITCHDEC,PROCEDUREDEC,ARRAYDEC,NAMEDEC,FILEDEC,                                    14015000 T  0098:0000:0
          GOTSCHK,FIELDDEC,AUXMEMERR,                                           %112-   14016000 P  0098:0000:0
           STREAMERR,DEFINEDEC,CALLSTATEMENT,HF,START;                                  14017000 T  0098:0000:0
         SWITCH DECLSW := OWNERR,SAVERR,BOOLEANDEC,REALDEC,INTEGERDEC,ALPHADEC,        14018000 T  0098:0000:0
                          LABELDEC,DUMPDEC,SUBDEC,OUTDEC,INDEC,MONITORDEC,           14019000 T  0098:0002:1
                          SWITCHDEC,PROCEDUREDEC,ARRAYDEC,NAMEDEC,FILEDEC,           14020000 T  0098:0002:1
                        STREAMERR,DEFINEDEC,AUXMEMERR,FIELDDEC;                 %112-   14021000 P  0098:0002:1
         DEFINE NLOCS=10#,LOCBEGIN=PRT1#,                                             14022000 T  0098:0013:3
                LBP=[36:12]#,                                                         14023000 T  0098:0013:3
                SPACEITDOWN = WRITE(LINE[PAGE])#;                               %902-   14023100 P  0098:0013:3
                                                                                      14024000 T  0098:0013:3
         BOOLEAN GOTSTORAGE;                                                          14025000 T  0098:0013:3
STACK(F+2) := GOTSTORAGE
           INTEGER PINFOO,BLKAD;                                                      14026000 T  0098:0013:3
STACK(F+3) := PINFOO
STACK(F+4) := BLKAD
              COMMENT LOCAL TO BLOCK TO SAVE WHERE A PROCEDURE IS ENTERED             14027000 T  0098:0013:3
              IN INFO;                                                                14028000 T  0098:0013:3
         REAL MAXSTACKO,LASTINFOT,RELAD,LO,TSUBLEVEL,STACKCTRO;                       14029000 T  0098:0013:3
STACK(F+5) := MAXSTACKO
STACK(F+6) := LASTINFOT
STACK(F+7) := RELAD
STACK(F+10) := LO
STACK(F+11) := TSUBLEVEL
STACK(F+12) := STACKCTRO
           INTEGER SGNOO,LOLD,SAVELO,PRTIO,NINFOO;                                    14030000 T  0098:0013:3
STACK(F+13) := SGNOO
STACK(F+14) := LOLD
STACK(F+15) := SAVELO
STACK(F+16) := PRTIO
```

```
STACK(F+17) = NINFOO
                INTEGER NCIIO;                                                  14031000 T  0098:0013:3
STACK(F+20) = NCIIO
                INTEGER PROAD ;                                                 14032000 T  0098:0013:3
STACK(F+21) = PROAD
                INTEGER FIRSTXO;                                                14033000 T  0098:0013:3
STACK(F+22) = FIRSTXO
                BOOLEAN FUNCTOGO,AJUMPO;                                        14034000 T  0098:0013:3
STACK(F+23) = FUNCTOGO
STACK(F+24) = AJUMPO
                    BEGINCTR←BEGINCTR+1;                                        14035000 T  0098:0013:3
                    IF SOP                                                      14036000 T  0098:0015:2
                                THEN BEGIN BLKAD←PROADO;                        14037000 T  0098:0015:2
                            IF LASTENTRY ≠ 0                                    14038000 T  0098:0016:1
                               THEN BEGIN GT1←BUMPL;                            14039000 T  0098:0017:2
                                          CONSTANTCLEAN;                        14040000 T  0098:0019:3
                                          EMITB(BFW,GT1,L)                      14041000 T  0098:0020:0
                                    END                                         14042000 T  0098:0021:2
                        END                                                     14043000 T  0098:0021:3
                    ELSE BEGIN BLKAD:=GETSPACE(TRUE,-6); % SEG. DESCR.          14044000 T  0098:0021:3
                                                                                14045000 T  0098:0023:3
                                                                                14046000 T  0098:0023:3
                                                                                14047000 T  0098:0023:3
                         END;                                                   14048000 T  0098:0023:3
                                                                                14049000 T  0098:0023:3
                                                                                14050000 T  0098:0023:3
                  FIRSTXO←FIRSTX;                                               14051000 T  0098:0023:3
                FIRSTX←0;                                                       14052000 T  0098:0024:1
                 LEVEL←LEVEL+1;                                                 14053000 T  0098:0025:2
                    LOLD←L;FUNCTOGO←FUNCTOG;AJUMPO←AJUMP;PRTIO←PRTI;SGNOO←SGNO;  14054000 T  0098:0026:1
                    SAVELO←SAVEL;AJUMP←FALSE; L←0;NINFOO←NEXTINFO;              14055000 T  0098:0030:0
                 NCIIO←NCII;                                                    14056000 T  0098:0033:2
                 NCII←0;                                                        14057000 T  0098:0034:0
                 STACKCTRO←STACKCTR;                                            14058000 T  0098:0034:1
                                                                                14059000 T  0098:0035:3
                                                                                14061000 T  0098:0035:3
                 ELBAT[I].CLASS←SEMICOLON;                                      14062000 T  0098:0035:3
               START: IF TABLE(I)≠SEMICOLON                                     14063000 T  0098:0038:0
                        THEN                                                    14064000 T  0098:0038:1
                          BEGIN                                                 14065000 T  0098:0039:2
                            FLAG(0);                                            14066000 T  0098:0039:3
                            I←I-1                                               14067000 T  0098:0040:1
                          END;                                                  14068000 T  0098:0040:1
                      GTA1[0]←J←0;                                              14069000 T  0098:0041:3
                      IF SPECTOG                                                14070000 T  0098:0043:3
                         THEN                                                   14071000 T  0098:0043:3
                           BEGIN                                                14072000 T  0098:0043:3
                             IF BUP=PJ                                          14073000 T  0098:0044:0
                               THEN                                             14074000 T  0098:0044:1
                                 BEGIN                                          14075000 T  0098:0045:2
                                    BEGIN LABEL GETLP;                          14076000 T  0098:0045:3
PRT(712) = *SEGMENT DESCRIPTOR*
                                                                        START OF SEGMENT **********   99
                                                IF STREAMTOG THEN F←0 ELSE      14077000 T  0099:0000:0
                                   F←FZERO;                                     14078000 T  0099:0001:3
                                 BUP←LASTINFO;                                  14079000 T  0099:0002:1
                                    DO                                          14080000 T  0099:0003:3
```

```
                                    BEGIN                                                                14081000 T  00991000410
                                  IF NOT STREAMTOG THEN                                                  14082000 T  00991000410
                                    BUP←LASTINFO;                                                        14083000 T  00991000411
                                    GETLP: G←TAKE(BUP);                                                  14084000 T  00991000513
                                           IF K←G.ADDRESS≠PJ                                             14085000 T  00991000712
                                             THEN                                                        14086000 T  00991000811
                                               BEGIN                                                     14087000 T  00991000912
                IF BUP ≠ BUP:=BUP- TAKE(BUP + 1).PURPT THEN                                    %102-       14088000 P  00991000913
                                                 GO TO GETLP                                             14089000 T  00991001211
                                               END;                                                      14090000 T  00991001211
                 TYPEV←G.CLASS;                                                                          14091000 T  00991001312
                                                   G.ADDRESS←F←F+1;                                      14115000 T  00991001411
                                              PUT(G.BUP); G.INCR←GT1;                                    14116000 T  00991001712
                                              PUT(G.MARK←PJ)                                             14117000 T  00991002010
                                             ;BUP←BUP-TAKE(BUP+1).PURPT                                  14118000 T  00991002011
                                    END                                                                  14119000 T  00991002313
                                   UNTIL PJ←PJ-1=0                                                       14120000 T  00991002410
                                  END;                                                                   14121000 T  00991002513
PRT(713) = *SEGMENT DESCRIPTOR*

                                                                                                 99 IS   27 LONG, NEXT SEG   98
                                      SPECTOG←FALSE;                                             14122000 T  00981004610
                                  GO TO HF                                                       14123000 T  00981004611
                                 END                                                             14124000 T  00981004712
                            END;                                                                 14125000 T  00981004712
                              STACKCT ← 0;                                                       14125500 T  00981004712
                         WHILE STEP:=DECLARATORS                                                 14126000 T  00981004810
                         DO                                                                      14127000 T  00981004811
                           BEGIN                                                                 14128000 T  00981004913
                             GTA1[J←J+1]←ELBAT[I].ADDRESS;                                       14129000 T  00981004913
                             STOPDEFINE←ERRORTOG←TRUE;                                           14130000 T  00981005211
                           END;                                                                  14131000 T  00981005313
                IF J =0 THEN GO TO CALLSTATEMENT;                                                14132000 T  00981005410
                    P2←P3←FALSE;                                                                 14133000 T  00981005513
                    GO TO DECLSW[GTA1[J]];                                                       14134000 T  00981005611
               OWNERR:FLAG(20);J←J+1;GO TO REALDEC;                                              14135000 T  00981005912
               SAVERR:FLAG(21);J←J+1;GO TO REALDEC;                                              14136000 T  00981006113
               AUXMEMERR:FLAG(618);J:=J+1;GO TO REALDEC;                                 %112-   14136100 C  00981006411
               STREAMERR:      IF ELCLASS = LEFTPAREN THEN                                % 6    14137000 T  00981006713
                         BEGIN                                                            % 6    14137100 T  00981006811
                               I ← I - 1;                                                 % 6    14137200 T  00981006912
                               GO TO CALLSTATEMENT;                                       % 6    14137300 T  00981007011
                         END;                                                             % 6    14137400 T  00981007112
                               FLAG(22);                                                  % 6    14137500 T  00981007112
                               J ← J + 1;                                                 % 6    14137600 T  00981007113
                               GO TO PROCEDUREDEC;                                        % 6    14137700 T  00981007312
               REALDEC:P3←TRUE;ENTER(REALID);GO TO START;                                         14138000 T  00981007313
               ALPHADEC:P3←TRUE;ENTER(ALFAID);GO TO START;                                        14139000 T  00981007610
               BOOLEANDEC:P3←TRUE;ENTER(BOOID);GO TO START;                                       14140000 T  00981007810
               INTEGERDEC:P3←TRUE;ENTER(INTID);GO TO START;                                       14141000 T  00981008010
                       MONITORDEC:IF SPECTOG                                                     14142000 T  00981008210
                                    THEN BEGIN COMMENT ERROR 463 MEANS THAT A MONITOR            14143000 T  00981008210
                                               DECLARATION APPEARS IN THE SPECIFICATION          14144000 T  00981008211
                                               PART OF A PROCEDURE;                              14145000 T  00981008211
                                         FLAG(463);                                              14146000 T  00981008211
                                       END;                                                      14147000 T  00981008313
                                 DO UNTIL FALSE;                                                 14148000 T  00981008313
                      DUMPDEC:IF SPECTOG                                                         14149000 T  00981008411
```

```
                        THEN BEGIN COMMENT ERROR 464 MEANS A DUMP DECLARATION
                                   APPEARS IN THE SPECIFICATION PART OF A
                                   PROCEDURE;
                                  FLAG(464);
                             END;
                        DO UNTIL FALSE;
                ARRAYDEC: ARRAE; GO TO START;
                FILEDEC: INDEC: OUTDEC:
                GOTSCHK:GOTSTORAGE← NOT SPECTOG OR GOTSTORAGE;GO TO START;
                NAMEDEC: IF T1←GTA1[J←J-1]≠ARRAYV THEN J←J+1;
                         TYPEV←NAMEID;
                         IF T1←GTA1[J←J-1]=0 THEN J←J+1
                                             ELSE
                                          IF T1=OWNV
                                            THEN
                                               BEGIN
                                                 P2←TRUE; IF SPECTOG THEN
                                                                FLAG(013)
                                               END
                                            ELSE

                                                 TYPEV←NAMEID+T1-REALV;
                             ENTER(TYPEV); GO TO START;
                SUBDEC:
                    BEGIN REAL TYPEV,T;
PRT(714) = *SEGMENT DESCRIPTOR*

STACK(F+25) = TYPEV
STACK(F+26) = T
                         IF GTA1[J←J-1]=REALV THEN TYPEV←REALSUBID ELSE TYPEV←SUBID;
                    STOPGSP←TRUE;
                        JUMPCHKNX;ENTRY(TYPEV);IF ELCLASS≠SEMICOLON THEN FLAG(57);
                    STOPGSP←FALSE;
                         STEPIT;
                              T←NEXTINFO;
                    PUTNBUMP(L); STMT; EMITO(LFU);     IF TYPEV=REALSUBID THEN
                              IF GET(L-2)≠533 THEN FLAG(58);PUT(TAKE(T)&L[24:36:12],T);
                    CONSTANTCLEAN;
                        END;
PRT(715) = *SEGMENT DESCRIPTOR*

                         GO TO START;
```

```
14150000 T  0098:0085:2
14151000 T  0098:0085:3
14152000 T  0098:0085:3
14153000 T  0098:0085:3
14154000 T  0098:0086:1
14155000 T  0098:0086:1
14156000 T  0098:0087:3
14158000 T  0098:0089:2
14160000 T  0098:0089:2
14161000 T  0098:0091:2
14161010 T  0098:0095:2
14161020 T  0098:0096:0
14161030 T  0098:0099:2
14161040 T  0098:0100:0
14161050 T  0098:0101:2
14161060 T  0098:0101:3
14161070 T  0098:0102:0
14161080 T  0098:0103:2
14161090 T  0098:0103:3
14161100 T  0098:0104:0
14161110 T  0098:0104:0
14161120 T  0098:0104:0
14162000 T  0098:0106:1
14163000 T  0098:0107:3
14163500 T  0098:0108:0

START OF SEGMENT **********  100

14164000 T  0100:0000:0
14164500 T  0100:0004:1
14165000 T  0100:0005:2
14166    T  0100:0008:1
14166000 T  0100:0009:2
14166500 T  0100:0009:3
14167000 T  0100:0010:1
14168000 T  0100:0013:2
14168500 T  0100:0019:2
14169000 T  0100:0019:3

100 IS   21 LONG, NEXT SEG   98
14170000 T  0098:0109:2
14171000 T  0098:0109:3
14172000 T  0098:0109:3
14173000 T  0098:0109:3
14174000 T  0098:0109:3
14175000 T  0098:0109:3
14176000 T  0098:0109:3
14177000 T  0098:0109:3
14178000 T  0098:0109:3
14179000 T  0098:0109:3
14180000 T  0098:0109:3
14181000 T  0098:0109:3
14182000 T  0098:0109:3
14183000 T  0098:0109:3
14184000 T  0098:0109:3
14185000 T  0098:0109:3
```

```
                    LABELDEC;IF SPECTOG AND FUNCTOG THEN FLAG(24);
                             STOPENTRY.STOPGSP.TRUE;
                             I.I-1;
                             DO
                               BEGIN
                                  STOPDEFINE.TRUE;
                                STEPIT;
                                  ENTRY(LABELID);
                                  PUTNBUMP(0)
                               END
                             UNTIL ELCLASS#COMMA;
                             STOPENTRY.STOPGSP.FALSE;
                             GO TO START;
                    SWITCHDEC;
                      BEGIN
                        LABEL START;
PRT(716) = *SEGMENT DESCRIPTOR*

                           INTEGER GT1,GT2,GT4,GT5;
STACK(F+25) = GT1
STACK(F+26) = GT2
STACK(F+27) = GT4
STACK(F+30) = GT5
                         BOOLEAN TB1;
STACK(F+31) = TB1
                         STOPENTRY.NOT SPECTOG;STOPGSP.TRUE;
                        SCATTERELBAT; GT1.0; TB1.FALSE;
                           ENTRY(SWITCHID);
                      GT2.NEXTINFO; PUTNBUMP(0);
                        DO
                           BEGIN
                              IF STEPI#LABELID OR ELBAT[I].LVL#LEVEL THEN FLAG(63);
                              PUTNBUMP(ELBAT[I]);GT1.GT1+1
                           END
                                    COMMENT
                        UNTIL STEPI#COMMA;

                           PUT(GT1,GT2);
                            STOPENTRY . STOPGSP . FALSE;
                      END SWITCHDEC;
PRT(717) = *SEGMENT DESCRIPTOR*

                    GO TO START;
                      DEFINEDEC;
                        BEGIN LABEL START;
PRT(720) = *SEGMENT DESCRIPTOR*

                           REAL J,K;
STACK(F+25) = J
STACK(F+26) = K
                         BOOLEAN STREAM PROCEDURE PARM(S,D,K,J); VALUE K,J;
PRT(721) = PARM
                           BEGIN SI.S;SI.SI+2; DI.D;DI.DI+2;
                                IF K SC#DC THEN TALLY.1;
                                DI.LOC J;DI.DI+7;
                                IF SC#DC THEN TALLY.1;
```

```
14186000 T  0098:010:3
14187000 T  0098:010:3
14188000 T  0098:011:0
14189000 T  0098:011:2
14190000 T  0098:011:1
14191000 T  0098:011:2
14192000 T  0098:011:2
14193000 T  0098:011:3
14194000 T  0098:011:0
14195000 T  0098:011:2
14196000 T  0098:011:2
14197000 T  0098:011:3
14198000 T  0098:011:2
14199000 T  0098:012:0
14200000 T  0098:012:1
14201000 T  0098:012:2
14202000 T  0098:012:2

START OF SEGMENT **********  101
14203000 T  0101:000:0

14204000 T  0101:000:0

14205000 T  0101:000:0
14206000 T  0101:000:3
14207000 T  0101:000:3
14217000 T  0101:000:1
14218000 T  0101:000:0
14219000 T  0101:000:0
14220000 T  0101:000:0
14221000 T  0101:001:0
14222000 T  0101:001:2
14222500 T  0101:001:0
14223000 T  0101:001:0
14223500 T  0101:001:3
14224000 T  0101:001:3
14251000 T  0101:001:1
14252000 T  0101:001:0

101 IS   17 LONG, NEXT SEG   98
14253000 T  0098:012:0
14254000 T  0098:012:1
14254050 T  0098:012:2

START OF SEGMENT **********  102
14254100 T  0102:000:0

14254200 T  0102:000:0

14254300 T  0102:000:0
14254400 T  0102:000:2
14254500 T  0102:000:0
14254600 T  0102:000:1
```

```
                PARM.TALLY;                                                                    14254700 T  0102:0003:2
            END;                                                                               14254800 T  0102:0003:3



        STOPENTRY.STOPGSP.TRUE;I.I-1;                                                          14255000 T  0102:0004:1
            DEFINING := BOOLEAN(REAL(DEFINING) & 1[47:47:1]);                  %108-           14255500 C  0102:0007:3
        DO                                                                                     14256000 T  0102:0009:2
         BEGIN                                                                                 14257000 T  0102:0010:0
                STOPDEFINE.TRUE;                                                               14258000 T  0102:0010:0
            STEPIT; MOVE(9,ACCUM[1],GTA1);                                                     14259000 T  0102:0011:1
            K.COUNT+1; J.GTA1[0]; ENTRY(DEFINEDID);                                            14259010 T  0102:0013:2
            GTA1[0].J.-100000-; J.0;                                                           14259015 T  0102:0016:0
            IF ELCLASS=LEFTPAREN OR ELCLASS=LFTBRKET THEN                                      14259020 T  0102:0018:1
            BEGIN                                                                              14259030 T  0102:0020:0
                DO BEGIN STOPDEFINE.TRUE;                                                      14259060 T  0102:0020:1
                    STEPIT;                                                                    14259070 T  0102:0021:3
                    IF (J.J+1)>9 OR PARM(ACCUM[1],GTA1,K,J) OR                                 14259080 T  0102:0022:0
                        K>62 THEN BEGIN ERR(141); GO TO START END;                             14259090 T  0102:0026:1
                    STOPDEFINE.TRUE;                                                           14259100 T  0102:0031:2
                END UNTIL STEPI#COMMA;                                                         14259110 T  0102:0031:3
                IF ELCLASS#RTPAREN AND ELCLASS#RTBRKET THEN ERR(141);                          14259120 T  0102:0033:2
                STOPDEFINE.TRUE;                                                               14259130 T  0102:0036:0
                STEPIT;                                                                        14259140 T  0102:0037:2
                PUT(TAKE(LASTINFO)&J[16:37:11],LASTINFO);                                      14259150 T  0102:0037:3
            END;                                                                               14259160 T  0102:0040:0
           IF ELCLASS#RELOP                                                                    14260000 T  0102:0040:0
             THEN                                                                              14261000 T  0102:0040:0
                BEGIN                                                                          14262000 T  0102:0040:1
                  FLAG(30);                                                                    14263000 T  0102:0041:2
                  I.I-1;                                                                       14264000 T  0102:0042:0
                END;                                                                           14265000 T  0102:0043:2
             MACROID.TRUE;                                                                     14265900 T  0102:0043:2
             DEFINEGEN(FALSE,J);                                                               14266000 T  0102:0044:0
             MACROID.FALSE;                                                                    14266100 T  0102:0045:2
           END                                                                                 14267000 T  0102:0045:3
          UNTIL STEPI#COMMA;                                                                   14268000 T  0102:0045:3
             DEFINING := BOOLEAN(REAL(DEFINING) & 0[47:47:1]);                 %108-           14268500 C  0102:0047:2
        START; STOPENTRY.STOPGSP.FALSE; END; GO TO START;                                      14269000 T  0102:0049:2
PRT(722) = *SEGMENT DESCRIPTOR*

                                                                          102 IS   51 LONG, NEXT SEG   98
                                                                                              T  0098:0124:1
            FIELDDEC;                                                     %112-           14269020 C  0098:0124:1
              BEGIN                                                       %112-           14269040 C  0098:0125:2
                REAL SAVEINFO, SB, NB;                                    %112-           14269060 C  0098:0125:2
PRT(723) = *SEGMENT DESCRIPTOR*

                                                                               START OF SEGMENT **********  103

STACK(F+25) = SAVEINFO
STACK(F+26) = SB
STACK(F+27) = NB
                BOOLEAN FOUNDLB; % TRUE IF LEFT-BRACKET WAS USED IN FIELD SPEC.%112-           14269080 C  0103:0000:0
STACK(F+30) = FOUNDLB
                LABEL EXIT, SAVEIT;                                       %112-           14269100 C  0103:0000:0
                STOPENTRY := STOPGSP := TRUE;                             %112-           14269120 C  0103:0000:0
                I := I - 1;                                               %112-           14269140 C  0103:0001:2
                DO                                                        %112-           14269160 C  0103:0002:1
```

```
BEGIN                                                                    %112-      14269180 C  0103:0003:2
  STOPDEFINE := TRUE;                                                    %112-      14269200 C  0103:0003:2
  STEPIT;                                                                %112-      14269220 C  0103:0003:3
  ENTRY(FIELDID);                                                        %112-      14269240 C  0103:0004:0
  SAVEINFO := LASTINFO;                                                  %112-      14269260 C  0103:0005:2
  IF ELCLASS = RELOP AND ACCUM[1] = "1=0000" THEN                        %112-      14269280 C  0103:0005:3
    BEGIN                                                                %112-      14269300 C  0103:0007:3
      IF STEPI = LFTBRKFT THEN % REMEMBER THIS                           %112-      14269320 C  0103:0008:0
        BEGIN                                                            %112-      14269340 C  0103:0009:2
          FOUNDLB := TRUE;                                               %112-      14269360 C  0103:0009:3
          STEPIT;                                                        %112-      14269380 C  0103:0010:1
        END                                                              %112-      14269400 C  0103:0011:2
      ELSE                                                               %112-      14269420 C  0103:0011:2
        FOUNDLB := FALSE;                                                %112-      14269440 C  0103:0011:2
      IF ELCLASS = FIELDID THEN                                          %112-      14269442 C  0103:0013:3
        BEGIN                                                            %112-      14269444 C  0103:0014:1
          SB := ELBAT[I].SBITF;                                          %112-      14269446 C  0103:0015:2
          NB := ELBAT[I].NBITF;                                          %112-      14269448 C  0103:0016:1
          GO TO SAVEIT;                                                  %112-      14269450 C  0103:0018:0
        END;                                                             %112-      14269452 C  0103:0018:1
      IF ELCLASS = LITNO THEN                                            %112-      14269460 C  0103:0018:1
        IF STEPI = COLON THEN                                            %112-      14269480 C  0103:0019:2
          IF STEPI = LITNO THEN                                          %112-      14269500 C  0103:0020:1
            IF (SB := ELBAT[I-2].ADDRESS) *                              %112-      14269520 C  0103:0022:0
                (NB := ELBAT[I].ADDRESS) # 0 AND                         %112-      14269540 C  0103:0024:1
                SB + NB ≤ 48 THEN                                        %112-      14269560 C  0103:0027:2
              BEGIN                                                      %112-      14269580 C  0103:0028:1
      SAVEIT:                                                            %112-      14269590 C  0103:0029:2
                PUT(TAKE(SAVEINFO) & SB SBITF & NB NBITF,                %112-      14269600 C  0103:0029:2
                  SAVEINFO);                                             %112-      14269620 C  0103:0032:0
                STEPIT;                                                  %112-      14269640 C  0103:0032:1
                IF FOUNDLB THEN % BETTER HAVE RIGHT BRACKET.             %112-      14269660 C  0103:0033:2
                  IF ELCLASS = RTBRKET THEN                              %112-      14269680 C  0103:0033:2
                    BEGIN                                                %112-      14269700 C  0103:0034:1
                      STEPIT;                                            %112-      14269705 C  0103:0035:2
                      GO TO EXIT;                                        %112-      14269710 C  0103:0035:3
                    END                                                  %112-      14269715 C  0103:0036:0
                  ELSE % MISSING RIGHT BRACKET.                          %112-      14269720 C  0103:0036:0
                ELSE                                                     %112-      14269740 C  0103:0036:0
                  GO TO EXIT;                                            %112-      14269760 C  0103:0036:1
              END;                                                       %112-      14269780 C  0103:0036:1
            FLAG(114);                                                   %112-      14269800 C  0103:0036:1
            DO STEPIT UNTIL ELCLASS = COMMA OR ELCLASS = SEMICOLON;      %112-      14269820 C  0103:0036:1
        EXIT:                                                            %112-      14269840 C  0103:0037:2
          END                                                            %112-      14269860 C  0103:0040:1
        UNTIL                                                            %112-      14269880 C  0103:0041:2
          ELCLASS # COMMA;                                               %112-      14269900 C  0103:0041:2
        STOPENTRY := STOPGSP := FALSE;                                   %112-      14269920 C  0103:0041:2
      END;                                                               %112-      14269940 C  0103:0042:0
PRT(724) = *SEGMENT DESCRIPTOR*                                          %112-      14269960 C  0103:0043:3

                                                                                103 IS   44 LONG, NEXT SEG   98
        GO TO START;                                                     %112-      14269980 C  0098:0126:0
      PROCEDUREDEC;                                                                 14270000 T  0098:0126:1
        BEGIN                                                                       14271000 T  0098:0127:2
          LABEL START,START1;                                                       14272000 T  0098:0127:2
PRT(725) = *SEGMENT DESCRIPTOR*
```

```
                                                                                                                    START OF SEGMENT **********  104
                    LABEL START2;                                                                                          14273000 T  0104:000:0
                   BOOLEAN FWDTOG; COMMENT THIS TOGGLE IS THE FORWARD DEC INDICATOR;                                       14274000 T  0104:000:0
STACK(F+25) = FWDTOG
                      IF NOT SPECTOG THEN FUNCTOG←FALSE;                                                                   14275000 T  0104:000:0
                  FWDTOG←FALSE  ;                                                                                          14276000 T  0104:000:3
                      MAXSTACKO← MAXSTACK;                                                                                 14277000 T  0104:000:1
                  IF G←GTA1[J←J-1]=STREAMV                                                                                 14278000 T  0104:000:2
                   THEN                                                                                                    14279000 T  0104:000:2
                     BEGIN STREAMTOG←TRUE;                                                                                 14280000 T  0104:000:3
                       IF G←GTA1[J←J-1]=0 THEN TYPEV←STRPROCID                                                             14281000 T  0104:000:2
                                         ELSE                                                                              14282000 T  0104:001:0
                                          BEGIN                                                                            14283000 T  0104:001:1
                                           IF TYPEV←PROCID +G>INTSTRPROCID OR                                              14284000 T  0104:001:2
                                              TYPEV <BOOSTRPROCID                                                          14285000 T  0104:001:2
                                            THEN FLAG(004);                                                                14286000 T  0104:001:2
                                       IF NOT SPECTOG THEN                                                                 14287000 T  0104:001:2
                                            FUNCTOG←TRUE;                                                                  14288000 T  0104:001:3
                                            CHKSOB                                                                         14289000 T  0104:001:2
                     END                   END                                                                             14290000 T  0104:001:2
                     ELSE                                                                                                  14291000 T  0104:001:3
                    IF G=SAVEV OR G=0 THEN TYPEV←PROCID                                                                    14292000 T  0104:001:3
                              ELSE                                                                                         14293000 T  0104:002:1
                               IF TYPEV←REALSTRPROCID+G<BOOPROCID OR TYPEV>INTPROCID                                       14294000 T  0104:002:2
                                 THEN FLAG(005)                                                                            14295000 T  0104:002:3
                           ELSE BEGIN FUNCTOG←TRUE;G←GTA1[J←J-1];                                                          14296000 T  0104:002:2
                                END;                                                                                       14297000 T  0104:002:1
                    IF NOT STREAMTOG THEN SEGMENTSTART(G=SAVEV);                                                           14298000 T  0104:002:1
                      SAF ← G=SAVEV;                                                                                       14299000 T  0104:003:2
                                                                                                                           14300000 T  0104:003:0
                                                                                                                           14301000 T  0104:003:0
                                                                                                                           14302000 T  0104:003:0
                  MODE←MODE+1;                                                                                             14303000 T  0104:003:0
                  LO←PROINFO;                                                                                              14304000 T  0104:003:3
                  SCATTERELBAT;                                                                                            14305000 T  0104:003:0
               COMMENT CHECK TO SEE IF DECLARED FORWARD PREVIOUSLY                                    ;                    14306000 T  0104:003:1
                  IF LEVELF=LEVEL                                                                                          14307000 T  0104:003:1
                    THEN                                                                                                   14308000 T  0104:003:2
                      BEGIN                                                                                                14309000 T  0104:003:3
                        IF G←TAKE(LINKF+1)≥0                                                                               14310000 T  0104:003:0
                          THEN FLAG(006);                                                                                  14311000 T  0104:003:3
                        FWDTOG←TRUE;                                                                                       14312000 T  0104:003:3
                      XMARK(DECLREF); % PROCEDURE DECLARED FORWARD, MARK LAST      %110-                                14312100 C  0104:004:0
                                      % XREF ENTRY AS A DECLARATION.             %110-                                14312101 C  0104:004:1
                        PROAD←ADDRSF;                                                                                      14313000 T  0104:004:1
                        PROINFO←ELBAT[I];MARK←LINKF+INCRF;STEPIT                                                           14314000 T  0104:004:2
                             ;PUT(-G,LINKF+1);                                                                             14315000 T  0104:004:3
                      END                                                                                                  14316000 T  0104:004:3
                    ELSE                                                                                                   14317000 T  0104:004:3
                      BEGIN STOPENTRY←TRUE; P2←TRUE;                                                                       14318000 T  0104:004:3
                        STOPGSP←LEVEL>1 AND STREAMTOG;                                                                     14318500 T  0104:005:3
                        ENTRY(TYPEV);    MARK←NEXTINFO;PUTNBUMP(0);                                                        14319000 T  0104:005:3
                        STOPGSP←FALSE;                                                                                     14319500 T  0104:005:3
                        PROINFO←TAKE(LASTINFO)& LASTINFO[35:35:13];PROAD←ADDRSF;                                           14320000 T  0104:005:1
                     P2←STOPENTRY←FALSE                                                                                    14321000 T  0104:005:3
                      END;                                                                                                 14322000 T  0104:005:3
                               PJ←0; LEVEL←LEVEL+1;                                                                        14323000 T  0104:006:1
```

```
          IF STREAMTOG THEN STREAMWORDS;                                                14324000 T  0104:0062:1
       IF ELCLASS=SEMICOLON THEN GO TO START1;                                          14325000 T  0104:0064:0
      IF ELCLASS≠LEFTPAREN THEN FLAG(007);                                              14326000 T  0104:0065:2
COMMENT; THE FOLLOWING 8  STATEMENTS FOOL THE SCANNER AND BLOCK,PUTTING                 14327000 T  0104:0067:2
         FORMAL PARAMETER ENTRIES IN THE ZERO ROW OF INFO;                              14328000 T  0104:0067:2
        RR1←NEXTINFO;                                                                   14329000 T  0104:0067:2
       LASTINFOT←LASTINFO; LASTINFO←NEXTINFO←1;                                         14330000 T  0104:0068:0
                 PUTNBUMP(0);                                                           14331000 T  0104:0070:0
               PTOG←TRUE;    I←I+1;                                                    14332000 T  0104:0070:1
       ENTRY(SECRET);                                                                   14333000 T  0104:0072:1
     IF FWDTOG THEN                                                                     14333100 T  0104:0073:3
          BEGIN                                                                         14333200 T  0104:0073:3
          IF GT1:=TAKE(MARK).[40:8] ≠ PJ THEN FLAG(48); % WRONG                         14333300 T  0104:0074:0
            % NUMBER OF PARAMETERS. WE DON"T WANT TO CLOBBER INFO.                     14333400 T  0104:0077:3
          END                                                                           14333500 T  0104:0077:3
ELSE                                                                                    14333600 T  0104:0077:3
  PUT(PJ,MARK);                                                                         14334000 T  0104:0077:3
            P←PJ;                                                                       14335000 T  0104:0079:2
     IF ELCLASS≠RTPAREN                                                                 14336000 T  0104:0080:0
       THEN FLAG(008);                                                                  14337000 T  0104:0080:0
     IF STEPI≠SEMICOLON                                                                 14338000 T  0104:0082:0
       THEN FLAG(009);                                                                  14339000 T  0104:0082:1
COMMENT MARK PARAMETERS VALUE IF THERE IS A VALUE PART;                                 14340000 T  0104:0084:0
   IF STEPI=VALUEV                                                                      14341000 T  0104:0084:0
      THEN                                                                              14342000 T  0104:0084:1
        BEGIN                                                                           14343000 T  0104:0085:2
          DO                                                                            14344000 T  0104:0085:3
            IF STEPI≠SECRET                                                             14345000 T  0104:0086:0
              THEN FLAG(010)                                                            14346000 T  0104:0086:1
              ELSE                                                                      14347000 T  0104:0087:3
                BEGIN                                                                   14348000 T  0104:0088:0
                  IF G←ELBAT[I].ADDRESS=0 OR G>PJ                                       14349000 T  0104:0088:1
                   THEN                                                                 14350000 T  0104:0091:2
                     FLAG(010);                                                         14351000 T  0104:0091:3
                                 G←TAKE(ELBAT[I]);                                      14352000 T  0104:0093:2
                                 PUT(G&1[10:47:1],ELBAT[I])                             14353000 T  0104:0094:1
                END                                                                     14354000 T  0104:0096:0
          UNTIL                                                                         14355000 T  0104:0096:1
            STEPI≠COMMA;                                                                14356000 T  0104:0096:1
   IF ELCLASS≠SEMICOLON                                                                 14357000 T  0104:0098:0
     THEN FLAG(011)                                                                     14358000 T  0104:0098:1
     ELSE STEPIT                                                                        14359000 T  0104:0099:3
       END;I←I-1;                                                                       14360000 T  0104:0100:1
 IF STREAMTOG                                                                           14361000 T  0104:0102:1
     THEN                                                                               14362000 T  0104:0102:1
       BEGIN                                                                            14363000 T  0104:0102:1
                 BUP←PJ; SPECTOG←TRUE;GO TO START1                                      14364000 T  0104:0103:2
       END                                                                              14365000 T  0104:0105:2
     ELSE                                                                               14366000 T  0104:0105:2
       BEGIN                                                                            14367000 T  0104:0105:2
         SPECTOG←TRUE;                                                                  14368000 T  0104:0105:3
         BUP←0;                                                                         14369000 T  0104:0106:1
         IF ELCLASS≠DECLARATORS                                                         14370000 T  0104:0107:2
            THEN FLAG(012)                                                              14371000 T  0104:0107:3
     END;                                                                               14372000 T  0104:0108:1
START:PTOG←FALSE;LASTINFO←LASTINFOT;NEXTINFO←IF FWDTOG THEN RR1 ELSE                    14373000 T  0104:0109:2
         MARK←PJ←1;                                                                     14374000 T  0104:0113:2
```

```
                START1;PINFO0←NEXTINFO;                                                                                    14375000 T  0104:011:4:1
                START2: END;                                                                                               14376000 T  0104:011:5:3
PRT(726) = *SEGMENT DESCRIPTOR*

                                                                                                                  104 IS  117 LONG, NEXT SEG   98
                     IF SPECTOG OR STREAMTOG                                                                          14377000 T  0098:012:8:0
                        THEN                                                                                          14378000 T  0098:012:8:0
                             GO TO START;                                                                             14379000 T  0098:012:8:1
                COMMENT IF SPECTOG IS ON THEN THE BLOCK WILL PROCESS THE SPECIFICATION                                 14380000 T  0098:012:9:3
                        PART SIMILARY TO DECLARATIONS WITH A FEW NECESSARY VARIATIONS;                                 14381000 T  0098:012:9:3
                HF:                                                                                                    14382000 T  0098:012:9:3
                   BEGIN                                                                                               14383000 T  0098:013:0:0
                     LABEL START,STOP;                                                                                 14384000 T  0098:013:0:0
PRT(727) = *SEGMENT DESCRIPTOR*

                                                                                                                START OF SEGMENT **********  105
                      DEFINE  TESTLEV = LEVEL>2 #;                                                                     14384100 T  0105:000:0:0
                IF STREAMTOG                                                                                           14385000 T  0105:000:0:0
                  THEN BEGIN                                                                                           14386000 T  0105:000:0:0
                   IF TESTLEV THEN JUMPCHKNX ELSE SEGMENTSTART(TRUE);PJ←P;                                             14387000 T  0105:000:1:1
                                 PTOG←FALSE;                                                                           14388000 T  0105:000:4:1
                      PUT(TAKE(GIT(PROINFO))&L[28:36:12],GIT(PROINFO));                                                14388100 T  0105:000:5:2
                   IF TESTLEV THEN BEGIN EMIT0(584);          END;                                                     14389000 T  0105:000:8:1
                           IF STEPI=BEGINV                                                                             14393000 T  0105:001:0:1
                             THEN                                                                                      14394000 T  0105:001:1:2
                                BEGIN                                                                                  14395000 T  0105:001:1:3
                                  WHILE STEPI=DECLARATORS OR ELCLASS=LOCALV                                            14396000 T  0105:001:2:0
                                     DO                                                                                14397000 T  0105:001:4:0
                                      BEGIN                                                                            14398000 T  0105:001:5:3
                                        IF ELBAT[I],ADDRESS=LABELV                                                     14399000 T  0105:001:5:3
                                         THEN                                                                          14400000 T  0105:001:6:1
                                           BEGIN                                                                       14401000 T  0105:001:7:2
                                             STOPDEFINE←STOPGSP←STOPENTRY←TRUE;                                        14402000 T  0105:001:7:3
                DO BEGIN STOPDEFINE←TRUE;STEPIT;ENTRY(STLABID);PUTNBUMP(0) END UNTIL                                   14403000 T  0105:001:9:2
                           ELCLASS≠COMMA;STOPGSP←STOPENTRY←FALSE                                                       14404000 T  0105:002:2:1
                                           END                                                                         14405000 T  0105:002:4:0
                                    ELSE                                                                               14406000 T  0105:002:5:2
                                      BEGIN                                                                            14407000 T  0105:002:5:2
                                              I←I+1;                                                                   14408000 T  0105:002:5:3
                                        ENTRY(LOCLID)                                                                  14409000 T  0105:002:7:2
                                      END                                                                              14410000 T  0105:002:7:2
                                  END;                                                                                 14411000 T  0105:002:7:3
                IF FUNCTOG  THEN                                                                                       14411100 T  0105:002:8:0
                     PUT((Z←TAKE(PROINFO))&LOCLID[2:41:7] &                                                            14411200 T  0105:002:8:1
                                 (PJ←2←REAL(TESTLEV))[16:37:11],PROINFO);                                              14411300 T  0105:003:1:3
                               COMPOUNDTAIL                                                                            14412000 T  0105:003:4:1
                          END                                                                                          14413000 T  0105:003:4:1
                        ELSE                                                                                           14414000 T  0105:003:5:2
                BEGIN                                                                                                  14415000 T  0105:003:5:2
                IF   FUNCTOG  THEN                                                                                     14415100 T  0105:003:5:3
                     PUT(( Z←TAKE(PROINFO))& LOCLID[2:41:7]&                                                           14415200 T  0105:003:5:3
                                 (PJ←2←REAL(TESTLEV))[16:37:11],PROINFO);                                              14415300 T  0105:003:8:1
                STREAMSTMT;                                                                                            14415400 T  0105:004:1:3
                END;                                                                                                   14415500 T  0105:004:2:0
              COMMENT THE FOLLOWING BLOCK CONSTITUTES THE STREAM PROCEDURE PURGE;                                      14416000 T  0105:004:2:0
                BEGIN                                                                                                  14417000 T  0105:004:2:0
                REAL NLOC,NLAB;                                                                                        14418000 T  0105:004:2:0
PRT(730) = *SEGMENT DESCRIPTOR*

                                                                                                                START OF SEGMENT **********  106
STACK(F+25) = NLOC
```

```
STACK(F+26) = NLAB
                    DEFINE SES=18#,SED=6#,TRW=5#;                                                    14419000 T  0106:000:0
                    DEFINE LOC=[36:12]#,LASTGT=[24:12]#;                                             14420000 T  0106:000:0
                    J← LASTINFO;                                                                     14421000 T  0106:000:0
                        NLOC←NLAB←0;                                                                 14422000 T  0106:000:1
                    DO                                                                               14423000 T  0106:002:0
                    BEGIN                                                                            14424000 T  0106:002:0
                    IF(GT1←TAKE(J)).CLASS=LOCLID THEN                                                14425000 T  0106:002:0
                        BEGIN                                                                        14426000 T  0106:000:0
                        IF BOOLEAN(GT1.FORMAL) THEN                                                  14427000 T  0106:000:1
                            BEGIN                                                                    14428000 T  0106:000:3
                            IF GT1<0 THEN                                                            14429000 T  0106:000:0
                            PUT(TAKE(GT2←MARK←P-GT1.ADDRESS+1)&FILEID[2:41:7]                        14430000 T  0106:000:1
                                ,GT2);                                                               14431000 T  0106:001:3
                            END                                                                      14432000 T  0106:001:0
                        ELSE  NLOC←NLOC+1;                                                           14433000 T  0106:001:0
                        END                                                                          14434000 T  0106:001:0
                    ELSE                                                                             14435000 T  0106:001:0
                        BEGIN                                                                        14436000 T  0106:001:0
                        IF  GT1.ADDRESS≠0 THEN NLAB←NLAB+1;                                          14437000 T  0106:001:1
                        IF(GT3←TAKE(GIT(J))).LASTGT≠0 AND GT3.LOC =0 THEN                            14438000 T  0106:001:7:3
                            BEGIN                                                                    14439000 T  0106:002:1:3
                            MOVE(9,INFO[0,J],ACCUM[0]);                                              14440000 T  0106:002:2:0
                            Q←ACCUM[1];                                                              14441000 T  0106:002:4:1
                            FLAG(267);                                                               14442000 T  0106:002:5:3
                            ERRORTOG←TRUE;                                                           14443000 T  0106:002:6:1
                            END;                                                                     14444000 T  0106:002:7:2
                        XREFDUMP(J); % DUMP XREF INFO                                %110-           14445000 T  0106:002:7:2
                        G←(GT2←TAKE(J+1)).PURPT;                                                     14445100 C  0106:002:7:2
                        IF  GT1.[2:8] # STLABID×2+1  THEN                                            14446000 T  0106:002:9:2
                            STACKHEAD[(0&GT2[12:12:36])MOD 125]←TAKE(J).LINK;                        14447000 T  0106:003:2:0
                        END UNTIL J←J-G≤1;                                                           14448000 T  0106:003:4:0
                                                                                                     14449000 T  0106:003:8:1
                        IF   TESTLEV   THEN BEGIN EMITC(1,0); EMITO(BFW) END                         14450000 T  0106:004:0:1
                                       ELSE  EMIT(0);                                                14451000 T  0106:004:0:1
                    PUT(TAKE(MARK)&NLOC[1:42:6]&L[16:36:12]&P[40:40:8],MARK);                        14451100 T  0106:004:3:3
                        IF FUNCTOG THEN                                                              14451200 T  0106:004:5:2
                            PUT(Z, PROINFO);                                                         14452000 T  0106:004:9:3
                        STREAMWORDS;                                                                 14457000 T  0106:004:9:3
                        STREAMTOG←FALSE;                                                             14460000 T  0106:005:1:2
                    IF NOT TESTLEV THEN BEGIN PROGDESCBLDR(PROAD,TRUE,(L+3)DIV 4,CHAR);              14461000 T  0106:005:1:3
                                              SEGMENT((L+3)DIV 4,PROINFO,ADDRESS);                   14461100 T  0106:005:2:1
                                                  RIGHT(L); L←0;                                     14461200 T  0106:005:6:0
                                       END;                                                          14461300 T  0106:005:8:1
                        IF LISTER AND FORMATOG THEN SPACEITDOWN;                                     14461400 T  0106:006:0:0
                        END;                                                                         14461500 T  0106:006:0:0
PRT(731) = *SEGMENT DESCRIPTOR*                                                                      14462000 T  0106:006:6:0

                                                                                        106 IS   67 LONG, NEXT SEG  105
                           LASTINFO←LASTINFOT;NEXTINFO←MARK←P+1;                                     14463000 T  0105:004:3:2
                        END                                                                          14464000 T  0105:004:5:3
                    ELSE                                                                             14465000 T  0105:004:5:3
                        BEGIN                                                                        14466000 T  0105:004:5:3
                     IF STEPI=FORWARDV                                                               14467000 T  0105:004:6:0
                       THEN                                                                          14468000 T  0105:004:6:1
                        BEGIN                                                                        14469000 T  0105:004:7:2
                          XREFIT(PROINFO,0,FORWARDREF); % WE NEED THIS SO WE CAN FIND                14469100 C  0105:004:7:3
```

```
                                        % THE FORWARD DECL. DURING XREF
          PUT(-TAKE(G+PROINFO.LINK+1),G);
               PURGE(PINFOO);
          STEPIT
        END
       ELSE
        BEGIN
      PROADO+PROAD;
          TSUBLEVEL+SUBLEVFL;SUBLEVEL+LEVEL ;STACKCTRO+STACKCTR;
     IF MODE=1 THEN FRSTLEVEL+LEVEL;STACKCTR+513+REAL(FUNCTOG);
          IF ELCLASS = BEGINV THEN
      BEGIN
               CALLINFO+(CALLX+CALLX+1)+1;
               NESTCTR+STACKCTR;
                 BLOCK(TRUE)
               ; PURGE(PINFOO);
               IF NESTOG THEN
               BEGIN GT1+TAKE(PROINFO),ADDRESS;
                    NESTPRT[GT1]+0&PROINFO[35:35:13]&CALLINFO[22:35:13];
                    CALL[CALLINFO-1]+(TAKE(GIT(PROINFO))+NESTCTR-511)&
                         CALLX[22:35:13];
               END;
               L+0;
                 GO TO STOP END;
                  BEGIN
FLAG(052);
                    RELAD+L  ;
          STMT;
                     HTTEOAP(FALSE,RELAD,PINFOO,PROAD);
                      END;
                STOP;
                    SUBLEVEL+TSUBLEVEL;
                    STACKCTR+STACKCTRO;
                    IF LISTER AND FORMATOG THEN SPACEITDOWN;
                      END;
                      END;
                   PROINFO+LO;
                   IF JUMPCTR=LEVEL
                      THEN
                           JUMPCTR+LEVEL-1;
                       LEVEL+LEVEL-1;
                     MODE+MODE-1;
                      MAXSTACK+MAXSTACKO;
              START;END;
PRT(732) = *SEGMENT DESCRIPTOR*

                     GO TO START;
            CALLSTATEMENT; FOULED + L;
               JUMPCHKX;IF SOP THEN BEGIN Z+STACKCTR-513;WHILE Z+Z-1≥0
                                        DO EMITL(0)   END;
                    IF SPECTOG THEN BEGIN
                                    FLAG(12);GO TO HF
                                   END;
                    BEGINCTR + BEGINCTR-1;
                    IF ERRORTOG
                      THEN COMPOUNDTAIL
                      ELSE
```

```
14469101 C  0105:0050:0
14470000 T  0105:0050:0
14471000 T  0105:0053:2
14472000 T  0105:0054:0
14473000 T  0105:0054:0
14474000 T  0105:0054:1
14475000 T  0105:0054:1
14476000 T  0105:0055:2
14477000 T  0105:0055:3
14478000 T  0105:0058:0
14479000 T  0105:0061:2
14481000 T  0105:0062:0
14481100 T  0105:0062:1
14481200 T  0105:0064:1
14482000 T  0105:0065:3
14483000 T  0105:0065:3
14483100 T  0105:0067:2
14483200 T  0105:0067:3
14483300 T  0105:0070:0
14483400 T  0105:0073:3
14483500 T  0105:0079:2
14483600 T  0105:0080:1
14483700 T  0105:0080:1
14484000 T  0105:0081:3
14485000 T  0105:0082:0
14486000 T  0105:0082:0
14487000 T  0105:0082:1
14488000 T  0105:0083:3
14489000 T  0105:0084:0
14490000 T  0105:0085:3
14491000 T  0105:0085:3
14492000 T  0105:0086:0
14493000 T  0105:0086:1
14493500 T  0105:0087:3
14494000 T  0105:0093:2
14495000 T  0105:0093:2
14496000 T  0105:0093:2
14497000 T  0105:0094:0
14498000 T  0105:0094:0
14499000 T  0105:0094:1
14500000 T  0105:0096:1
14501000 T  0105:0097:3
14502000 T  0105:0099:2
14503000 T  0105:0099:3

105 IS  101 LONG, NEXT SEG   98
14504000 T  0098:0131:2
14505000 T  0098:0131:3
14506000 T  0098:0132:1
14506500 T  0098:0137:2
14507000 T  0098:0139:3
14508000 T  0098:0140:0
14509000 T  0098:0141:3
14510000 T  0098:0141:3
14511000 T  0098:0142:1
14512000 T  0098:0142:1
14513000 T  0098:0143:3
```

```
                    BEGIN                                                                                  14514000 T   0098:014:0
                     STMT;                                                                                 14515000 T   0098:014:1
                     IF ELCLASS←TABLE(I+1)=DECLARATORS                                                     14516000 T   0098:014:2
                       THEN                                                                                14517000 T   0098:014:1
                         BEGIN                                                                             14518000 T   0098:014:2
                           ELBAT[I],CLASS←SEMICOLON;                                                       14519000 T   0098:014:3
                           BEGINCTR←BEGINCTR+1;                                                            14520000 T   0098:015:0
                           GO TO START                                                                     14521000 T   0098:015:3
                         END                                                                               14522000 T   0098:015:0
                       ELSE                                                                                14523000 T   0098:015:0
                         COMPOUNDTAIL                                                                      14524000 T   0098:015:0
                    END;                                                                                   14525000 T   0098:015:1
                FUNCTOG←FUNCTOGO;                                                                          14599000 T   0098:015:2
                IF SOP THEN HTTEOAP(FALSE,FIRSTX,NINFOO,BLKAD)                                             14600000 T   0098:015:3
                    ELSE BEGIN IF NESTOG THEN SORTNEST; PURGE(NINFOO); END;                                14601000 T   0098:015:3
                SEGMENT((L+3)DIV 4,PROADO);                                                                14602000 T   0098:015:2
                    IF LEVEL>1 THEN RIGHT(L);                                                              14603000 T   0098:016:2
              IF LEVEL + LEVEL-1 = 0 THEN CONSTANTCLEAN;                                                   14604000 T   0098:016:2
                                                                                                           14605000 T   0098:016:3
              AJUMP←AJUMPO;                                                                                14606000 T   0098:016:3
                                                                                                           14607000 T   0098:016:1
                 FIRSTX←FIRSTXO;                                                                           14608000 T   0098:016:1
              SAVEL←SAVELO;                                                                                14609000 T   0098:016:2
              STACKCTR←STACKCTRO;                                                                          14610000 T   0098:016:0
                                                                                                           14611000 T   0098:016:1
                                                                                                           14612000 T   0098:016:1
        END BLOCK;                                                                                         14613000 T   0098:016:1
```

98 IS 176 LONG. NEXT SEG 3

```
        COMMENT THIS SECTION CONTAINS THE VARIABLE ROUTINE AND ITS SIDEKICKS;      15000000 T   0003:062:1
                                                                                  15001000 T   0003:062:1
                                                                                  15002000 T   0003:062:1
                                                                                  15003000 T   0003:062:1
                                                                                  15004000 T   0003:062:1
                                                                                  15005000 T   0003:062:1
                                                                                  15006000 T   0003:062:1
                                                                                  15007000 T   0003:062:1
                                                                                  15008000 T   0003:062:1
                                                                                  15009000 T   0003:062:1
                                                                                  15012000 T   0003:062:1
                                                                                  15013000 T   0003:062:1
                                                                                  15014000 T   0003:062:1
                                                                                  15015000 T   0003:062:1
                                                                                  15016000 T   0003:062:1
                                                                                  15017000 T   0003:062:1
                                                                                  15018000 T   0003:062:1
                                                                                  15019000 T   0003:062:1
                                                                                  15020000 T   0003:062:1
                                                                                  15021000 T   0003:062:1
                                                                                  15022000 T   0003:062:1
                                                                                  15023000 T   0003:062:1
                                                                                  15024000 T   0003:062:1
                                                                                  15025000 T   0003:062:1
                                                                                  15026000 T   0003:062:1
```

```
COMMENT  THE FOLLOWING BLOCK HANDLES THE FOLLOWING CASES
         OF SIMPLE VARIABLES:
            1. V ← EXP          ,WHERE V IS FORMAL-CALL BY NAME.
            2. V ← EXP          ,ALL V EXCEPT FORMAL-NAME.
            3. V,[S:L] ← EXP    ,WHERE V IS FORMAL-CALL BY NAME.
            4. V,[S:L] ← EXP    ,ALL V EXCEPT FORMAL-NAME.
            5. V,[S:L]          ,ALL V.
            6. V                ,ALL V.
         CODE EMITED FOR THE ABOVE CASES IS AS FOLLOWS:
            1. VN,EXP,M*,XCH,←.
            2. EXP,M*,VL,←.
            3. VN,DUP,COC,EXP,T,M*,XCH,←.
            4. VV,EXP,T,M*,VL,←.
            5. ZEROL,VV,T .
            6. VV .
            WHERE VN = DESC V
                  EXP= ARITH. OR BOOLEAN EXPRESSION,AS REQUIRED.
                  M* = CALL ON MONITOR ROUTINE,IF REQUIRED.
                  VL = LITC V
                  VV = OPDC V
                  ←  = STORE INSTRUCTION(ISD,ISN,SND OR STD).
                  T  = BIT TRANSFER CODE(DIA,DIB,TRB).
                  ZEROL = LITC 0
                  DUP,COC,XCH = THE INSTRUCTIONS DUP,COC,AND XCH.
                  OF COURSE, EXP WILL CAUSE RECURSION,IN GENERAL,AND THUS
                  THE PARAMETER P1 AND THE LOCALS CAN NOT BE HANDLED IN A
                  GLOBAL FASHION.
                  THE PARAMETER P1 IS USED TO TELL THE VARIABLE ROUTINE
                  WHO CALLED IT.  SOME OF THE CODE GENERATION AND SOME
                  SYNTAX CHECKS DEPEND UPON A PARTICULAR VALUE OF P1 .
                                                                    ;
      PROCEDURE VARIABLE(P1); INTEGER P1;
                  BEGIN
                  REAL TALL,  COMMENT ELBAT WORD FOR VARIABLE;
STACK(F+2) = TALL
                        T1  ,  COMMENT 1ST INTEGER OF PARTIAL WORD SYNTAX;
STACK(F+3) = T1
                        T2  ,  COMMENT 2ND INTEGER OF PARTIAL WORD SYNTAX;
STACK(F+4) = T2
                        J   ;  COMMENT SUBSCRIPT COUNTER ;
STACK(F+5) = J
               REAL REMEMBERSEQNO; % REMEMBERS SEQUENCE NUMBER OF VARIABLE %110-
STACK(F+6) = REMEMBERSEQNO
                                   % ON LEFT HAND SIDE OF ASSIGNMENT SO WE %110-
                                   % CAN XREF IT CORRECTLY.                %110-
```

```
15027000 T  0003:0628:1
15028000 T  0003:0628:1
15029000 T  0003:0628:1
15030000 T  0003:0628:1
15031000 T  0003:0628:1
15032000 T  0003:0628:1
15033000 T  0003:0628:1
15034000 T  0003:0628:1
15035000 T  0003:0628:1
15036000 T  0003:0628:1
15037000 T  0003:0628:1
15038000 T  0003:0628:1
15039000 T  0003:0628:1
15040000 T  0003:0628:1
15041000 T  0003:0628:1
15042000 T  0003:0628:1
15043000 T  0003:0628:1
15044000 T  0003:0628:1
15045000 T  0003:0628:1
15046000 T  0003:0628:1
15047000 T  0003:0628:1
15048000 T  0003:0628:1
15049000 T  0003:0628:1
15050000 T  0003:0628:1
15051000 T  0003:0628:1
15052000 T  0003:0628:1
15053000 T  0003:0628:1
15054000 T  0003:0628:1
15055000 T  0003:0628:1
15056000 T  0003:0628:1
15057000 T  0003:0628:1
15058000 T  0003:0628:1
15059000 T  0003:0628:1
15060000 T  0003:0628:1
15061000 T  0003:0628:1
15062000 T  0003:0628:1
15063000 T  0003:0628:1
15064000 T  0003:0628:1
15065000 T  0003:0628:1
15066000 T  0003:0628:1
15067000 T  0003:0628:1
15068000 T  0003:0628:1
15069000 T  0003:0628:1
15070000 T  0003:0628:1
15071000 T  0003:0628:1
15072000 T  0003:0628:1
                    START OF SEGMENT ********** 107
15073000 T  0107:0000:0
15074000 T  0107:0000:0
15075000 T  0107:0000:0
15075550 C  0107:0000:0
15075551 C  0107:0000:0
15075552 C  0107:0000:0
```

```
                        LABEL EXIT, NEXT, LASS;                                  %114-      15076000 P  0107!000!0
                        DEFINE                                                   %114-      15076100 P  0107!000!0
                          FORMALNAME = [9:2] = 2#,                               %114-      15076110 C  0107!000!0
                          LONGID = NAME!D#,                                      %114-      15076120 C  0107!000!0
                          PARTIALWORD = (T1 ≠ 0)#,                               %114-      15076130 C  0107!000!0
                          ERREXIT(NUM) = BEGIN ERR(NUM); GO TO EXIT; END#;       %114-      15076140 C  0107!000!0
                        BOOLEAN                                                  %114-      15076200 P  0107!000!0
                          SPCLMON,                                               %114-      15076300 C  0107!000!0
STACK(F+7) = SPCLMON
                          DUPIT;  % TRUE IF WE ARE DOING UPDATE TYPE ASSIGN-     %114-      15076400 C  0107!000!0
STACK(F+10) = DUPIT
                                  % MENT, I.E., I != * + 1.,                     %114-      15076500 C  0107!000!0
                        TALL←ELBAT[I] ;                                                     15077000 T  0107!000!0
                        IF   ELCLASS ≤ INTPROCID  THEN                                      15078000 T  0107!000!2
                           BEGIN                                                            15079000 T  0107!000!3
                             IF TALL.LINK ≠PROINFO.LINK THEN                                15080000 T  0107!000!0
                                  BEGIN ERR(211); GO TO EXIT END;                           15081000 T  0107!000!0
                  COMMENT  211  VARIABLE-FUNCTION IDENTIFIER USED OUTSIDE OF ITS SCOPE*;    15082000 T  0107!000!3
                        TALL←TALL &(ELCLASS+4)[2:4:7] & 513[16:37:11];                      15083000 T  0107!000!3
                          END                                                               15084000 T  0107!000!2
                        ELSE  CHECKER(TALL);                                                15085000 T  0107!000!2
                        REMEMBERSEQNO != CARDNUMBER;                             %110-      15085100 C  0107!001!0
                        IF TALL.CLASS ≤ INTID THEN % SIMPLE VAR OR FORMAL NAME              15085200 C  0107!001!2
                          BEGIN                                                  %114-      15085400 C  0107!001!0
                            IF STEPI ≠ ASSIGNOP THEN                             %114-      15085600 C  0107!001!1
                              IF P1 = FL THEN                                    %114-      15085800 C  0107!001!3
                                BEGIN                                            %114-      15086000 P  0107!001!2
                              IF ELCLASS < AMPERSAND THEN                        %114-      15086200 C  0107!001!3
                                EMITN(TALL.ADDRESS)                              %114-      15086400 C  0107!001!0
                              ELSE                                               %114-      15086600 C  0107!001!3
                                EMITV(TALL.ADDRESS);                             %114-      15086800 C  0107!001!0
                              GO TO EXIT;                                        %114-      15087000 P  0107!001!3
                          END                                                    %114-      15087200 C  0107!002!0
                        ELSE                                                     %114-      15087400 C  0107!002!0
                          IF ELCLASS = PERIOD THEN % PARTIAL WORD                %114-      15087600 C  0107!002!0
                            IF DOTSYNTAX(T1,T2) THEN % ERROR                     %114-      15087800 C  0107!002!3
                              GO TO EXIT                                         %114-      15088000 P  0107!002!2
                            ELSE                                                 %114-      15088200 C  0107!002!2
                              STEPIT;                                            %114-      15088400 C  0107!002!2
                      IF ELCLASS = ASSIGNOP THEN                                 %114-      15088600 C  0107!002!0
                        BEGIN                                                    %114-      15088800 C  0107!002!2
                          STACKCT != 1;                                          %114-      15089000 P  0107!002!3
                          IF PARTIALWORD THEN % MAKE SURE LEFT-MOST              %114-      15089200 C  0107!002!0
                            BEGIN                                                %114-      15089400 C  0107!002!2
                              IF P1 ≠ FS THEN % NOT LEFT-MOST                    %114-      15089600 C  0107!002!3
                                ERREXIT(201);                                    %114-      15089800 C  0107!002!0
                              XREFIT(TALL,REMEMBERSEQNO,ASSIGNREF);              %114-      15090600 C  0107!003!0
                            END                                                  %114-      15090800 C  0107!003!1
                          ELSE                                                   %114-      15091000 P  0107!003!1
                            XMARK(ASSIGNREF);                                    %114-      15091200 C  0107!003!1
                          IF TABLE(I+1) = ASTRISK THEN % MIGHT BE UPDATE                     15091400 C  0107!003!2
                            IF (DUPIT!=(TABLE(I+2) ≥ EQVOP AND TABLE(I+2)                    15091600 C  0107!003!2
                                ≤ MULOP) OR TABLE(I+2) = AMPERSAND) THEN                     15091800 C  0107!004!0
                              STEPIT; % STEP OVER ASTERISK                       %114-      15092000 P  0107!004!3
                          IF TALL.FORMALNAME THEN % FORMAL PARAMETER             %114-      15092200 C  0107!004!1
                            BEGIN                                                %114-      15092400 C  0107!004!0
                              EMITN(TALL.ADDRESS);                               %114-      15092600 C  0107!004!1
```

```
    IF PARTIALWORD OR DUPIT THEN % NEED VALUE%114-          15092800 C  0107:0049:3
      BEGIN                                                 %114-          15093000 P  0107:0051:2
        EMITO(DUP);                                         %114-          15093200 C  0107:0051:3
        EMITO(COC);                                         %114-          15093400 C  0107:0052:0
      END;                                                  %114-          15093600 C  0107:0053:2
  END                                                       %114-          15094800 C  0107:0053:2
ELSE % ITS A SIMPLE VARIABLE                                %114-          15095000 P  0107:0053:2
    IF PARTIALWORD OR DUPIT THEN                            %114-          15095400 C  0107:0053:2
      EMITV(TALL,ADDRESS);                                  %114-          15095600 C  0107:0054:1
IF PARTIALWORD AND DUPIT THEN                               %114-          15095700 C  0107:0056:1
  BEGIN                                                     %114-          15095800 C  0107:0057:3
    EMITO(DUP);                                             %114-          15095900 C  0107:0058:0
    EMITI(0,T1,T2);                                         %114-          15096000 P  0107:0059:2
  END;                                                      %114-          15096100 C  0107:0060:0
STACKCT := REAL(PARTIALWORD OR DUPIT);                      %114-          15096400 C  0107:0060:0
STEPIT;                                                     %114-          15096600 C  0107:0062:0
IF DUPIT THEN % ALREADY GOT FIRST PRIMARY                   %114-          15096800 C  0107:0062:1
  SIMPARITH                                                 %114-          15097000 P  0107:0062:1
ELSE                                                        %114-          15097200 C  0107:0063:2
  AEXP;                                                     %114-          15097400 C  0107:0063:3
EMITD(48+T2,T1,T2);                                         %114-          15097600 C  0107:0064:1
STACKCT := 0;                                               %114-          15097800 C  0107:0066:1
GT1 := IF TALL.CLASS = INTID THEN                           %114-          15098000 P  0107:0067:2
         IF P1 = FS THEN                                    %114-          15098200 C  0107:0068:1
           ISD                                              %114-          15098400 C  0107:0069:3
         ELSE                                               %114-          15098600 C  0107:0070:1
           ISN                                              %114-          15098800 C  0107:0071:2
       ELSE                                                 %114-          15099000 P  0107:0071:2
         IF P1 = FS THEN                                    %114-          15099200 C  0107:0071:3
           STD                                              %114-          15099400 C  0107:0072:1
         ELSE                                               %114-          15099600 C  0107:0073:2
           SND;                                             %114-          15099800 C  0107:0073:3
       IF TALL.FORMALNAME THEN                              %114-          15100000 P  0107:0074:1
         BEGIN                                              %114-          15100200 C  0107:0075:3
           EMITO(XCH); % TO GET DESCRIPTOR ON TOP           %114-          15100400 C  0107:0076:0
           IF TALL.ADDRESS > 1023 THEN % SET VARIANT        %114-          15100600 C  0107:0077:2
             EMITO(PRTE);                                   %114-          15100800 C  0107:0078:0
           EMITO(GT1);                                      %114-          15101000 P  0107:0079:3
         END                                                %114-          15101200 C  0107:0080:0
       ELSE                                                 %114-          15101400 C  0107:0080:0
         EMITPAIR(TALL,ADDRESS,GT1);                        %114-          15101600 C  0107:0080:0
     END                                                    %114-          15101800 C  0107:0082:0
    ELSE % NOT ASSIGNMENT TO SIMPLE VARIABLE                %114-          15102000 P  0107:0082:0
      BEGIN                                                 %114-          15102200 C  0107:0082:0
        IF P1 # FP THEN % EXPECTED ASSIGNMENT               %114-          15102400 C  0107:0082:1
          ERREXIT(202);                                     %114-          15102600 C  0107:0083:3
        EMITI(TALL,T1,T2); % EMIT OP CALL AND PARTIAL       %114-          15103400 C  0107:0085:2
      END;                      % WORD CODE                 %114-          15103600 C  0107:0086:1
  END OF SIMPLE VARIABLES                                   %114-          15103800 C  0107:0086:1
ELSE                                                                       15128000 T  0107:0086:1
IF TALL.CLASS#LABELID THEN                                                 15128100 T  0107:0086:1
COMMENT  THE FOLLOWING BLOCK HANDLES THESE CASES OF SUBSCRIPTED            15129000 T  0107:0088:0
         VARIABLES:                                                        15130000 T  0107:0088:0
            1. V[*]          ,ROW DESIGNATOR FOR SINGLE-DIMENSION.         15131000 T  0107:0088:0
            2. V[R,*]        ,ROW DESIGNATOR FOR MULTI-DIMENSION.          15132000 T  0107:0088:0
            3. V[R]          ,ARRAY ELEMENT,NAME OR VALUE.                 15133000 T  0107:0088:0
            4. V[R],[S:L]    ,PARTIAL WORD DESIGNATOR, VALUE.              15134000 T  0107:0088:0
```

```
   5. V[R] ←       ,ASSIGNMENT TO ARRAY ELEMENT.                          15135000 T  0107100881:0
   6. V[R].[S:L] ← ,ASSIGNMENT TO PARTIAL WORD,LEFT-MOST.                 15136000 T  0107100881:0
R IS A K-ORDER SUBSCRIPT LIST,I.E. R= R1,R2,...,RK.                       15137000 T  0107100881:0
IN THE CASE OF NO MONITORING ON V, THE FOLLOWING CODE                     15138000 T  0107100881:0
IS EMITTED FOR THE ABOVE CASES:                                           15139000 T  0107100881:0
        1. CASE #1 IS A SPECIAL CASE OF #2,NAMELY,SINGLE                  15140000 T  0107100881:0
           DIMENSION. THE CODE EMITTED IS:                                15141000 T  0107100881:0
              VL,LOD .                                                    15142000 T  0107100881:0
           EXECUTION: PLACES ARRAY DESCRIPTER IN REG A.                   15143000 T  0107100881:0
        2. THIS CODE IS BASIC TO THE SUBSCRIPTION PROCESS.                15144000 T  0107100881:0
           EACH SUBSCRIPT GENERATES THE FOLLOWING SEQUENCE                15145000 T  0107100881:0
           OF CODE:                                                       15146000 T  0107100881:0
              AEXP,L*,IF FIRST SUBSCRIPT THEN VN ELSE CDC                 15147000 T  0107100881:0
                ,LOD.                                                     15148000 T  0107100881:0
           FOR A K-ORDER SUBSCRIPTION,K-1 SEQUENCE ARE                    15149000 T  0107100881:0
           PRODUCED. THE AEXP IN EACH SEQUENCE REFERS TO                  15150000 T  0107100881:0
           THE CODE PRODUCED BY THE ARITHMETIC EXPRESSION                 15151000 T  0107100881:0
           PROCEDURE FOR THE ACTUAL SUBSCRIPT EXPRESSIONS.                15152000 T  0107100881:0
           L* REFERS TO THE CODE PRODUCED FOR SUBTRACTING                 15153000 T  0107100881:0
           NON-ZERO LOWER BOUNDS FROM THE SUBSCRIPT                       15154000 T  0107100881:0
           EXPRESSION(L* YIELDS NO CODE FOR ZERO BOUNDS).                 15155000 T  0107100881:0
           EXECUTION: PLACES ARRAY ROW DESCRIPTOR IN REG A                15156000 T  0107100881:0
                    . THE SPECIFIC ROW DEPENDS UPON THE                   15157000 T  0107100881:0
                   VALUES OF THE K-1 SUBSCRIPTS.                          15158000 T  0107100881:0
FOR THE REMAINING CASES,                                                  15159000 T  0107100881:0
           SEQUENCES OF CODE ARE EMITED AS IN CASE #2.                    15160000 T  0107100881:0
           HOWEVER,THE ACTUAL SEQUENCES ARE:                              15161000 T  0107100881:0
            ONE SEQUENCE ,(AEXP,L*),FOR THE 1ST SUBSCRIPT,                15162000 T  0107100881:0
            K-1 SEQUENCES,(IF FIRST SUBSCRIPT THEN VN                     15163000 T  0107100881:0
            ELSE CDC,LOD,AEXP,L*), FOR THE REMAINING                      15164000 T  0107100881:0
            SUBSCRIPTS,IF K>1.                                            15165000 T  0107100881:0
           AT THIS POINT, CASES #3-6 ARE DIFFERENTIATED                   15166000 T  0107100881:0
           AND ADDITION CODE,PARTICULAR TO EACH CASE,IS                   15167000 T  0107100881:0
           EMITTED.                                                       15168000 T  0107100881:0
        3. ADD THE SEQUENCE:                                              15169000 T  0107100881:0
             IF FIRST SUBSCRIPT THEN VV ELSE COC.                         15170000 T  0107100881:0
           EXECUTION: THE ARRAY ELEMENT IS PUT IN REG A.                  15171000 T  0107100881:0
        4. ADD THE SEQUENCE:                                              15172000 T  0107100881:0
             IF FIRST SUBSCRIPT THEN VV ELSE COC,ZEROL,                   15173000 T  0107100881:0
               XCH,T.                                                     15174000 T  0107100881:0
        5. ADD THE SEQUENCE:                                              15175000 T  0107100881:0
             IF FIRST SUBSCRIPT THEN VN ELSE CDC,EXP,                     15176000 T  0107100881:0
               XCH,*.                                                     15177000 T  0107100881:0
        6. ADD THE SEQUENCE:                                              15178000 T  0107100881:0
             IF FIRST SUBSCRIPT THEN VN ELSE CDC,DUP,LOD,                 15179000 T  0107100881:0
               EXP,T, XCH,*.                                              15180000 T  0107100881:0
    EXP,T,*,ZEROL,ETC. HAVE SAME MEANINGS AS DEFINED IN                   15181000 T  0107100881:0
    SIMPLE VARIABLE BLOCK.                                        ;       15182000 T  0107100881:0
BEGIN                                                                     15183000 T  0107100881:0
  IF STEPI ≠ LFTBRKET THEN % ARRAY ITEM NOT FOLLOWED BY    %114-          15183100 C  0107100881:1
    BEGIN                  % A SUBSCRIPT                   %114-          15183200 C  0107100891:3
      IF ELCLASS = PERIOD THEN                             %114-          15183300 C  0107100901:0
        IF DOTSYNTAX(T1,T2) THEN % ERROR IN PARTIAL WORD   %114-          15183400 C  0107100911:2
          GO TO EXIT                                       %114-          15183500 C  0107100921:1
        ELSE                                               %114-          15183600 C  0107100921:1
          STEPIT;                                          %114-          15183700 C  0107100921:1
      IF ELCLASS = ASSIGNOP THEN                           %114-          15183800 C  0107100931:3
```

```
BEGIN                                                          %114-
  IF PARTIALWORD THEN                                          %114-
    BEGIN                                                      %114-
      IF P1 ≠ FS THEN % PARTIAL WORD NOT LEFT-MOST             %114-
        ERREXIT(209);                                          %114-
      XREFIT(TALL,REMEMBERSEQNO,ASSIGNREF);                    %114-
    END                                                        %114-
  ELSE % ASSIGNMENT TO ID WITH NO PARTIAL WORD                 %114-
    XMARK(ASSIGNREF);                                          %114-
   IF TABLE(I+1) = ASTRISK THEN                                %114-
    IF (DUPIT := (TABLE(I+2) ≥ EQVOP AND TABLE(I+2)
        ≤ MULOP) OR TABLE(I+2) = AMPERSAND) THEN               %114-
      STEPIT;                                                  %114-
  IF PARTIALWORD OR DUPIT THEN % NEED VALUE ON STACK
    IF TALL.CLASS ≤ INTARRAYID THEN % NOT NAME ITEM
      EMITPAIR(TALL.ADDRESS,LOD)                               %114-
    ELSE                                                       %114-
      EMITN(TALL.ADDRESS);                                     %114-
    IF PARTIALWORD AND DUPIT THEN                              %114-
      BEGIN                                                    %114-
        EMITO(DUP);                                            %114-
        EMITI(0,T1,T2);                                        %114-
      END;                                                     %114-
    STACKCT := STACKCT + REAL(PARTIALWORD OR DUPIT);
    STEPIT;                                                    %114-
    IF DUPIT THEN % WE HANDLED FIRST PRIMARY                   %114-
      SIMPARITH                                                %114-
    ELSE                                                       %114-
      AEXP;                                                    %114-
    EMITD(48-T2,T1,T2);                                        %114-
    EMITPAIR(TALL.ADDRESS,IF P1 = FS THEN STD ELSE             %114-
                                                      SND);    %114-
      STACKCT := 0; % A AND B ARE EMPTY                        %114-
    END                                                        %114-
  ELSE % NOT ASSIGNMENT                                        %114-
    EMITI(TALL,T1,T2);                                         %114-
  GO TO EXIT;                                                  %114-
 END OF ASSIGNMENT TO NON SIMPLE NON SUBSCRIPTED               %114-
  VARIABLE;                                                    %114-
J ← 0;
STACKCT ← 0;
COMMENT  207  VARIABLE-MISSING LEFTBRACKET ON SUBSCRIPTED VARIABLE  *;
   NEXT;      IF STEPI = FACTOP THEN
                BEGIN
                IF   J+1≠ TALL.INCR THEN
                       BEGIN ERR(203);GO EXIT END;
COMMENT  203  VARIABLE- THE NUMBER OF SUBSCRIPTS USED IN A ROW        *
                ROW DESIGNATER  DOES NOT MATCH THE ARRAY             *
                DECLARATION.                                         *;
                IF STEPI ≠ RTBRKET THEN
                       BEGIN ERR(204);GO EXIT END;
COMMENT  204  VARIABLE- COMPILER EXPECTS A  ]  IN A ROW DESIGNATER    *;

COMMENT  205  VARIABLE- A ROW DESIGNATER APPEARS OUTSIDE OF A FILL    *
                STATEMENT OR ACTUAL PARAMETER LIST.                  *;
                IF  J=0  THEN
                       EMITPAIR(TALL.ADDRESS,LOD);
                STEPIT;
```

```
15183900 C  01071009:1
15184000 P  01071009:2
15184100 P  01071009:3
15184200 P  01071009:0
15184300 P  01071009:2
15184400 P  01071009:1
15184500 C  01071010:2
15184600 C  01071010:2
15184700 C  01071010:2
15184750 C  01071010:0
15184800 C  01071010:3
15184850 C  01071011:1
15185000 C  01071011:1
15185100 C  01071011:3
15185200 C  01071011:1
15185300 C  01071011:1
15185400 C  01071012:0
15185500 C  01071012:1
15185600 C  01071012:0
15185700 C  01071012:3
15185800 C  01071012:0
15185900 C  01071012:1
15186000 C  01071012:0
15186100 C  01071012:0
15186200 C  01071012:0
15186300 C  01071012:1
15186400 C  01071012:2
15186500 C  01071012:3
15186600 C  01071013:0
15186700 C  01071013:2
15186800 C  01071013:1
15186900 C  01071013:3
15187000 C  01071013:0
15187100 C  01071013:2
15187200 C  01071013:2
15187300 C  01071013:2
15187400 C  01071013:1
15187500 C  01071013:2
15187600 C  01071013:2
15234000 T  01071013:2
15234500 T  01071014:0
15235000 T  01071014:1
15253000 T  01071014:1
15254000 T  01071014:2:0
15255000 T  01071014:2:1
15256000 T  01071014:4:0
15257000 T  01071014:6:0
15258000 T  01071014:6:0
15259000 T  01071014:6:0
15260000 T  01071014:6:0
15261000 T  01071014:7:2
15262000 T  01071014:8:1
15263000 T  01071014:8:1
15264000 T  01071014:8:1
15265000 T  01071014:8:1
15266000 T  01071014:8:1
15267000 T  01071014:9:3
15274000 T  01071015:1:3
```

```
                    GO TO EXIT;                                                  15275000 T  0107:015:0
                END OF ROW DESIGNATOR PORTION  ;                                 15276000 T  0107:015:1
          IF ELCLASS=LITNO AND ELBAT[I],ADDRESS=0 AND TABLE(I+1)=RTBRKET         15276010 T  0107:015:1
               AND TALL.CLASS≥NAMEID THEN                                        15276020 T  0107:015:6:0
               BEGIN                                                             15276030 T  0107:015:8:1
               I←I+1;                                                            15276040 T  0107:015:9:2
               IF STEPI=ASSIGNOP THEN BEGIN                                      15276050 T  0107:016:0:0
LASS;               IF T1≠0 THEN EMITV(TALL.ADDRESS);                          15276060 T  0107:016:1:3
                    STEPIT;  AEXP;  EMITD(48+T2,T1,T2);                          15276070 T  0107:016:4:1
                    EMITN(TALL.ADDRESS);                                         15276080 T  0107:016:7:2
                    EMITO(IF TALL.CLASS≠NAMEID THEN                              15276090 T  0107:016:8:1
                         IF P1=FS THEN ISD ELSE ISN ELSE                         15276100 T  0107:017:0:0
                         IF P1=FS THEN STD ELSE SND);                            15276110 T  0107:017:3:2
               STACKCT ← 0;                                                      15276115 T  0107:017:5:3
                    GO TO EXIT     END                                           15276120 T  0107:017:6:1
               ELSE                                                              15276130 T  0107:017:7:2
               IF ELCLASS = PERIOD THEN BEGIN                                    15276140 T  0107:017:7:2
                    IF DOTSYNTAX(T1,T2) THEN GO TO EXIT;                         15276150 T  0107:017:8:1
                    IF STEPI = ASSIGNOP THEN IF P1=FS THEN GO TO LASS            15276160 T  0107:018:0:1
                    ELSE BEGIN ERR(209); GO EXIT END;                            15276170 T  0107:018:2:1
                              END;                                               15276180 T  0107:018:4:1
               IF P1=FS THEN BEGIN ERR(210); GO EXIT END;                        15276190 T  0107:018:4:1
                                                                                 15276200 T  0107:018:7:2
               EMITI(IF P1=FL THEN TALL ELSE TALL&REALID[2:41:7],T1,T2);         15276210 T  0107:018:7:2
                                                                                 15276220 T  0107:019:1:2
               GO TO EXIT;                                                       15276230 T  0107:019:1:2
               END;                                                              15276240 T  0107:019:1:3
               AEXP;                                                             15277000 T  0107:019:1:3
               STACKCT ← 1;                                                      15278000 T  0107:019:2:0
               J ← J + 1;                                                        15280000 T  0107:019:3:2
               IF  ELCLASS = COMMA THEN                                          15287000 T  0107:019:4:0
                    BEGIN                                                        15288000 T  0107:019:5:2
COMMENT  *****      MONITOR FUNCTION M4 GOES HERE ;                              15289000 T  0107:019:5:3
    IF J = 1 THEN EMITV(TALL.ADDRESS) ELSE EMITO(COC);                           15290000 T  0107:019:5:3
                                                                                 15291000 T  0107:019:9:2
                    GO TO NEXT;                                                  15292000 T  0107:019:9:2
                END OF SUBSCRIPT COMMA HANDLER ;                                 15293000 T  0107:019:9:3
               IF ELCLASS ≠ RTBRKET THEN BEGIN ERR(206);GO EXIT END;             15294000 T  0107:019:9:3
COMMENT  206  VARIABLE- MISSING RIGHT BRACKET ON SUBSCRIPTED VARIABLE*;          15295000 T  0107:020:2:0
               GT1←IF TALL.CLASS≥NAMEID THEN 1 ELSE TALL.INCR;                   15295100 T  0107:020:2:0
               IF J≠GT1  THEN                                                    15296000 T  0107:020:6:0
                              BEGIN ERR(208);GO TO EXIT END;                     15297000 T  0107:020:6:1
COMMENT  208  VARIABLE- NUMBER OF SUBSCRIPTS DOES NOT MATCH WITH      *          15298000 T  0107:020:8:1
                    ARRAY DECLARATION.                                 *;        15299000 T  0107:020:8:1
               IF STEPI = PERIOD THEN % PARTIAL WORD                  %114-     15300000 P  0107:020:8:1
                 IF DOTSYNTAX(T1,T2) THEN % ERROR                     %114-     15300100 C  0107:020:9:3
                   GO TO EXIT                                         %114-     15300200 C  0107:021:1:2
                 ELSE                                                 %114-     15300300 C  0107:021:1:2
                   STEPIT;                                            %114-     15300400 C  0107:021:1:2
               IF ELCLASS = ASSIGNOP THEN % ASSIGNMENT TO SUBSCRIPTED             15300500 C  0107:021:2:0
                 BEGIN                    % VARIABLE                  %114-     15300600 C  0107:021:3:2
                   IF PARTIALWORD THEN                                %114-     15300700 C  0107:021:3:3
                     IF P1 ≠ FS THEN                                  %114-     15300800 C  0107:021:4:0
                       ERREXIT(209); % PARTIALWORD NOT LEFT-MOST      %114-     15300900 C  0107:021:5:3
                   XREFIT(TALL,REMEMBERSEQNO,ASSIGNREF);              %114-     15301000 P  0107:021:7:2
                   IF J = 1 THEN % SINGLE-DIMENSIONED                 %114-     15301100 C  0107:021:9:3
                     EMITN(TALL.ADDRESS)                              %114-     15301200 C  0107:022:1:1
```

```
ELSE                                                          %114-      15301300 C  0107:022:0
  EMITO(CDC);                                                 %114-      15301400 C  0107:022:0
IF TALL.CLASS ≥ LONGID THEN % EXPLICIT INDEX OP               %114-      15301500 C  0107:022:3
  EMITO(INX);                  % REQUIRED                      %114-      15301600 C  0107:022:1
IF P1 = FR THEN % CALLED FROM FOR STATEMENT                   %114-      15301700 C  0107:022:0
  GO TO EXIT;                                                 %114-      15301800 C  0107:022:1
IF TABLE(I+1) = ASTRISK THEN                                  %114-      15301900 C  0107:022:2
  IF (DUPIT:=(TABLE(I+2) ≥ EQVOP AND TABLE(I+2)               %114-      15302000 P  0107:022:2
     ≤ MULOP) OR TABLE(I+2) = AMPERSAND) THEN                 %114-      15302100 C  0107:023:0
    STEPIT;                                                   %114-      15302200 C  0107:023:3
IF PARTIALWORD OR DUPIT THEN % NEED VALUE ON STACK                       15302300 C  0107:023:1
  BEGIN                                                       %114-      15302400 C  0107:023:0
    EMITO(DUP);                                               %114-      15302500 C  0107:023:1
    EMITO(LOD);                                               %114-      15302600 C  0107:023:2
  END;                                                        %114-      15302700 C  0107:024:0
IF PARTIALWORD AND DUPIT THEN                                 %114-      15302800 C  0107:024:0
  BEGIN                                                       %114-      15302900 C  0107:024:2
    EMITO(DUP);                                               %114-      15303000 P  0107:024:3
    EMITI(0,T1,T2);                                           %114-      15303100 C  0107:024:1
  END;                                                        %114-      15303200 C  0107:024:3
STEPIT;                                                       %114-      15303300 C  0107:024:3
IF DUPIT THEN                                                 %114-      15303400 C  0107:024:0
  SIMPARITH                                                   %114-      15303500 C  0107:024:1
ELSE                                                          %114-      15303600 C  0107:024:2
  AEXP;                                                       %114-      15303700 C  0107:024:3
EMITD(48-T2,T1,T2);                                           %114-      15303800 C  0107:024:1
EMITO(XCH);                                                   %114-      15303900 C  0107:024:0
  IF TALL.ADDRESS > 1023 THEN                                 %114-      15304000 P  0107:024:2
    EMITO(PRTE);                                              %114-      15304100 C  0107:025:0
  EMITO(IF TALL.CLASS MOD 2 = INTARRAYID MOD 2 THEN           %114-      15304200 C  0107:025:3
          IF P1 = FS THEN ISD ELSE ISN                        %114-      15304300 C  0107:025:0
        ELSE IF P1 = FS THEN STD ELSE SND);                   %114-      15304400 C  0107:025:1
  STACKCT := 0; % A & B ARE EMPTY                             %114-      15304500 C  0107:025:3
  P1 := 0;                                                    %114-      15304600 C  0107:026:1
 END OF ASSIGNMENT TO SUBSCRIPTED VARIABLE                    %114-      15304700 C  0107:026:3
ELSE                                                          %114-      15304800 C  0107:026:3
  BEGIN % HANDLING OF NO ASSIGNMENT CASE                      %114-      15304900 C  0107:026:3
    SPCLMON := P1 = FP OR PARTIALWORD OR ELCLASS ≥             %114-      15305000 P  0107:026:0
                  AMPERSAND;                                  %114-      15305100 C  0107:026:0
    IF J = 1 THEN % SINGLE DIMENSIONED                        %114-      15305200 C  0107:026:2
      IF TALL.CLASS ≥ LONGID THEN % NAME ITEM                 %114-      15305300 C  0107:026:0
        BEGIN                                                 %114-      15305400 C  0107:026:3
          EMITN(TALL.ADDRESS);                                %114-      15305500 C  0107:026:0
          EMITO(INX);                                         %114-      15305600 C  0107:026:3
          IF SPCLMON THEN                                     %114-      15305700 C  0107:027:0
            EMITO(LOD);                                       %114-      15305800 C  0107:027:1
        END                                                   %114-      15305900 C  0107:027:3
      ELSE % REFERENCE TO SINGLE DIMENSIONED ARRAY            %114-      15306000 P  0107:027:3
        IF SPCLMON THEN                                       %114-      15306100 C  0107:027:3
          EMITV(TALL.ADDRESS)                                 %114-      15306200 C  0107:027:1
        ELSE                                                  %114-      15306300 C  0107:027:0
          EMITN(TALL.ADDRESS)                                 %114-      15306400 C  0107:027:0
    ELSE % MULTI-DIMENSIONED CASE                             %114-      15306500 C  0107:027:3
      EMITO(IF SPCLMON THEN COC ELSE CDC);                    %114-      15306600 C  0107:027:0
    IF P1 = FS THEN % EXPECTED AN ASSIGNMENT                  %114-      15306700 C  0107:027:1
      ERREXIT(210);                                           %114-      15306800 C  0107:027:3
    STACKCT := 1; % BECAUSE REGISTERS ARE NON-EMPTY           %114-      15306900 C  0107:028:2
```

```
                    IF PARTIALWORD THEN                                  %114-        15307000 P  0107;028;0
                      BEGIN                                              %114-        15307100 C  0107;028;1
                        EMITI(0,T1,T2);                                  %114-        15307200 C  0107;028;2
                        P1 := 0;                                         %114-        15307300 C  0107;028;1
                      END;                                               %114-        15307400 C  0107;028;3
                  END OF NO ASSIGNMENT CASE;                             %114-        15307500 C  0107;028;3
                END OF SUBSCRIPTED VARIABLES                                        15376000 T  0107;028;3
                ELSE                                                                15376100 T  0107;028;3
                BEGIN COMMENT LABELID;                                              15376200 T  0107;028;3
                T1:=TAKE(T2:=GIT(TALL));                                            15376300 T  0107;028;0
                PUT(L,T2);                                                          15376400 T  0107;028;0
                IF T1=0 THEN T1:=L;                                                 15376500 T  0107;028;2
                IF (T1-L-T1) DIV 4 > 127 THEN BEGIN T1-0;FLAG(50);END;              15376600 T  0107;029;2
                EMIT(T1*4+3);                                                       15376700 T  0107;029;3
                STEPIT;                                                             15376800 T  0107;029;2
                END OF LABELID;                                                     15376900 T  0107;029;3
          EXIT ;  END  OF THE VARIABLE ROUTINE;                                     15377000 T  0107;029;3
                                                                          107 IS  302 LONG, NEXT SEG    3
```

```
          COMMENT THIS SECTION GENERATES CODE FOR STREAM PROCEDURES;                16000000 T  0003;062;1
          COMMENT  DO  LABEL DECS UPON APPEARANCE OF LABEL  ;                       16000050 T  0003;062;1
          PROCEDURE DECLARELABEL ;                                                  16000100 T  0003;062;1
PRT(733) = DECLARELABEL
                    BEGIN                                                           16000200 T  0003;062;1
                    KLASSF - STLABID;                                               16000300 T  0003;062;1
                    VONF - FORMALF - FALSF;                                         16000400 T  0003;062;3
                    ADDRSF - 0;                                                     16000500 T  0003;063;2
                    MAKEUPACCUM;  E; PUTNBUMP(0);                                   16000600 T  0003;063;3
                    FLBAT[I] - ACCUM[0]& LASTINFO[35:35:13];                        16000700 T  0003;063;3
                    END;                                                            16000800 T  0003;063;0
```

```
PROCEDURE STREAMSTMT ;                                                              16001000 T  0003;063;0
        BEGIN                                                                       16002000 T  0003;063;0
        DEFINE LFTPAREN=LEFTPAREN#,LOC=[36:12]#,LASTGT=[24:12]#,                    16003000 T  0003;063;0
                                                                        START OF SEGMENT ********** 108
            LOCFLD=36:36:12#,LGTFLD=24:24:12#;                                      16004000 T  0108;000;0
        DEFINE LEVEL=LVL#,ADDOP=ADOP#;                                              16005000 T  0108;000;0
DEFINE                                                                              16006000 T  0108;000;0
        JFW  =  39#,   COMMENT  7.5.5.1  JUMP FORWARD UNCONDITIONAL  ;              16007000 T  0108;000;0
         RCA =  40#, COMMENT    7.5.7.6  RECALL CONTROL ADDRESS      ;              16008000 T  0108;000;0
        JRV  =  47#,  COMMENT   7.5.5.2  JUMP REVERSE UNCONDITIONAL  ;              16009000 T  0108;000;0
        CRF  =  35#,  COMMENT    7.5.10.6 CALL REPEAT FIELD          ;              16010000 T  0108;000;0
         BNS =  42#,  COMMENT   7.5.5.5  BEGIN LOOP                  ;              16011000 T  0108;000;0
        NOP  =   1#,  COMMENT                                        ;              16012000 T  0108;000;0
        ENS  =  41#,  COMMENT   7.5.5.6  END LOOP                    ;              16013000 T  0108;000;0
        TAN  =  30#,  COMMENT   7.5.3.7  TEST FOR ALPHAMERIC         ;              16014000 T  0108;000;0
        BIT  =  31#,  COMMENT   7.5.3.8  TEST BIT                    ;              16015000 T  0108;000;0
         JFC =  37#,  COMMENT   7.5.5.3  JUMP FORWARD CONDITIONAL    ;              16016000 T  0108;000;0
        SFD  =  06#,  COMMENT   7.5.7.8  SET DESTINATION ADDRESS     ;              16017000 T  0108;000;0
        RSA  =  43#,  COMMENT   7.5.7.4  RECALL SOURCE ADDRESS       ;              16018000 T  0108;000;0
        TRP  =  60#,  COMMENT   7.5.2.2  TRANSFER PROGRAM CHARACTERS ;              16019000 T  0108;000;0
```

```
                    BSS  =   3#,  COMMENT   7.5.6.6  SKIP SOURCE BIT                ;          16020000 T  0108:000:0
                    BSD  =   2#,  COMMENT   7.5.6.5  SKIP DESTINATION BITS          ;          16021000 T  0108:000:0
                    SEC  =  34#,     COMMENT   7.5.10.1  SET COUNT                  ;          16022000 T  0108:000:0
                    JNS  =  38#;   COMMENT  7.5.5.7  JUMP OUT LOOP                  ;          16023000 T  0108:000:0
               PROCEDURE ADJUST;;                                                              16023100 T  0108:000:0
PRT(734) = ADJUST



          COMMENT FIXC EMITS  BASICLY FORWARD JUMPS, HOWEVER IN THE CASE                       16024000 T  0108:000:0
                  OF INSTRUCTIONS INTERPTED AS JUMPS BECAUSE OF A CRF ON                       16025000 T  0108:000:0
                  A VALUE = 0 AND THE JUMP ≥ 64 SYLLABLES A JFW 1 AND                          16026000 T  0108:000:0
                  A RCA L (L IS STACK ADDRESS OF A PSEUDO LABEL WHICH                          16027000 T  0108:000:0
                  MUST ALSO BE MANUFACTURED) IS EMITTED,  ;                                    16028000 T  0108:000:0
          PROCEDURE FIXC(S); VALUE S; REAL S;                                                  16029000 T  0108:000:0
PRT(735) = FIXC
                    BEGIN                                                                      16030000 T  0108:000:0
                    REAL SAVL,D,F;                                                             16031000 T  0108:000:0
                                                                                  START OF SEGMENT **********  109
STACK(F+2) = SAVL
STACK(F+3) = D
STACK(F+4) = F
                    IF D← (SAVL←L) - (L←S)-1 ≤ 63 THEN EMITC(D,GET(S))                         16032000 T  0109:000:0
                        ELSE  FLAG(700);                                                       16033000 T  0109:000:1
                    L←SAVL ;                                                                   16034000 T  0109:000:1
                    END  FIXC  ;                                                               16057000 T  0109:000:2
                                                                                    109 IS    10 LONG, NEXT SEG  108



          COMMENT EMITJUMP IS CALLED BY GOTOS AND JUMPCHAIN,                                   16058000 T  0108:000:0
                  THIS ROUTINE WILL EMIT A JUMP IF THE DISTANCE IS ≤ 63                        16059000 T  0108:000:0
                  SYLLABLES ,OTHERWISE, IT GETS A PRT CELL AND STUFFS THE                      16060000 T  0108:000:0
                  STACK ADDRESS INTO THE LABEL ENTRY IN INFO AND EMITS AN                      16061000 T  0108:000:0
                  RCA ON THIS STACK CELL. AT EXECUTION TIME ACTUAL PARAPART                    16062000 T  0108:000:0
                  INSURES US THAT THIS CELL WILL CONATIN A LABEL DESCRIPTOR                    16063000 T  0108:000:0
                  POINTING TO OUR LABEL IN QUESTION.  ;                                        16064000 T  0108:000:0
          PROCEDURE EMITJUMP( E); VALUE E; REAL E;                                             16065000 T  0108:000:0
PRT(736) = EMITJUMP
                    BEGIN                                                                      16066000 T  0108:000:0
                    REAL T,D;                                                                  16067000 T  0108:000:0
                                                                                  START OF SEGMENT **********  110
STACK(F+2) = T
STACK(F+3) = D
                    REAL ADDR;                                                                 16068000 T  0110:000:0
STACK(F+4) = ADDR
                    IF ABS(                                                                    16069000 T  0110:000:0
                    D←(T←TAKE(GIT(E)).LOC)-L-1)≥64 THEN                                        16070000 T  0110:000:0
                        FLAG(700)                                                              16071000 T  0110:000:1
                    ELSE EMITC(D,IF D <0 THEN JRV ELSE JFW);                                   16079000 T  0110:000:2
                    END  EMIT JUMP;                                                            16080000 T  0110:000:2
                                                                                    110 IS    12 LONG, NEXT SEG  108



          COMMENT   WHEN JUMPCHAIN IS CALLED THERE IS A LINKEDLIST IN THE CODE              16081000 T  0108:000:0
```

```
                         ARRAY WHERE JFWS MUST BE PLACED. THE 1ST LINK IS POINTED          16082000 T  0108:000:0
                         TO BY THE LOC FIELD OF EACH LABEL ENTRY IN INFO. THE LAST         16083000 T  0108:000:0
                         LINK IS = 4096.  ;                                               16084000 T  0108:000:0
                    PROCEDURE JUMPCHAIN( E); VALUE E;REAL E;                              16085000 T  0108:000:0
PRT(737) = JUMPCHAIN
                    BEGIN                                                                 16086000 T  0108:000:0
                    REAL SAVL ,LINK;                                                      16087000 T  0108:000:0
                                                                                  START OF SEGMENT **********  111
STACK(F+2) = SAVL
STACK(F+3) = LINK
                    SAVL ← L;                                                             16088000 T  0111:000:0
                    L ←  TAKE(GIT(E)).LASTGT ;                                            16089000 T  0111:000:1
                    WHILE L# 4095 DO                                                      16090000 T  0111:003:2
                         BEGIN                                                            16091000 T  0111:000:0
                         LINK ← GET(L);                                                   16092000 T  0111:000:0
                         EMITJUMP( E);                                                    16093000 T  0111:000:3
                         L ← LINK                                                         16094000 T  0111:000:0
                         END;                                                             16095000 T  0111:000:0
                    L←SAVL;                                                               16096000 T  0111:000:2
                    END  JUMPCHAIN  ;                                                     16097000 T  0111:000:3
                                                                                 111 IS   12 LONG, NEXT SEG  108



               COMMENT NESTS COMPILES THE NEST STATEMENT.                                 16098000 T  0108:000:0
                         A VARIABLE NEST INDEX CAUSES THE CODE,                           16099000 T  0108:000:0
                           CRF V, BNS 0 ,NOP,NOP, TO BE GENERATED INITIALLY.              16100000 T  0108:000:0
                         AT THE RIGHT PAREN THE BNS IS FIXED WITH THE LENGTH OF           16101000 T  0108:000:0
                         THE NEST (NUMBER OF SYLLABLES) IF THE LENGTH ≤63,OTHERWISE       16102000 T  0108:000:0
                         IT IS FIXED WITH A 1 AND THE NOPS REPLACED WITH JFW 1,           16103000 T  0108:000:0
                         RCA P.  THIS IS DONE BECAUSE THE VALUE OF V AT EXECUTION         16104000 T  0108:000:0
                         MAY = 0 AND THIS CODE CAUSES A JUMP AROUND THE NEST.             16105000 T  0108:000:0
                         JUMPOUT INFO IS REMEMBERED IN A RECURSIVE CELL AND               16106000 T  0108:000:0
                         NEST LEVEL INCREASED BY ONE.                                     16107000 T  0108:000:0
                         WHEN THE RIGHT PAREN IS REACHED,(IF THE STATEMENTS IN            16108000 T  0108:000:0
                         THE NEST COMPILED), JOINFO IS CHECKED FOR THE EXISTANCE          16109000 T  0108:000:0
                         OF JUMPOUT STATEMENTS IN THE NEST,IF SO,THE THE JUMPS            16110000 T  0108:000:0
                         ARE FIXED BY FAKING TOTOS INTO COMPILING THE REQUIRED            16111000 T  0108:000:0
                         JUMPS.                                                           16112000 T  0108:000:0
                         FINALLY THE BNS IS FIXED,IF REQUIRED,AND NEST LEVEL              16113000 T  0108:000:0
                         AND JOINFO RESTORED TO THEIR ORIGINAL VALUES.  ;                 16114000 T  0108:000:0
                    PROCEDURE NESTS;                                                      16115000 T  0108:000:0
PRT(740) = NESTS
                    BEGIN                                                                 16116000 T  0108:000:0
                    LABEL EXIT;                                                           16117000 T  0108:000:0
                                                                                  START OF SEGMENT **********  112
                    REAL JOINT,BNSFIX;                                                    16118000 T  0112:000:0
STACK(F+2) = JOINT
STACK(F+3) = BNSFIX
                    IF ELCLASS#LITNO THEN                                                 16119000 T  0112:000:0
                         BEGIN                                                            16120000 T  0112:000:1
                         EMITC(ELBAT[I],ADDRESS,CRF); BNSFIX← L;                          16121000 T  0112:001:2
                         EMIT(BNS);                                                       16122000 T  0112:000:3
                         END                                                              16123000 T  0112:000:1
                    ELSE EMITC(ELBAT[I].ADDRESS,BNS);                                     16124000 T  0112:000:1
                    IF STEPI # LFTPAREN THEN BEGIN ERR(262); GO TO EXIT END;              16125000 T  0112:000:1
```

```
     NESTLEVEL←NESTLEVEL + 1;                                                16126000 T  0112:0009:3
     JOINT ← JOINFO;                                                         16127000 T  0112:0010:1
     JOINFO ← 0;                                                             16128000 T  0112:0011:3
     DO   BEGIN                                                              16129000 T  0112:0012:0
          STEPIT; ERRORTOG ← TRUE; STREAMSTMT                                16130000 T  0112:0013:2
          END UNTIL ELCLASS ≠ SEMICOLON ;                                    16131000 T  0112:0014:0
     IF ELCLASS ≠ RTPAREN THEN BEGIN ERR(262);GO TO EXIT END;                16132000 T  0112:0016:0
     EMIT (  ENS);                                                           16133000 T  0112:0018:1
     IF JOINFO ≠ 0 THEN                                                      16134000 T  0112:0019:2
          BEGIN                                                              16135000 T  0112:0020:0
COMMENT   PREPARE TO CALL JUMPCHAIN FORJUMPOUTS;                             16136000 T  0112:0020:1
          ADJUST;                                                            16137000 T  0112:0020:1
          PUT(TAKE(GIT(JOINFO))&L[LOCFLD],GIT(JOINFO));                      16138000 T  0112:0021:2
          JUMPCHAIN(TAKE(JOINFO)&JOINFO[35:35:13]);                          16139000 T  0112:0024:1
          END;                                                               16140000 T  0112:0026:1
     IF BNSFIX ≠ 0 THEN FIXC(BNSFIX);                                        16141000 T  0112:0026:1
     NESTLEVEL ← NESTLEVEL-1;                                                16142000 T  0112:0028:1
     JOINFO ← JOINT ;                                                        16143000 T  0112:0030:0
EXIT; END  NESTS  ;                                                          16144000 T  0112:0030:1
                                                                      112 IS   34 LONG, NEXT SEG  108


COMMENT   LABELS HANDLES STREAM LABELS.                                      16145000 T  0108:0000:0
          ALL LABELS ARE ADJUSTED TO THE BEGINING OF THE NEXT                16146000 T  0108:0000:0
          WORD (IN THE PROGRAMSTREAM).                                       16147000 T  0108:0000:0
          IF A GO TO HAS NOT BEEN ENCOUNTERED BEFORE THE LABEL               16148000 T  0108:0000:0
          THEN THE NEST LEVEL FIELD IS ENTERED AND THE DEFINED BIT.          16149000 T  0108:0000:0
          [1:1], SET TO ONE. FOR DEFINED LABELS.IF WHERE A GO TO             16150000 T  0108:0000:0
          HAS APPEARED, A CHECK IS MADE THAT THE CURRENT NEST LEVEL          16151000 T  0108:0000:0
          MATCHES THE LEVEL OF THE LABEL.                                    16152000 T  0108:0000:0
          MULTIPLE OCCURANCES ARE ALSO CHECKED FOR AND FLAGGED.              16153000 T  0108:0000:0
          FINALLY,JUMPCHAIN IS CALLED TO FIX UP ANY FORWARD GO TOS           16154000 T  0108:0000:0
          AND GET A PRT LOCATION FOR ANY JUMPS ≥64 SYLLABLES.  ;             16155000 T  0108:0000:0
     PROCEDURE LABELS;                                                       16156000 T  0108:0000:0
PRT(741) = LABELS
               BEGIN                                                         16157000 T  0108:0000:0
               REAL GT1;                                                     16157100 T  0108:0000:0
                                                                  START OF SEGMENT **********  113
STACK(F+2) = GT1
               ADJUST;                                                       16158000 T  0113:0000:0
               GT1 ← ELBAT[I];                                               16159000 T  0113:0000:1
               XMARK(LBLREF); % MARK LABEL OCCURENCE FOR XREF          %110- 16159100 C  0113:0001:3
               IF STEPI ≠ COLON THEN ERR(258)                                16160000 T  0113:0005:3
               ELSE                                                          16161000 T  0113:0007:3
                    BEGIN                                                    16162000 T  0113:0008:0
                    IF TAKE(GT2←GIT(GT1)).LOC ≠ 0 THEN FLAG(259);            16163000 T  0113:0008:1
                    IF GT1>0 THEN                                            16164000 T  0113:0012:1
                         BEGIN                                               16165000 T  0113:0013:2
                         PUT(-(TAKE(GT1)&NESTLEVEL[11:43:5]),GT1);           16166000 T  0113:0013:3
                         PUT(-L,GT2)                                         16167000 T  0113:0016:1
                         END                                                 16168000 T  0113:0017:2
                    ELSE                                                     16169000 T  0113:0017:3
                         BEGIN                                               16170000 T  0113:0017:3
                         IF GT1.LEVEL≠NESTLEVEL THEN FLAG(257);              16171000 T  0113:0018:0
                         PUT((-L)&TAKE(GT2)[LGTFLD],GT2);                    16172000 T  0113:0020:1
                         JUMPCHAIN(GT1);                                     16173000 T  0113:0023:3
```

```
                    END;                                                                    16174000 T  0113:002410
                END                                                                         16175000 T  0113:002410
            ;  STEPIT;                                                                      16176000 T  0113:002410
            END  LABELS  ;                                                                  16177000 T  0113:002411
                                                                                      113 IS   27 LONG, NEXT SEG  108



        COMMENT   IFS COMPILES IF STATEMENTS,                                               16178000 T  0108:000010
                  FIRST THE TEST IS COMPILED, NOTE THAT IN THE                              16179000 T  0108:000010
                  CONSTRUCTS  "SC RELOP DC" AND  "SC RELOP STRING" THAT                     16180000 T  0108:000010
                  THE SYLLABLE EMITTED IS FETCHED FROM ONE OF TWO FIELDS                    16181000 T  0108:000010
                  IN THE ELBAT WORD FOR THE RELATIONAL OPERATOR, OTHERWISE                   16182000 T  0108:000010
                  THE CODE IS EMITTED STRAIGHTAWAY.                                         16183000 T  0108:000010
                  A TEST IS MADE TO SEE WHETHER THE STATEMENT AFTER THE                     16184000 T  0108:000010
                  "THEN" COULD POSSIBLY BE LONGER THAN 63 SYLLABLES, AND IF                 16185000 T  0108:000010
                  SO, Z NOPS ARE EMITTED FOR FIXC IN CASE A RCA WILL HAVE                   16186000 T  0108:000010
                  TO BE GENERATED.                                                          16187000 T  0108:000010
                  THIS PROCEDURE DOES NO OPTIMAZATION IN THE CASES                          16188000 T  0108:000010
                  IF THEN GO TO L, IF THEN STATEMENT ELSE GO TO L, OR                       16189000 T  0108:000010
                  IF THEN GO TO L1 ELSE GO TO L2   ;                                        16190000 T  0108:000010
        PROCEDURE  IFS;   BEGIN                                                             16191000 T  0108:000010
PRT(742) = IFS
                  DEFINE COMPARECODE=[42:6]#, TESTCODE=[36:6]#;                             16192000 T  0108:000010
                                                                                START OF SEGMENT **********  114
                  LABEL  IFSB, IFTOG, IFSC, EXIT;                                           16193000 T  0114:000010
                  SWITCH IFSW ← IFSB, IFTOG, IFSC;                                          16194000 T  0114:000010
                  REAL ADDR, FIX1, FIX2  ;                                                  16195000 T  0114:000411
STACK(F+2) = ADDR
STACK(F+3) = FIX1
STACK(F+4) = FIX2
                  ADDR←1  ;                                                                 16196000 T  0114:000411
                  GO TO IFSW[STEPI -SBV+1] ;                                                16197000 T  0114:000513
                  IF ELCLASS=LOCLID THEN                                                    16198000 T  0114:000912
                       BEGIN                                                                16199000 T  0114:000913
                       EMITC(ELBAT[I], ADDRESS, CRF);                                       16200000 T  0114:001010
                       ADDR←0;                                                              16201000 T  0114:001210
                       END                                                                  16202000 T  0114:001211
                  ELSE                                                                      16203000 T  0114:001211
                       IF ELCLASS=LITNO THEN ADDR ← ELBAT[I], ADDRESS                       16204000 T  0114:001211
                  ELSE BEGIN ERR(250); GO TO EXIT END;                                      16205000 T  0114:001411
                  IF STEPI ≠ SCV THEN BEGIN ERR(263);GO TO EXIT END;                        16206000 T  0114:001713
           IFSC;  IF STEPI ≠ RELOP THEN BEGIN ERR(264);GO TO EXIT END;                      16207000 T  0114:002011
                  IF STEPI = DCV THEN EMITC( ADDR, ELBAT[I-1], COMPARECODE)                 16208000 T  0114:002313
                  ELSE                                                                      16209000 T  0114:002611
                       IF ELCLASS = STRNGCON THEN                                           16210000 T  0114:002713
                            EMITC(ACCUM[1], [18:6], ELBAT[I-1], TESTCODE)                   16211000 T  0114:002811
                  ELSE                                                                      16212000 T  0114:003112
                  IF ELCLASS=LITNO THEN EMITC(C, ELBAT[I-1], TESTCODE) ELSE                 16212500 T  0114:003210
                  IF ELCLASS≤IDMAX AND Q="5ALPHA" THEN EMITC(17, TAN)                       16213000 T  0114:003610
                  ELSE BEGIN ERR(265);GO TO EXIT END;                                       16214000 T  0114:003913
                  GO TO IFTOG  ;                                                            16215000 T  0114:004312
           IFSB;  EMITC(1, BIT);                                                            16216000 T  0114:004313
          IFTOG;  IF STEPI ≠ THENV THEN BEGIN ERR(266); GO TO EXIT  END;                    16217000 T  0114:004512
                  FIX1 ← L;                                                                 16218000 T  0114:004713
                  EMIT(JFC);                                                                16219000 T  0114:004811
```

```
                    IF STEPI#ELSEV THEN%                                                    16220000 T  0114:0049:2
                    STREAMSTMT;                                                             16229000 T  0114:0050:0
                    IF ELCLASS= ELSEV THEN                                                  16230000 T  0114:0051:2
                         BEGIN                                                              16231000 T  0114:0052:0
                         FIX2 ← L;  EMIT(JFW);                                              16232000 T  0114:0052:1
                         FIXC(FIX1);                                                        16233000 T  0114:0054:0
                         STEPIT;                                                            16234000 T  0114:0054:1
                         STREAMSTMT;                                                        16235000 T  0114:0055:2
                         FIXC(FIX2);                                                        16236000 T  0114:0055:3
                         END                                                                16237000 T  0114:0056:1
                    ELSE FIXC(FIX1);                                                        16238000 T  0114:0056:1
               EXIT:END   IFS  ;                                                            16239000 T  0114:0057:3
                                                                                  114 IS   61 LONG, NEXT SEG  108


               COMMENT GOTOS HANDLES GO TO AND THE LAST PART OF JUMP OUT TO                 16240000 T  0108:0000:0
                         STATEMENTS.                                                        16241000 T  0108:0000:0
                         IF THE LABEL HAS BEEN ENCOUNTERED THEN EMITJUMP IS CALLED          16242000 T  0108:0000:0
                         AN PRODUCES A JRV OR RCA IN THE CASE OF JUMPS≥64 SYLLABL           16243000 T  0108:0000:0
                         ES. OTHERWISE, A LINK IS EMITTED POINTING ANY PREVIOUS             16244000 T  0108:0000:0
                         GO TOS IN THE CASE OF FORWARD JUMPS.                               16245000 T  0108:0000:0
                         FINALLY, IF THE NEST LEVEL IS DEFINED THEN IT IS CHECKED           16246000 T  0108:0000:0
                         AGAINST THE CURRENT LEVEL MINUS THE NUMBER OF LEVELS TO            16247000 T  0108:0000:0
                         BE JUMPED OUT. OTHERWISE,NEST LEVEL IS DEFINED.  ;                 16248000 T  0108:0000:0
               PROCEDURE GOTOS;                                                             16249000 T  0108:0000:0
PRT(743) = GOTOS
               BEGIN                                                                        16250000 T  0108:0000:0
               LABEL EXIT;                                                                  16251000 T  0108:0000:0
                                                                                 START OF SEGMENT **********  115
               IF STEPI ≠TOV THEN I←I-1 ;                                                   16252000 T  0115:0000:0
                    IF STEPI ≠ STLABID THEN IF ELCLASS ≤ IDMAX THEN                         16253000 T  0115:0002:1
                         DECLARELABEL ELSE BEGIN ERR(260); GO TO EXIT END;                  16253100 T  0115:0005:2
               IF(GT2←TAKE(GIT(GT1←ELBAT[I]))),MON=1                                        16254000 T  0115:0007:3
               OR GT2.LOC≠0 THEN EMITJUMP(GT1)                                              16255000 T  0115:0010:1
               ELSE                                                                         16256000 T  0115:0013:3
                 BEGIN   PUT(0&L[24:36:12],GIT(GT1));                                       16257000 T  0115:0014:0
                    IF GT1>0 THEN                                                           16258000 T  0115:0017:2
                         BEGIN                                                              16259000 T  0115:0017:3
                         PUT(=(TAKE(GT1)&(NESTLEVEL-JUMPLEVEL)[11:43:5]),GT1);              16260000 T  0115:0018:0
                         EMITN(1023);                                                       16261000 T  0115:0021:2
                         END                                                                16262000 T  0115:0022:0
                    ELSE                                                                    16263000 T  0115:0022:0
                         BEGIN                                                              16264000 T  0115:0022:0
                         IF GT1.LEVEL ≠ NESTLEVEL-JUMPLEVEL THEN FLAG(257);                 16265000 T  0115:0022:1
                         EMIT(GT2,LASTGT);                                                  16266000 T  0115:0025:3
                    END;                                                                    16267000 T  0115:0027:2
                 END;                                                                       16268000 T  0115:0027:2
               JUMPLEVEL←0 ;                                                                16269000 T  0115:0027:2
          EXIT: END  GOTOS   ;                                                              16270000 T  0115:0027:3
                                                                                  115 IS   29 LONG, NEXT SEG  108


           COMMENT  RELEASES COMPILES THE STREAM RELEASE STATEMENT.                         16271000 T  0108:0000:0
```

```
                THE CODE GENERATED IS :
                     SED  FILE
                     RSA  0,
                AT EXECUTION TIME THIS CAUSES AN INVALID ADDRESS WHICH IS
                INTERPETED BY THE MCP TO MEAN RELEASE THE FILE POINTED TO
                BY THE DESTINATION ADDRESS.
                THE MONITOR BIT IS SET IN INFO FOR THE LOCAL VARIABLE SO
                THAT ACUTAL PARAPART MAY BE INFORMED LATER THAT A FILE
                MUST BE PASSED FOR THIS FORMAL PARAMETER;

COMMENT         INDEXS  COMPILE STATEMENTS BEGINING WITH SI,DI,CI,TALLY
                OR LOCALIDS .
                THREE CASES PRESENT THEMSELVES,
                     LETING X BE EITHER OF  SI,DI,CI OR TALLY, THEY ARE:
                     CASE I    LOCLID ← X
                     CASE II   X ← X ...
                     CASE III  X ← EITHER LOC,LOCLID,SC OR DC.
                THE VARIABLE "INDEX" IS COMPUTED,DEPENDING UPON WHICH
                CASE EXISTS,SUCH THAT ARRAY ELEMENT "MACRO[INDEX]"CONTAINS
                THE CODE TO BE EMITTED.
                EACH ELEMENT OF MACRO HAS  1-3 SYLLABLES ORDERED FROM
                RIGHT TO LEFT, UNUSED SYLLABLES MUST = 0.     EACH MACRO
                MAY REQUIRE AT MOST ONE REPEAT PART.
                IN THIS PROCEDURE,INDEXS,THE VARIABLE "ADDR" CONTAINS THE
                PROPER REPEAT PART BY THE TIME THE LABEL "GENERATE" IS
                ENCOUNTERED. THE SYLLABLES ARE FETCHED FROM MACRO[TYPE]
                ONE AT A TIME AND IF THE REPEAT PART ≠ 0 THEN"ADDR" IS
                USED AS THE REPEAT PART,THUS BUILDING A SYLLABLE WITH
                THE PROPER  ADDRESS AND OPERATOR .
                NOTE: IF MACRO[TYPE] = 0 THEN THIS SIGNIFIES A SYNTAX
                      ERROR.                                          ;
          PROCEDURE INDEXS;
PRT(744) = INDEXS
                BEGIN
                LABEL EXIT,GENERATE,L,L1;

                INTEGER TCLASS,INDEX,ADDR,J;
STACK(F+2) = TCLASS
STACK(F+3) = INDEX
STACK(F+4) = ADDR
STACK(F+5) = J

                TCLASS ← ELCLASS ;
                IF STEPI ≠ ASSIGNOP THEN BEGIN ERR(251); GO TO EXIT END;
                IF TCLASS = LOCLID THEN
                     BEGIN
                     XMARK(ASSIGNREF);                                  %110-
                     IF SIV>STEPI OR ELCLASS>TALLYV THEN GO TO L;
                     INDEX ← 32 + ELCLASS-SIV;
                     ADDR ← ELBAT[I-2].ADDRESS;
```

```
16272000 T  0108:0000:0
16273000 T  0108:0000:0
16274000 T  0108:0000:0
16275000 T  0108:0000:0
16276000 T  0108:0000:0
16277000 T  0108:0000:0
16278000 T  0108:0000:0
16279000 T  0108:0000:0
16280000 T  0108:0000:0
16281000 T  0108:0000:0
16282000 T  0108:0000:0
16283000 T  0108:0000:0
16284000 T  0108:0000:0
16285000 T  0108:0000:0
16286000 T  0108:0000:0
16287000 T  0108:0000:0
16288000 T  0108:0000:0
16289000 T  0108:0000:0
16290000 T  0108:0000:0
16291000 T  0108:0000:0
16292000 T  0108:0000:0
16293000 T  0108:0000:0
16294000 T  0108:0000:0
16295000 T  0108:0000:0
16296000 T  0108:0000:0
16297000 T  0108:0000:0
16298000 T  0108:0000:0
16299000 T  0108:0000:0
16300000 T  0108:0000:0
16301000 T  0108:0000:0
16302000 T  0108:0000:0
16303000 T  0108:0000:0
16304000 T  0108:0000:0
16305000 T  0108:0000:0
16306000 T  0108:0000:0
16307000 T  0108:0000:0
16308000 T  0108:0000:0
16309000 T  0108:0000:0
16310000 T  0108:0000:0
16311000 T  0108:0000:0

16312000 T  0108:0000:0
16313000 T  0108:0000:0
START OF SEGMENT ********** 116
16314000 T  0116:0000:0

16315000 T  0116:0000:0
16316000 T  0116:0000:1
16317000 T  0116:0003:3
16318000 T  0116:0004:0
16318500 C  0116:0004:1
16319000 T  0116:0009:2
16320000 T  0116:0011:3
16321000 T  0116:0013:3
```

```
                    GO TO GENERATE;                                                                  16322000 T  0116:0015:3
                    END;                                                                             16323000 T  0116:0016:0
               IF TCLASS = STEPI THEN                                                                16324000 T  0116:0016:0
                    BEGIN                                                                            16325000 T  0116:0017:2
                    IF STEPI ≠ ADDOP OR STEPI≠ LITNO AND ELCLASS ≠ LOCLID THEN                       16326000 T  0116:0017:3
                         GO TO L;                                                                    16327000 T  0116:0019:3
                    INDEX ← TCLASS-SIV                                                               16328000 T  0116:0021:3
                         +REAL(ELBAT[I-1].ADDRESS=SUB) × 4                                           16329000 T  0116:0022:0
                         + REAL(ELCLASS =LOCLID) × 8;                                                16330000 T  0116:0024:0
                    END                                                                              16331000 T  0116:0027:2
               ELSE                                                                                  16332000 T  0116:0027:2
                    BEGIN                                                                            16333000 T  0116:0027:2
                    INDEX ← TCLASS -SIV                                                              16334000 T  0116:0027:3
                        + ( IF ELCLASS = LOCLID THEN 16 ELSE                                         16335000 T  0116:0028:0
                         IF ELCLASS = LOCV THEN 20 ELSE                                              16336000 T  0116:0030:0
                         IF ELCLASS = SCV THEN 24 ELSE                                               16337000 T  0116:0032:0
                         IF ELCLASS= DCV  THEN 28 ELSE 25);                                          16338000 T  0116:0034:0
                    IF ELCLASS = LOCV THEN                                                           16339000 T  0116:0037:2
                         IF STEPI ≠ LOCLID THEN GO TO L;                                             16340000 T  0116:0038:0
                    IF ELCLASS = LITNO AND TCLASS = TALLYV THEN                                      16341000 T  0116:0040:0
                         BEGIN EMITC(ELBAT[I].ADDRESS,SEC);GO TO EXIT END;                           16342000 T  0116:0041:3
                    END ;                                                                            16343000 T  0116:0044:1
               ADDR ← ELBAT[I].ADDRESS;                                                              16344000 T  0116:0044:1
GENERATE;                                                                                            16345000 T  0116:0046:0
               IF MACRO[INDEX]= 0 THEN                                                               16346000 T  0116:0046:0
  L:                BEGIN ERR(250);GO TO EXIT END;                                                   16347000 T  0116:0047:2
               J ← 8; TCLASS ←0 ;                                                                    16348000 T  0116:0049:2
 L1: MOVECHARACTERS(2,MACRO[INDEX],J←J-2,TCLASS,6 );                                                 16349000 T  0116:0050:1
               IF TCLASS≠0 THEN                                                                      16350000 T  0116:0054:0
                    BEGIN                                                                            16351000 T  0116:0054:1
                    EMITC(IF TCLASS≥64 THEN ADDR ELSE 0,TCLASS);                                     16352000 T  0116:0055:2
                    GO TO L1                                                                         16353000 T  0116:0058:0
                    END;                                                                             16354000 T  0116:0058:1
               EXIT:END  INDEXS ;                                                                    16355000 T  0116:0058:1
                                                                                                  116 IS   62 LONG, NEXT SEG  108


               COMMENT  DSS COMPILES DESTINATION STREAM STATEMENTS,                                  16356000 T  0108:0000:0
                        DS←  LIT"STRING" IS HANDLED AS A SPECIAL CASE BECAUE THE                    16357000 T  0108:0000:0
                        STRING MUST BE SCANED FROM RIGHT TO LEFT,REPEATEDLY IF                      16358000 T  0108:0000:0
                        NECESSARY, AND EMITTED TO THE PROGRAM STREAM,  IN                           16359000 T  0108:0000:0
                        ALL OTHER CASES,THE ELBAT WORD CONTAINS THE OPERATOR IN                     16360000 T  0108:0000:0
                        THE OPCODE FIELD  ;                                                         16361000 T  0108:0000:0
          PROCEDURE DSS;                                                                            16362000 T  0108:0000:0
PRT(745) = DSS
                    BEGIN                                                                           16363000 T  0108:0000:0
                    INTEGER ADDR,J,K,L,T;                                                           16364000 T  0108:0000:0
                                                                                              START OF SEGMENT **********  117
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  0117:0000:0
                    DEFINE OPCODE=[27:6]#;                                                          16366000 T  0117:0000:0
                    IF STEPI ≠ ASSIGNOP THEN BEGIN ERR(251); GO TO EXIT END;                        16367000 T  0117:0000:0
```

```
                    IF STEPI = LOCLID THEN                                                  16368000 T  0117:0002:1
                         BEGIN                                                              16369000 T  0117:0003:3
                         EMITC(ELBAT[I],ADDRESS,CRF);                                       16370000 T  0117:0004:0
                         ADDR← 0;                                                           16371000 T  0117:0006:0
                         IF STEPI = LITV THEN GO TO L1                                      16372000 T  0117:0006:1
                         END                                                                16373000 T  0117:0007:3
                    ELSE IF ELCLASS= LITNO THEN                                             16374000 T  0117:0008:0
                         BEGIN                                                              16375000 T  0117:0009:3
                         ADDR ← ELBAT[I].ADDRESS; STEPIT ;                                  16376000 T  0117:0010:0
                         END                                                                16377000 T  0117:0012:0
                    ELSE ADDR ← 1 ;                                                         16378000 T  0117:0012:0
                    IF Q = "4FILLO" THEN EMITC(ADDR,10) ELSE                        %E      16378500 T  0117:0013:2
                    IF ELCLASS = TRNSFER THEN EMITC(ADDR,ELBAT[I].OPCODE)                   16379000 T  0117:0015:3
                    ELSE                                                                    16380000 T  0117:0019:2
                    IF ELCLASS = LITV THEN                                                  16381000 T  0117:0020:0
                         BEGIN                                                              16382000 T  0117:0021:2
                         EMITC(ADDR,TRP);                                                   16383000 T  0117:0021:3
                         IF STEPI≠STRNGCON THEN                                             16384000 T  0117:0022:1
                              BEGIN ERR(255);GO TO EXIT END;                                16384500 T  0117:0023:3
                         IF ADDR MOD 2 ≠ 0 THEN                                             16385000 T  0117:0025:3
                              BEGIN                                                         16386000 T  0117:0026:1
                              EMIT(ACCUM[1],[18:6]);  J ← 1;                                16387000 T  0117:0027:2
                              END ;                                                         16388000 T  0117:0029:3
                         FOR K ←J+2 STEP 2 UNTIL ADDR DO                                    16389000 T  0117:0029:3
                              BEGIN                                                         16390000 T  0117:0033:3
                              FOR L ←6,7 DO                                                 16391000 T  0117:0033:3
                              MOVECHARACTERS(1,ACCUM[1],2+(IF J←J+1>COUNT THEN J←1          16392000 T  0117:0038:0
STACK(F+7) = *TEMPORARY STORAGE*
                                                  ELSE J),T,L  );                           16393000 T  0117:0041:3
                              EMIT(T);                                                      16394000 T  0117:0044:1
                         END  END                                                           16395000 T  0117:0045:3
                    ELSE                                                                    16396000 T  0117:0046:0
                L1:      ERR(250);                                                          16397000 T  0117:0046:0
               EXIT:END  DSS  ;                                                             16398000 T  0117:0047:3
                                                                                  117 IS   52 LONG, NEXT SEG  108



               COMMENT SKIPS  COMPILES THE SKIP BIT STATEMENT,                              16399000 T  0108:0000:0
                         IF THE REPEAT INDEX IS A LOCALID THEN A CRF IS EMITTED,            16400000 T  0108:0000:0
                         A  BSS OR BSD IS THEN EMITTED FOR SKIP SOURCE BITS (SB)            16401000 T  0108:0000:0
                         OR   SKIP DESTINATION BITS (DB) RESPECTIVELY ;                     16402000 T  0108:0000:0
                    PROCEDURE SKIPS ;                                                       16403000 T  0108:0000:0
PRT(746) = SKIPS
                         BEGIN                                                              16404000 T  0108:0000:0
                         REAL ADDR;                                                         16405000 T  0108:0000:0
                                                                                  START OF SEGMENT **********  118
STACK(F+2) = ADDR
                         IF STEPI = LOCLID THEN                                             16406000 T  0118:0000:0
                              BEGIN                                                         16407000 T  0118:0001:2
                              EMITC(ELBAT[I],ADDRESS,CRF); ADDR←0; STEPIT;                  16408000 T  0118:0001:3
                              END                                                           16409000 T  0118:0004:1
                         ELSE IF ELCLASS = LITNO THEN                                       16410000 T  0118:0004:1
                              BEGIN                                                         16411000 T  0118:0005:3
                              ADDR← ELBAT[I].ADDRESS; STEPIT                                16412000 T  0118:0006:0
                              END                                                           16413000 T  0118:0007:3
```

```
                    ELSE ADDR ← 1 ;                                                    16414000 T  0118:0008:0
                    IF ELCLASS =SBV THEN EMITC(ADDR,BSS)                               16415000 T  0118:0009:3
                    ELSE
                    IF ELCLASS =DBV THEN  EMITC(ADDR,BSD)                              16416000 T  0118:0011:2
                    ELSE ERR(250);                                                     16417000 T  0118:0011:3
                    END  SKIPS  ;                                                      16418000 T  0118:0014:0
                                                                                       16419000 T  0118:0015:3
                                                                          118 IS   18 LONG. NEXT SEG  108



               COMMENT JUMPS COMPILES JUMP OUT AND JUMP OUT TO STATEMENTS,              16420000 T  0108:0000:0
                       JUMP OUT TO STATEMENTS CAUSE JUMP LEVEL TO BE SET TO             16421000 T  0108:0000:0
                       THE NUMBER OF LEVELS SPECIFIED. THEN THIS NUMBER OF              16422000 T  0108:0000:0
                       JNS ARE EMITTED AND GOTOS IS CALLED TO COMPILE THE               16423000 T  0108:0000:0
                       JUMP INSTRUCTION.                                                16424000 T  0108:0000:0
                       SIMPLE JUMP OUTS ARE HANDLED BY EMITTING ONE JNS,ENTERING        16425000 T  0108:0000:0
                       A PSEUDO STLABID IN INFO AND SETTING ELBAT[I] SUCH THAT          16426000 T  0108:0000:0
                       THE GOTOS PROCEDURE WILL PERFORM THE ACTION OF SETTING           16427000 T  0108:0000:0
                       UP THE LINKS FOR LATER FIX UPS. THE NEST STATEMENT CAUSES        16428000 T  0108:0000:0
                       THESE FIX UPS(IF EMITTING OF JUMP INSTRUCTIONS) BY CALLING       16429000 T  0108:0000:0
                       GO TOS WHEN THE RIGHT PAREN IS ENCOUNTERED.     ;                16430000 T  0108:0000:0
                 PROCEDURE JUMPS;                                                       16431000 T  0108:0000:0
PRT(747) = JUMPS
                    BEGIN                                                               16432000 T  0108:0000:0
                    JUMPLEVEL←1;                                                        16433000 T  0108:0000:0
                         IF STEPI≠DECLARATORS THEN IF ACCUM[1]≠"3OUTOO" THEN            16434000 T  0108:0001:3
                                                      FLAG(261);                        16434100 T  0108:0004:0
                    IF STEPI = LITNO THEN JUMPLEVEL← ELBAT[I].ADDRESS                   16435000 T  0108:0005:3
                    ELSE BEGIN                                                          16436000 T  0108:0007:2
                         IF ELCLASS≠ TOV AND ELCLASS≠ STLABID THEN                      16437000 T  0108:0010:0
                         BEGIN                                                          16438000 T  0108:0011:3
               COMMENT   SIMPLE JUMP OUT STATEMENT;                                     16439000 T  0108:0012:0
                         IF JOINFO = 0 THEN                                             16440000 T  0108:0012:0
                              BEGIN                                                     16441000 T  0108:0013:2
                              JOINFO ← NEXTINFO ;                                       16442000 T  0108:0013:3
                              PUTNBUMP(STACKHEAD[0].LINK&(STLABID×2+1)                  16443000 T  0108:0014:0
                                   [2:40:8]&2[27:40:8 ]);                               16444000 T  0108:0016:1
                              PUTNBUMP(0&(JOINFO-LASTINFO    )[ 4:40:8]);             16445000 T  0108:0018:1
                              PUTNBUMP (0);                                             16446000 T  0108:0021:2
                              LASTINFO ← JOINFO;                                        16447000 T  0108:0021:3
                              END;                                                      16448000 T  0108:0022:1
                         ELBAT[I←  I-1]←  TAKE(JOINFO)&JOINFO[35:35:13];                16449000 T  0108:0022:1
                         END; I←I-1 ;                                                   16450000 T  0108:0026:0
                    END;                                                                16451000 T  0108:0027:3
                    FOR GT1← 1 STEP 1 UNTIL JUMPLEVEL DO                                16452000 T  0108:0027:3
                         EMIT(  JNS);                                                   16453000 T  0108:0029:2
                    GOTOS;                                                              16454000 T  0108:0032:0
                    END JUMPS;                                                          16455000 T  0108:0032:1



               COMMENT STREAMSTMT  ENVOKES THE APPROPRIATE PROCEDURE TO HANDLE          16456000 T  0108:0032:1
                       THE VARIOUS AND SUNDRY STREAM PROCEDURE STATEMENTS.              16457000 T  0108:0032:1
                       THE STATEMENTS ARE BROKEN DOWN AS FOLLOWS:                       16458000 T  0108:0032:1
                            IDENTIFIED BY          PROCEDURE ENVOKED                    16459000 T  0108:0032:1
```

```
                    END                         GO TO FINI                                  16460000 T  0108:0032:1
                    SEMICOLON                   GO TO FINI                                  16461000 T  0108:0032:1
                    )                           GO TO FINI                                  16462000 T  0108:0032:1
                    IF                          IFS                                         16463000 T  0108:0032:1
                    GO                          GOTOS                                       16464000 T  0108:0032:1
                    RELEASE                     RELEASES                                    16465000 T  0108:0032:1
                  BEGIN                         COMPOUNDTAIL                                16466000 T  0108:0032:1
                  SI,DI,CI,TALLY,LOCALID INDEXS                                             16467000 T  0108:0032:1
                  DS                            DSS                                         16468000 T  0108:0032:1
                  SKIP                          SKIPS                                       16469000 T  0108:0032:1
                    JUMP                        JUMPS                                       16470000 T  0108:0032:1
                    LABELID                     LABELS                                      16471000 T  0108:0032:1
                    LITERAL NO.,LOCALID(        NESTS                                       16472000 T  0108:0032:1
          UPON EXITING,STREAMSTMT ASSURES THAT "I" POINTS TO                                16473000 T  0108:0032:1
          THE SEMICOLON ,END OR ) IN SYNTACICALLY CORRECT PROGRAMS;                         16474000 T  0108:0032:1
     LABEL  L,L1,L2,L3,L4,L5,L6,L7,L8,L9,L10,EXIT,FINI,START;                               16475000 T  0108:0032:1
     SWITCH TYPE ← FINI,L,FINI,L3,L4,L5,L6,L7,L7,L7,L7,L8,L9,L10;                           16476000 T  0108:0032:1
START;    GO TO TYPE[ ELCLASS-ENDV+1];                                                    16477000 T  0108:0043:2
          IF ELCLASS= RTPAREN THEN GO TO FINI ;                                           16478000 T  0108:0047:2
          IF ELCLASS= STLABID THEN GO TO L2 ;                                             16481000 T  0108:0048:0
                                                                                          16482000 T  0108:0049:3
     IF ELCLASS <IDMAX AND ELCLASS≠LOCLID THEN BEGIN                                      16482100 T  0108:0049:3
          DECLARELABEL; GO TO L2;  END;                                                   16482200 T  0108:0051:3
     IF ELCLASS = LITNO OR ELCLASS = LOCLID AND TABLE(I+1)                                16482300 T  0108:0052:1
           = LFTPAREN THEN GO TO L1;                                                      16482400 T  0108:0055:2
     IF ELCLASS = LOCLID  THEN  GO TO L7;                                                 16482500 T  0108:0057:2
  L1;    ERR( 250 ); GO TO FINI ;                                                        16483000 T  0108:0058:1
         L1;   NESTS;    GO TO EXIT;                                                      16484000 T  0108:0060:0
         L2;   LABELS;   GO TO START;                                                     16485000 T  0108:0062:0
         L3;   IFS;      GO TO FINI;                                                      16486000 T  0108:0063:2
         L4;   GOTOS;    GO TO EXIT;                                                      16487000 T  0108:0064:0
    L5;                                                                                   16488000 T  0108:0065:2
         L6;   I←I+1 ; COMPOUNDTAIL; GO TO FINI;                                          16489000 T  0108:0065:2
         L7;   INDEXS;   GO TO EXIT;                                                      16490000 T  0108:0067:2
         L8;   DSS;      GO TO EXIT;                                                      16491000 T  0108:0069:2
         L9;   SKIPS;    GO TO EXIT;                                                      16492000 T  0108:0070:0
         L10;  JUMPS;    GO TO EXIT;                                                      16493000 T  0108:0071:2
       EXIT;   STEPIT;                                                                    16494000 T  0108:0072:0
       FINI;   END   STREAMSTMT;                                                          16495000 T  0108:0072:1
                                                                                  108 IS   74 LONG, NEXT SEG    3


          MOVE(1,1,CODE(0));                                                              16495100 T  0003:0636:0
          TIME1 ← TIME(1); PROGRAM;                                                       16495200 T  0003:0641:2
               ENDOFITALL;                                                      %108=     16495210 P  0003:0642:1
          IF (XREF OR DEFINING.[1:1]) AND XLUN > 0 THEN                         %110=     16495300 P  0003:0643:2
               BEGIN DEFINE LSS= <#,GTR=>#,NEQ= ≠#,LEQ=≤#;                      %DFB      17002000 C  0003:0645:3
PRT(750) = *SEGMENT DESCRIPTOR*
                                                                                  START OF SEGMENT **********  119
          DEFINE XREFINFO[INDEX] = INFO[((INDEX).CF DIV 2).[33:7],                %110=     17002005 C  0119:0000:0
                                        ((INDEX).CF DIV 2).LINKC]#,               %110=     17002006 C  0119:0000:0
                 CF = [33:15]#,                                                   %110=     17002007 C  0119:0000:0
                 FF = [18:15]#,                                                   %110=     17002008 C  0119:0000:0
                 NEWID[INDEX] = (IF BOOLEAN(INDEX) THEN XREFINFO[INDEX].FF                 17002009 C  0119:0000:0
                                 ELSE XREFINFO[INDEX].CF)#;                       %110=     17002010 C  0119:0000:0
```

```
                    ARRAY TIMINGS[0:2,0:3];                                  %110=        17002012 C  0119:000:0
PRT(751) = TIMINGS
                    PROCEDURE SAVETIMES(I);                                  %110=        17002015 C  0119:000:1
PRT(752) = SAVETIMES
                      VALUE I; INTEGER I;                                    %110=        17002020 C  0119:000:1
                    BEGIN                                                    %110=        17002025 C  0119:000:1
                      INTEGER J;                                             %110=        17002030 C  0119:000:1
                                                                                   START OF SEGMENT **********  120
STACK(F+2) = J
                      FOR J := 1 STEP 1 UNTIL 3 DO                           %110=        17002035 C  0120:000:0
                        TIMINGS[I,J] := TIME(J);                             %110=        17002040 C  0120:000:2
                    END;                                                     %110=        17002045 C  0120:000:3
                                                                                 120 IS    8 LONG, NEXT SEG  119



                    PROCEDURE UPDATETIMES(I);                                %110=        17002050 C  0119:000:1
PRT(753) = UPDATETIMES
                      VALUE I; INTEGER I;                                    %110=        17002055 C  0119:000:1
                    BEGIN                                                    %110=        17002060 C  0119:000:1
                      INTEGER J;                                             %110=        17002065 C  0119:000:1
                                                                                   START OF SEGMENT **********  121
STACK(F+2) = J
                      FOR J := 1 STEP 1 UNTIL 3 DO                           %110=        17002070 C  0121:000:0
                        TIMINGS[I,J] := TIME(J) - TIMINGS[I,J];              %110=        17002075 C  0121:000:2
                    END;                                                     %110=        17002080 C  0121:000:7
                                                                                 121 IS   10 LONG, NEXT SEG  119
```

```
                    CLOSE(CARD,RELEASE);                                     %108=        17002490 C  0119:000:1
                    CLOSE(TAPE,RELEASE);                                     %108=        17002500 C  0119:000:1
                                  LOCK(NEWTAPE);                             %108=        17002510 C  0119:000:0
                    WRITE(LINE[PAGE]);                                       %108=        17002520 C  0119:000:0
                    SAVETIMES(0); % SAVE TIMES FOR START OF IDENTIFIER SORT. %110=        17002525 C  0119:001:0
                   FOR XREFPT:=XREFPT STEP 1 UNTIL 29 DO XREFAY2[XREFPT]:=100000000;          17003000 C  0119:001:1
                   WRITE(DSK2,30,XREFAY2[*]);                                 %DFB        17004000 C  0119:001:3
                    TOTALNO := XLUN; % REMEMBER NUMBER OF IDENTIFIERS.       %110=        17004500 C  0119:002:3
                         XREFPT+XLUN+0;                                      %DFB        17004600 C  0119:002:1
                    FOR I := 0 STEP 1 UNTIL 8191 DO                          %110=        17004700 C  0119:002:3
                      PUT(0,I);                                              %110=        17004710 C  0119:002:0
                          BEGIN                                              %DFB        17005000 C  0119:002:2
                          BOOLEAN PROCEDURE INPUT1(A);                       %DFB        17006000 C  0119:002:2
PRT(754) = *SEGMENT DESCRIPTOR*
                                                                                   START OF SEGMENT **********  122
PRT(755) = INPUT1
                              ARRAY A[0];                                    %DFB        17007000 C  0122:000:0
                              BEGIN                                          %DFB        17008000 C  0122:000:0
                              LABEL L,EOF;                                   %DFB        17009000 C  0122:000:0
                                                                                   START OF SEGMENT **********  123
                              READ(DSK1,10,A[*])[EOF];                       %DFB        17010000 C  0123:000:0
PRT(756) = EOF
                              GO TO L;                                       %DFB        17011000 C  0123:000:2
                              EOF: INPUT1:=TRUE;                             %DFB        17012000 C  0123:000:3
                                   REWIND(DSK1);                             %DFB        17013000 C  0123:000:1
```

```
                L;                                                    %DFB        17014000 C  0123:0008:1
                END;                                                  %DFB        17015000 C  0123:0009:2
                                                                            123 IS   14 LONG, NEXT SEG  122


            PROCEDURE OUTPUT1(B,A);                                   %DFB        17016000 C  0122:0000:0
PRT(757) = OUTPUT1
              VALUE B;                                                %DFB        17017000 C  0122:0000:0
              BOOLEAN B;                                              %DFB        17018000 C  0122:0000:0
              ARRAY A[0];                                             %DFB        17019000 C  0122:0000:0
                 BEGIN                                                %DFB        17020000 C  0122:0000:0
                 IF B THEN                                            %DFB        17021000 C  0122:0000:0
                   BEGIN                                              %110-       17022000 C  0122:0000:0
                     REWIND(DSK1);                                    %110-       17022100 C  0122:0000:1
                     UPDATETIMES(0); % UPDATE TIMES FOR IDENTIFIER SORT.          17022200 C  0122:0002:1
                     TIMINGS[0,0] := XLUN; % NUMBER OF IDENTIFIERS SORTED.        17022300 C  0122:0003:2
                   END                                                %110-       17022400 C  0122:0005:2
                 ELSE                                                 %DFB        17023000 C  0122:0005:2
                   BEGIN                                              %DFB        17024000 C  0122:0005:2
                   IF BOOLEAN(A[8]) THEN                              %110-       17025000 C  0122:0005:3
                     XREFINFO[A[8]].FF := XLUN := XLUN + 1            %110-       17025100 C  0122:0006:0
                   ELSE                                               %110-       17025200 C  0122:0012:1
                     XREFINFO[A[8]].CF := XLUN := XLUN + 1;           %110-       17025300 C  0122:0014:0
                   A[8].IDNOF := XLUN;                                %110-       17025400 C  0122:0021:3
                   WRITE(DSK1,10,A[*]);                               %DFB        17026000 C  0122:0023:3
                   END;                                               %DFB        17027000 C  0122:0028:0
                 END;                                                 %DFB        17028000 C  0122:0028:0


            BOOLEAN STREAM PROCEDURE COMPS1(A,B);                     %DFB        17029000 C  0122:0028:0
PRT(760) = COMPS1
              BEGIN                                                   %DFB        17030000 C  0122:0028:0
              SI:=A;                                                  %DFB        17031000 C  0122:0029:2
              DI:=B;                                                  %DFB        17032000 C  0122:0029:2
              IF 63 SC < DC THEN                                      %110-       17033000 C  0122:0029:3
                TALLY := 1                                            %110-       17033100 C  0122:0030:0
              ELSE                                                    %110-       17033200 C  0122:0030:0
                BEGIN                                                 %110-       17033300 C  0122:0030:1
                  SI := A;                                            %110-       17033400 C  0122:0030:1
                  DI := B;                                            %110-       17033500 C  0122:0030:1
                  IF 63 SC = DC THEN                                  %110-       17033600 C  0122:0031:2
                    TALLY := 2;                                       %110-       17033700 C  0122:0031:3
                END;                                                  %110-       17033800 C  0122:0031:3
              COMPS1:=TALLY;                                          %DFB        17034000 C  0122:0031:3
              END;                                                    %DFB        17035000 C  0122:0032:0


            STREAM PROCEDURE HVS1(A);                                 %DFB        17036000 C  0122:0033:2
PRT(761) = HVS1
              BEGIN                                                   %DFB        17037000 C  0122:0033:2
              DI:=A;                                                  %DFB        17038000 C  0122:0033:2
```

```
                DS:=8 LIT "9";                                          %DFB        17039000 C  0122:0033:2
                SI:=A;                                                  %DFB        17040000 C  0122:0034:1
                DS:= 7 WDS;                                             %DFB        17041000 C  0122:0034:1
                DS := 8 LIT 3"777777777"; % ID.NO. AND SEG.NO. FIELDS   %110-       17041100 C  0122:0035:2
                END;                                                    %DFB        17042000 C  0122:0036:0

            BOOLEAN PROCEDURE COMP1(A,B);                               %DFB        17042100 C  0122:0036:1
PRT(762) = COMP1
                    ARRAY A,B[0];                                       %DFB        17042200 C  0122:0036:1
                    IF REAL(COMP1:=COMPS1(A,B)) = 2 THEN % IDS EQUAL    %110-       17042300 C  0122:0036:1
                      COMP1 := A[8].IDNOF < B[8].IDNOF;                 %110-       17042350 C  0122:0040:0

            PROCEDURE HV1(A);                                           %DFB        17042400 C  0122:0045:2
PRT(763) = HV1
                    ARRAY A[0];                                         %DFB        17042500 C  0122:0045:2
                    HVS1(A);                                            %DFB        17042600 C  0122:0045:2

            XLUN:=0;                                                    %DFB        17043000 C  0122:0047:2
                    REWIND(DSK1);                                       %DFB        17044000 C  0122:0048:1
                    SORT(OUTPUT1,INPUT1,0,HV1,COMP1,10,IF TOTALNO < 1000 THEN   %110-   17045000 C  0122:0050:1
PRT(764) = SORT TEMPORARY
PRT(765) = MERGE
                                7000 ELSE 10000);                       %110-       17045100 C  0122:0070:1
PRT(766) = SORT
                END;                                                    %DFB        17046000 C  0122:0072:1
PRT(767) = *SEGMENT DESCRIPTOR*
                                                                                 122 IS   79 LONG, NEXT SEG  119
                BEGIN                                                   %DFB        17047000 C  0119:0031:2
                 ARRAY IDTYPE[0:(IDMAX+4)x4-1];                         %112-       17047100 C  0119:0031:2
PRT(770) = *SEGMENT DESCRIPTOR*
                                                                                START OF SEGMENT **********  124
PRT(771) = IDTYPE
                 STREAM PROCEDURE SETUPHEADING(S,D,SEG,SEQNO,FWDTOG,LBLTOG,     %110-   17047200 C  0124:0005:2
PRT(772) = SETUPHEADING
                                                FWDSEQNO,TYPE,OWNTOG,PARAMTOG,  %110-   17047300 C  0124:0005:2
                                                VALTOG);                        %110-   17047350 C  0124:0005:2
                   VALUE SEG,SEQNO,FWDTOG,LBLTOG,FWDSEQNO,OWNTOG,PARAMTOG,      %110-   17047400 C  0124:0005:2
                     VALTOG;                                                    %110-   17047450 C  0124:0005:2
                  BEGIN                                                         %110-   17047500 C  0124:0005:2
                    SI := S;                                                    %110-   17047700 C  0124:0006:0
                    DI := D;                                                    %110-   17047800 C  0124:0006:0
                    63 (IF SC = " " THEN JUMP OUT ELSE DS := CHR);              %110-   17047900 C  0124:0006:1
                    DS := 6 LIT "  --  ";                                      %110-   17048000 C  0124:0009:2
                    OWNTOG (DS := 4 LIT "OWN ");                                %110-   17048100 C  0124:0010:0
                    SI := TYPE;                                                 %110-   17049300 C  0124:0012:0
                    32 (IF SC = "." THEN JUMP OUT ELSE DS := CHR);              %110-   17049400 C  0124:0012:0
                    PARAMTOG (DS := 6 LIT "  --  ";                           %110-   17049410 C  0124:0015:2
                              DS := 4 LIT "NAME";                               %110-   17049420 C  0124:0017:2
```

```
                        VALTOG (DI := DI - 4; DS := 5 LIT "VALUE");
                        DS := 10 LIT " PARAMETER");
                  DS := 18 LIT "  --  DECLARED AT ";
                  SI := LOC SEQNO;
                  DS := 8 DEC;
                  FWDTOG (DS := 17 LIT "  --  FORWARD AT ";
                          SI := LOC FWDSEQNO;
                          DS := 8 DEC);
                  LBLTOG (DS := 16 LIT "  --  OCCURS AT ";
                          SI := LOC FWDSEQNO;
                          DS := 8 DEC);
                END OF SETUPHEADING;



                STREAM PROCEDURE ADDASEQNO(SEQNO,N,STARS,D);
PRT(773) = ADDASEQNO
                  VALUE SEQNO,N,STARS;
                BEGIN
                  DI := D;
                  DI := DI + 8;
                  N (DI := DI + 10);
                  STARS(DI := DI - 1; DS := LIT "*");
                  SI := LOC SEQNO;
                  DS := 8 DEC;
                  DS := LIT " ";
                  STARS (DI := DI - 1; DS := LIT "*");
                END;



                STREAM PROCEDURE BLANKET(D);
PRT(774) = BLANKET
                BEGIN
                  DI := D;
                  DS := 8 LIT " ";
                  SI := D;
                  DS := 16 WDS;
                END OF BLANKET;



                PROCEDURE PRINTXREFSTATISTICS;
PRT(775) = PRINTXREFSTATISTICS
                BEGIN
                  SWITCH FORMAT STATS :=

PRT(776) = STATS
                    (///, "CROSS REFERENCE STATISTICS", /,
                          "------ --------- ----------", /),
                    ("PHASE ONE - SORT",I6," IDENTIFIERS"),
                    ("PHASE TWO - SORT",I7," REFERENCES"),
```

```
%110-        17049430 C  01241001713
%110-        17049440 C  01241002010
%110-        17049500 C  01241002210
%110-        17050400 C  01241002411
%110-        17050500 C  01241002411
%110-        17050600 C  01241002512
%110-        17050700 C  01241002811
%110-        17050800 C  01241002811
%110-        17050900 C  01241002912
%110-        17051000 C  01241003211
%110-        17051100 C  01241003211
%110-        17051200 C  01241003312

%110-        17051300 C  01241003313
%110-        17051400 C  01241003313
%110-        17051500 C  01241003313
%110-        17051600 C  01241003313
%110-        17051700 C  01241003410
%110-        17051800 C  01241003410
%110-        17051900 C  01241003411
%110-        17052000 C  01241003610
%110-        17052100 C  01241003810
%110-        17052200 C  01241003810
%110-        17052300 C  01241003811
%110-        17052400 C  01241003912
%110-        17052500 C  01241004112

%110-        17052600 C  01241004112
%110-        17052700 C  01241004112
%110-        17052800 C  01241004210
%110-        17052900 C  01241004210
%110-        17053000 C  01241004313
%110-        17053100 C  01241004313
%110-        17053200 C  01241004410

%110-        17053300 C  01241004410

%110-        17053400 C  01241004410
%110-        17053500 C  01241004410
     START OF SEGMENT **********  125
     START OF SEGMENT **********  126

%110-        17053600 C  01261000010
%110-        17053700 C  01261000010
%110-        17053800 C  01261000010
%110-        17053900 C  01261000010
```

```
                        ("PHASE THREE - PRINT CROSS REFERENCE (",I7," LINES)"),   %110-        17054000 C  0126:000:0
                        (X5,I4,"!",2I1," ELAPSED TIME (MIN:SEC)"),                %110-        17054100 C  0126:000:0
                        (X5,I4,"!",2I1," PROCESSOR TIME"),                        %110-        17054200 C  0126:000:0
                        (X5,I4,"!",2I1," I/O TIME",/);                            %110-        17054300 C  0126:000:0
                                                                                                126 IS   87 LONG, NEXT SEG  125
                      INTEGER I,J,K;                                              %110-        17054400 C  0125:000:0
STACK(F+2) = I
STACK(F+3) = J
STACK(F+4) = K
                      WRITE(LINE,STATS[0]);                                       %110-        17054500 C  0125:000:0
                      FOR I := 0 STEP 1 UNTIL 2 DO                                %110-        17054600 C  0125:000:0
                        BEGIN                                                     %110-        17054700 C  0125:000:2
                          WRITE(LINE,STATS[I+1],TIMINGS[I,0]);                    %110-        17054800 C  0125:000:2
PRT(777) = *LIST, LABEL, OR SEGMENT DESCRIPTOR*
                          FOR J := 1 STEP 1 UNTIL 3 DO                            %110-        17054900 C  0125:001:2
                            BEGIN                                                 %110-        17055000 C  0125:001:0
                              K := (TIMINGS[I,J] + 30) DIV 60; % ROUND TO NEAREST SECON        17055010 C  0125:001:0
                              WRITE(LINE,STATS[J+3],K DIV 60,(K:=K MOD 60) DIV 10,%110-        17055020 C  0125:001:1
PRT(1000) = *LIST, LABEL, OR SEGMENT DESCRIPTOR*
                                                                  K MOD 10);      %110-        17055025 C  0125:003:1
                            END;                                                  %110-        17055030 C  0125:003:1
                        END;                                                      %110-        17055100 C  0125:003:1
                    END PRINTXREFSTATISTICS;                                      %110-        17055200 C  0125:004:2
                                                                                                125 IS   44 LONG, NEXT SEG  124



                      DEFINE REFCOUNT = TIMINGS[1,0]#; % NUMBER OF REFERENCES SORTED.          17069300 C  0124:004:0
                      BOOLEAN FIRSTTIME; % TRUE ON FIRST CALL OF OUTPUT PROCEDURE.             17069400 C  0124:004:0
PRT(1001) = FIRSTTIME
                      ARRAY PAY[0:17];                                            %DFB         17069500 C  0124:004:0
PRT(1002) = PAY
                      REAL LASTADDRESS;                                           %110-        17069600 C  0124:004:0
PRT(1003) = LASTADDRESS
                      BOOLEAN PROCEDURE INPUT2(A);                                %DFB         17070000 C  0124:004:0
PRT(1004) = INPUT2
                      ARRAY A[0];                                                 %DFB         17071000 C  0124:004:0
                        BEGIN                                                     %DFB         17072000 C  0124:004:0
                        LABEL L,EOF;                                              %DFB         17073000 C  0124:004:0
                                                                                                START OF SEGMENT **********  127
                        DEFINE I = LASTADDRESS#;                                  %110-        17073100 C  0127:000:0
                        IF XREFPT:=XREFPT+1=30 THEN                               %DFB         17074000 C  0127:000:0
                          BEGIN                                                   %DFB         17075000 C  0127:000:3
                          READ(DSK2,30,XREFAY2[*])[EOF];                          %DFB         17076000 C  0127:000:0
PRT(1005) = EOF
                          XREFPT:=0;                                              %DFB         17077000 C  0127:000:3
                          END;                                                    %DFB         17078000 C  0127:000:0
                      IF ( I  :=XREFAY2[XREFPT]).[21:27] GTR 99999999 THEN GO TO EOF;         17079000 C  0127:000:0
                      A[0] := I & NEWID[I.REFIDNOF] REFIDNOF;                     %110-        17080000 C  0127:001:1
                      REFCOUNT := REFCOUNT + 1;                                   %110-        17080100 C  0127:002:0
                      GO TO L;                                                    %DFB         17081000 C  0127:003:3
                      EOF: INPUT2:=TRUE;                                          %DFB         17082000 C  0127:003:2
                           BLANKET(PAY);                                          %DFB         17083000 C  0127:003:3
                      XREFAY1[8] := XREFPT := LASTADDRESS := 0;                   %110-        17084000 C  0127:003:2
                      FILL IDTYPE[*] WITH                                         %110-        17084010 C  0127:003:3
                        "UNKNOWN,                              ", % 0             %110-        17084020 C  0127:003:0
                                                                                                START OF SEGMENT **********  128
```

```
                    "STREAM LABEL.                ", % 1                %110-       17084030 C  01271003912
                    "STREAM VARIABLE.             ", % 2                %110-       17084040 C  01271003912
                    "DEFINE.                      ", % 3                %110-       17084050 C  01271003912
                    "LIST.                        ", % 4                %110-       17084060 C  01271003912
                    "FORMAT.                      ", % 5                %110-       17084070 C  01271003912
                    "SWITCH FORMAT.               ", % 6                %110-       17084080 C  01271003912
                    "REAL SUBROUTINE.             ", % 7                %110-       17084090 C  01271003912
                    "SUBROUTINE.                  ", % 8                %110-       17084100 C  01271003912
                    "SWITCH LABEL.                ", % 9                %110-       17084110 C  01271003912
                    "PROCEDURE.                   ", % 10               %110-       17084120 C  01271003912
                    "INTRINSIC.                   ", % 11               %110-       17084130 C  01271003912
                    "STREAM PROCEDURE.            ", % 12               %110-       17084140 C  01271003912
                    "BOOLEAN STREAM PROCEDURE.    ", % 13               %110-       17084150 C  01271003912
                    "REAL STREAM PROCEDURE.       ", % 14               %110-       17084160 C  01271003912
                    "INTEGER STREAM PROCEDURE.    ", % 15               %110-       17084170 C  01271003912
                    "INTEGER STREAM PROCEDURE.    ", % 16               %110-       17084180 C  01271003912
                    "BOOLEAN PROCEDURE.           ", % 17               %110-       17084182 C  01271003912
                    "REAL PROCEDURE.              ", % 18               %110-       17084184 C  01271003912
                    "INTEGER PROCEDURE.           ", % 19               %110-       17084186 C  01271003912
                    "INTEGER PROCEDURE.           ", % 20               %110-       17084188 C  01271003912
                    "BOOLEAN.                     ", % 21               %110-       17084190 C  01271003912
                    "REAL.                        ", % 22               %110-       17084200 C  01271003912
                    "INTEGER.                     ", % 23               %110-       17084210 C  01271003912
                    "INTEGER.                     ", % 24               %110-       17084220 C  01271003912
                    "BOOLEAN ARRAY.               ", % 25               %110-       17084230 C  01271003912
                    "REAL ARRAY.                  ", % 26               %110-       17084240 C  01271003912
                    "INTEGER ARRAY.               ", % 27               %110-       17084250 C  01271003912
                    "INTEGER ARRAY.               ", % 28               %110-       17084260 C  01271003912
                    "                             ", % 29               %110-       17084270 C  01271003912
                    "NAME.                        ", % 30               %110-       17084280 C  01271003912
                    "INTEGER NAME.                ", % 31               %110-       17084290 C  01271003912
                    "LABEL.                       ", % 32               %112-       17084300 C  01271003912
                    "FIELD.                       "; % 33 (CLASS = 125) %112-       17084400 C  01271003912
                                                                                  128 IS  136 LONG. NEXT SEG  127
                  L:                                                       %DFB       17085000 C  01271003912
                  END;                                                     %DFB       17086000 C  01271004010
                                                                                  127 IS   45 LONG. NEXT SEG  124


                  PROCEDURE OUTPUT2(B,A);                                  %DFB       17087000 C  01241004810
PRT(1006) = OUTPUT2
                    VALUE B;                                               %DFB       17088000 C  01241004810
                    BOOLEAN B;                                             %DFB       17089000 C  01241004810
                    ARRAY A[0];                                            %DFB       17090000 C  01241004810
                    BEGIN      DEFINE PRINTER=LINE#;                       %DFB       17091000 C  01241004810
                                                                                  START OF SEGMENT **********  129
                  LABEL EOF2, SKIP;                                        %110-       17091100 C  01291000010
                  OWN BOOLEAN B2, FWDTOG, LBLTOG, WAITINGFORFWDREF;        %110-       17091110 C  01291000010
PRT(1007) = B2
PRT(1010) = FWDTOG
PRT(1011) = LBLTOG
PRT(1012) = WAITINGFORFWDREF
                  DEFINE MATCH(A,B) = REAL(BOOLEAN(A) EQV BOOLEAN(B)) =    %110-       17091115 C  01291000010
                                      REAL(NOT FALSE)#;                    %110-       17091116 C  01291000010
                  REAL I;                                                  %110-       17091120 C  01291000010
```

```
STACK(F+2) = I
                    DEFINE LINECOUNT = TIMINGS[2,0]#; % NUMBER OF LINES PRINTED.          17091140 C  0129:000:0
                    OWN REAL FWDSEQNO;                                          %110-     17091150 C  0129:000:0
PRT(1013) = FWDSEQNO
                    IF FIRSTTIME THEN % PRINT HEADINGS AND SAVE TIMINGS.        %110-     17091155 C  0129:000:0
                      BEGIN                                                     %110-     17091160 C  0129:000:1
                        FIRSTTIME := FALSE;                                     %110-     17091162 C  0129:000:2
                        TIME1 := TIME(1);                                       %110-     17091165 C  0129:000:0
                        DATIME;                                                 %110-     17091170 C  0129:000:2
                        UPDATETIMES(1);                                         %110-     17091175 C  0129:000:3
                        SAVETIMES(2); % SAVE TIMES FOR START OF XREF PRINT.     %110-     17091180 C  0129:000:1
                      END;                                                      %110-     17091200 C  0129:000:2
                    IF NOT B2 THEN                                              %110-     17091210 C  0129:000:2
                      IF B THEN % END OF SORT - LIST OUT REST OF SEQ. NO.        %110-     17091300 C  0129:000:0
                        IF XREFPT ≠ 0 THEN % WE GOT SOME TO LIST OUT            %110-     17091400 C  0129:000:1
                          BEGIN                                                 %110-     17091500 C  0129:000:0
                            WRITE(LINE[DBL],15,PAY[*]);                         %110-     17091510 C  0129:000:1
                            LINECOUNT := LINECOUNT + 1;                         %110-     17091520 C  0129:001:2
                          END                                                   %110-     17091530 C  0129:001:1
                        ELSE % NOTHING TO LIST OUT                              %110-     17091600 C  0129:001:1
                      ELSE % NOT END OF SORT                                    %110-     17091700 C  0129:001:1
                        IF NOT MATCH(LASTADDRESS,A[0]) AND A[0].REFIDNOF ≠ 0 AND          17091800 C  0129:001:2
                           A[0].REFIDNOF ≥ XREFAY1[8].IDNOF THEN                %110-     17091900 C  0129:002:2
                          IF A[0].TYPEREF = FORWARDREF THEN %                   %110-     17092000 C  0129:002:3
                            WAITINGFORFWDREF := TRUE                            %110-     17092100 C  0129:002:3
                          ELSE                                                  %110-     17092200 C  0129:002:1
                            IF A[0].TYPEREF = LBLREF THEN %                     %110-     17092300 C  0129:002:2
                              BEGIN                                             %110-     17092400 C  0129:002:2
                                LBLTOG := TRUE;                                 %110-     17092500 C  0129:002:3
                                FWDSEQNO := A[0].SEQNOF;                        %110-     17092600 C  0129:003:1
                              END                                               %110-     17092700 C  0129:003:2
                            ELSE                                                %110-     17092800 C  0129:003:2
                              IF A[0].TYPEREF = DECLREF THEN                    %110-     17092900 C  0129:003:2
                                IF WAITINGFORFWDREF THEN % THIS MUST BE IT      %110-     17093000 C  0129:003:4
                                  BEGIN                                         %110-     17093100 C  0129:003:5
                                    WAITINGFORFWDREF := FALSE;                  %110-     17093200 C  0129:003:6
                                    FWDTOG := TRUE;                             %110-     17093300 C  0129:003:7
                                    FWDSEQNO := A[0].SEQNOF;                    %110-     17093400 C  0129:003:8
                                  END                                           %110-     17093500 C  0129:003:9
                                ELSE % ITS A NORMAL DECLARATION - NOT FORWARD   %110-     17093600 C  0129:003:9
                                  BEGIN                                         %110-     17093700 C  0129:003:9
                                    IF A[0].REFIDNOF > XREFAY1[8].IDNOF THEN    %110-     17093850 C  0129:004:0
                                      DO                                        %110-     17093900 C  0129:004:2
                                        READ(DSK1,10,XREFAY1[*]) [EOF2]         %110-     17093950 C  0129:004:3
PRT(1014) = EOF2
                                      UNTIL                                     %110-     17094000 C  0129:004:7
                                        A[0].REFIDNOF ≤ XREFAY1[8].IDNOF;       %110-     17094050 C  0129:004:8
                                    IF A[0]. REFIDNOF < XREFAY1[8].IDNOF THEN   %110-     17094100 C  0129:005:2
                                      GO TO SKIP;                               %110-     17094150 C  0129:005:3
                                    IF XREFPT > 0 THEN % THERE IS STUFF TO PRINT          17094200 C  0129:005:3
                                     BEGIN                                      %110-     17094240 C  0129:005:4
                                      IF SINGLTOG THEN                          %110-     17094250 C  0129:005:5
                                        WRITE(LINE,15,PAY[*])                   %110-     17094300 C  0129:005:5
                                      ELSE                                      %110-     17094350 C  0129:005:9
                                        WRITE(LINE[DBL],15,PAY[*]);             %110-     17094400 C  0129:006:1
                                      LINECOUNT := LINECOUNT + 1;               %110-     17094410 C  0129:006:5
                                     END                                        %110-     17094420 C  0129:006:9
```

```
      ELSE                                                    %110-      17094450 C  0129:006:2
        IF NOT SINGLTOG THEN                                  %110-      17094500 C  0129:006:2
          WRITE(LINE);                                        %110-      17094550 C  0129:007:1
      XREFPT := 0;                                            %110-      17094600 C  0129:007:2
      BLANKET(PAY[*]);                                        %110-      17094650 C  0129:007:0
       SETUPHEADING(XREFAY1[*],PAY[*],XREFAY1[8],                        17094700 C  0129:007:2
         SEGNOF,A[0],SEQNOF,FWDTOG,LBLTOG,                    %110-      17094800 C  0129:007:2
        FWDSEQNO,IDTYPE((IF (I :=                             %110-      17094900 C  0129:008:3
        XREFAY1[9].CLASS) > IDMAX THEN IF I =                 %112-      17095000 C  0129:008:0
        FIELDID THEN 33 ELSE 0 ELSE I) * 4],                  %112-      17095100 C  0129:008:2
        REAL(I ≥ BOOID AND XREFAY1[9],[9:2] = 1),                        17095300 C  0129:008:1
        REAL((I ≥ BOOID OR I = LOCLID) AND BOOLEAN                       17095310 C  0129:009:2
        (XREFAY1[9],[9:1])), XREFAY1[9],[10:1]);                         17095320 C  0129:009:1
      FWDTOG := LBLTOG := FALSE;                              %110-      17095400 C  0129:009:2
      WRITE(LINE,15,PAY[*]);                                  %110-      17095500 C  0129:009:2
      LINECOUNT := LINECOUNT + 1;                             %110-      17095510 C  0129:010:3
      BLANKET(PAY[*]);                                        %110-      17095550 C  0129:010:2
    END                                                       %110-      17095600 C  0129:010:0
  ELSE % IT MUST BE A NORMAL REFERENCE                        %110-      17095700 C  0129:010:0
    IF A[0].SEQNOF ≠ LASTADDRESS.SEQNOF THEN                  %110-      17095750 C  0129:010:0
      BEGIN                                                   %110-      17095800 C  0129:010:2
        ADDASEQNO(A[0].SEQNOF,XREFPT,A[0],[5:1],                         17095900 C  0129:010:3
                  PAY[*]);                                    %110-      17096000 C  0129:011:0
        IF (XREFPT := XREFPT + 1) = 11 THEN %FULL                        17096100 C  0129:011:2
          BEGIN                                               %110-      17096200 C  0129:011:1
            WRITE(LINE,15,PAY[*]);                            %110-      17096300 C  0129:011:2
            LINECOUNT := LINECOUNT + 1;                       %110-      17096350 C  0129:011:3
            XREFPT := 0;                                      %110-      17096400 C  0129:012:2
            BLANKET(PAY[*]);                                  %110-      17096450 C  0129:012:0
          END                                                 %110-      17096500 C  0129:012:2
        END                                                   %110-      17096550 C  0129:012:2
      ELSE % REFERENCE TO SAME SEQ. NO. SKIP IT               %110-      17096575 C  0129:012:2
  FLSE % THIS IS A REFERENCE TO THE SAME SEQ. NO. - SKIP                 17096600 C  0129:012:2
ELSE % HIT END OF IDENTIFIER FILE - JUST SKIP OVER REFERENCES             17096700 C  0129:012:3
  EOF2: B2 := TRUE; %  SO SORT CAN GO TO NORMAL EOJ           %110-      17096800 C  0129:012:0
IF NOT B THEN SKIP; LASTADDRESS := A[0];                      %110-      17096850 C  0129:012:8:0
END OF OUTPUT2;                                               %110-      17096900 C  0129:013:0:0
                                                                  129 IS  135 LONG. NEXT SEG  124
```

```
                PROCEDURE HV2(A);                                 %DFB      17112000 C  0124:004:8:0
PRT(1015) = HV2
                  ARRAY A[0];                                     %DFB      17113000 C  0124:004:8:0
                  A[0] := 3"777777777777777"; % BIGGEST FLOATING PT. NO. %110-  17114000 C  0124:004:8:0


                BOOLEAN PROCEDURE COMP2(A,B);                     %DFB      17115000 C  0124:005:2:0
PRT(1016) = COMP2
                  ARRAY A,B[0];                                   %DFB      17116000 C  0124:005:2:0
                    COMP2 := IF A[0].REFIDNOF < B[0].REFIDNOF THEN % DIF IDS  17117000 C  0124:005:2:0
                               TRUE                               %110-     17117100 C  0124:005:4:0
                             ELSE                                 %110-     17117200 C  0124:005:5:2
                               IF A[0].RFFIDNOF = B[0].REFIDNOF THEN %110-   17117300 C  0124:005:5:3
```

```
                    IF A[0].[1:4] LSS B[0].[1:4] THEN                    %110-          17117400 C  0124:0057:3
                      TRUE                                               %110-          17117500 C  0124:0060:1
                    ELSE                                                 %110-          17117600 C  0124:0061:2
                      IF A[0].[1:4] = B[0].[1:4] THEN                    %110-          17117700 C  0124:0061:3
                        IF A[0].SEQNOF < B[0].SEQNOF THEN                %110-          17117702 C  0124:0064:0
                          TRUE                                           %110-          17117704 C  0124:0066:1
                        ELSE                                             %110-          17117706 C  0124:0067:3
                          IF A[0].SEQNOF = B[0].SEQNOF THEN              %110-          17117708 C  0124:0068:0
                            BOOLEAN(A[0].[5:1])                          %110-          17117710 C  0124:0070:0
                          ELSE                                           %110-          17117712 C  0124:0071:3
                            FALSE                                        %110-          17117714 C  0124:0072:0
                        ELSE                                             %110-          17117720 C  0124:0072:1
                          FALSE                                          %110-          17117730 C  0124:0073:2
                    ELSE                                                 %110-          17117800 C  0124:0073:2
                      FALSE;                                             %110-          17117900 C  0124:0073:3


                  SAVETIMES(1); % SAVE TIMES FOR START OF REFERENCES SORT %110-          17117910 C  0124:0076:0
                  FIRSTTIME := TRUE; % LET OUTPUT PROCEDURE KNOW ABOUT FIRST CAL         17117920 C  0124:0077:3
                  XREFPT:=29;       REWIND(DSK2);                        %DFB           17118000 C  0124:0078:1
                  SORT(OUTPUT2,INPUT2,0,HV2,COMP2,1,6000);               %110-          17119000 C  0124:0081:2
                UPDATETIMES(2); % UPDATE TIMES FOR PRINTING CROSS REFERENCE%110-          17119100 C  0124:0102:1
                PRINTXREFSTATISTICS;                                     %110-          17119200 C  0124:0103:2
              END;                                                       %DFB           17120000 C  0124:0103:3
PRT(1017) = *SEGMENT DESCRIPTOR*

                                                                                  124 IS  108 LONG, NEXT SEG  119
              END;                                                       %DFB           17121000 C  0119:0032:0
PRT(1020) = *SEGMENT DESCRIPTOR*

                                                                                  119 IS   36 LONG, NEXT SEG    3
                            END MAIN BLOCK;                              %108-          17121500 C  0003:0647:2
PRT(1021) = *SEGMENT DESCRIPTOR*

                                                                                    3 IS  651 LONG, NEXT SEG    2
              END.                                                       %DFB           17122000 C  0002:0038:0
                                                                                    2 IS   42 LONG, NEXT SEG    1
PRT(546) = EXP INTRINSIC, SEGMENT NUMBER =  130.
PRT(545) = LN INTRINSIC, SEGMENT NUMBER =  131.
PRT(353) = OUTPUT(W) INTRINSIC, SEGMENT NUMBER =  132.
PRT(5) = BLOCK CONTROL INTRINSIC, SEGMENT NUMBER =  133.
PRT(766) = SORT INTRINSIC, SEGMENT NUMBER =  134.
PRT(547) = X TO THE I INTRINSIC, SEGMENT NUMBER =  135.
PRT(412) = GO TO SOLVER  INTRINSIC, SEGMENT NUMBER =  136.
PRT(14) = ALGOL WRITE   INTRINSIC, SEGMENT NUMBER =  137.
PRT(15) = ALGOL READ    INTRINSIC, SEGMENT NUMBER =  138.
PRT(16) = ALGOL SELECT  INTRINSIC, SEGMENT NUMBER =  139.
PRT(765) = MERGE INTRINSIC, SEGMENT NUMBER =  140.
PRT(430) = DYNAMIC DIALS INTRINSIC, SEGMENT NUMBER =  141.
PRT(210) = FILE ATTRBUTS INTRINSIC, SEGMENT NUMBER =  142.

                                                                                    1 IS    2 LONG, NEXT SEG    0
                                                                                  143 IS   69 LONG, NEXT SEG    0
NUMBER OF ERRORS DETECTED =   0.  COMPILATION TIME =  362 SECONDS.

PRT SIZE = 530; TOTAL SEGMENT SIZE =  8408 WORDS; DISK SIZE = 540 SEGS; NO. PGM. SEGS = 143

ESTIMATED CORE STORAGE REQUIRED = 26836 WORDS.

ESTIMATED AUXILIARY MEMORY REQUIRED =     0 WORDS.
```

NUMBER OF CARD-IMAGES PROCESSED = 7848.

ESPOL /DISK
==========

SOURCE FILE: SYMBOL /ESPOL

```
A -- STREAM VARIABLE -- NAME PARAMETER -- DECLARED IN SEGMENT 3 AT 02022000
     02023000
A -- STREAM VARIABLE -- NAME PARAMETER -- DECLARED IN SEGMENT 3 AT 02054000
     02055000
A -- STREAM VARIABLE -- NAME PARAMETER -- DECLARED IN SEGMENT 3 AT 02062500
     02063500
A -- REAL ARRAY -- DECLARED IN SEGMENT 15 AT 02264100
     02265100  02265150  02265250  02265300  02265600  02265650  02265700  02265750  02265850  02265900  02265950
     02266000
A -- INTEGER -- VALUE PARAMETER -- DECLARED IN SEGMENT 3 AT 03047100
     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 *04315000*
A -- REAL -- VALUE PARAMETER -- DECLARED IN SEGMENT 3 AT 04500000
     04529000  04534000  04542000  04548000  04550000  04551000  04556000
A -- STREAM VARIABLE -- VALUE PARAMETER -- DECLARED IN SEGMENT 38 AT 05065000
     05066000
A -- REAL -- DECLARED IN SEGMENT 39 AT 05130000
     *05131000* 05132000
A -- STREAM VARIABLE -- VALUE PARAMETER -- DECLARED IN SEGMENT 44 AT 05334100
A -- REAL ARRAY -- DECLARED IN SEGMENT 46 AT 05413000
     05426000  05427000  05431000  05432000
A -- STREAM VARIABLE -- NAME PARAMETER -- DECLARED IN SEGMENT 46 AT 05415000
     05417000 *05419000* 05421000  05422000
A -- DEFINE -- DECLARED IN SEGMENT 69 AT 08015000
     08181000  08188000
A -- REAL -- VALUE PARAMETER -- DECLARED IN SEGMENT 69 AT 08020000
     08021000
A -- REAL -- NAME PARAMETER -- DECLARED IN SEGMENT 69 AT 08027000
     *08035000**08037000*
A -- REAL -- DECLARED IN SEGMENT 93 AT 13359000
     *13364000* 13365000  13366000
A -- STREAM VARIABLE -- VALUE PARAMETER -- DECLARED IN SEGMENT 96 AT 13673100
A -- REAL ARRAY -- NAME PARAMETER -- DECLARED IN SEGMENT 122 AT 17007000
     17006000  17010000
A -- REAL ARRAY -- NAME PARAMETER -- DECLARED IN SEGMENT 122 AT 17019000
     17016000  17025000  17025100  17025300 *17025400* 17026000
A -- STREAM VARIABLE -- NAME PARAMETER -- DECLARED IN SEGMENT 122 AT 17029000
     17031000  17033400
A -- STREAM VARIABLE -- NAME PARAMETER -- DECLARED IN SEGMENT 122 AT 17036000
     17038000  17040000
A -- REAL ARRAY -- NAME PARAMETER -- DECLARED IN SEGMENT 122 AT 17042200
     17042100  17042300  17042350
A -- REAL ARRAY -- NAME PARAMETER -- DECLARED IN SEGMENT 122 AT 17042500
```

```
        17042400  17042600
A  --  REAL ARRAY  --  NAME PARAMETER  --  DECLARED IN SEGMENT 124 AT 17071000
        17070000 *17080000*
A  --  REAL ARRAY  --  NAME PARAMETER  --  DECLARED IN SEGMENT 124 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 124 AT 17113000
        17112000 *17114000*
A  --  REAL ARRAY  --  NAME PARAMETER  --  DECLARED IN SEGMENT 124 AT 17116000
        17115000  17117000  17117300  17117400  17117700  17117702  17117708  17117710
ABS  --  REAL STREAM PROCEDURE  --  DECLARED IN SEGMENT 38 AT 05065000
        05066000  05092000
ACC  --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 67 AT 07653500
        07654500
ACCUM  --  ALPHA ARRAY  --  DECLARED IN SEGMENT 3 AT 01304000
        02128500  02251000  02255000  02259000 *02327000* 02366100 *02371000**02499000* 02502000  02581000 *02590000*
        02592000  02602600 *02647000* 02650000  02652000  02686000  02686500  02696000  02698000 *02700000* 02705000
        02754000  02822000  02846000  02865000  02881000 *02958500**05043000* 05044000  05325460 *07086200* 07086300
        07668500  07670000  07671000  10264700  10264900  10269000  10274000  12117000 *12118000**12119000**12120000*
        12122000  13281000  13282000 *13307000**13308000**13309000* 13310000 *13372000**13755000* 14259000  14259080
        14269280  14440000  14441000  16000700  16211000  16387000  16392000  16434000
ACCUM  --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 3 AT 02041000
        02043000
ACCUM  --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 9 AT 02089500
        02122500
ACCUM  --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 37 AT 05016000
        05028000
ACCUM  --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 57 AT 07038100
        07038500
ACCUM  --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 87 AT 10236000
        10239000
ACLASS  --  REAL  --  DECLARED IN SEGMENT 57 AT 07038000
       *07041000* 07047000  07057000  07069000 *07081000* 07089000
ACTUALPARAPART  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 07035000
        07410000  07438000
ADD  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01633000
        05226000  07525000  07646615  08126000  08214000
ADD  --  REAL  --  DECLARED IN SEGMENT 91 AT 13221000
       *13248000* 13250000  13252000  13252500 *13259000*
ADDASEQNO  --  STREAM PROCEDURE  --  DECLARED IN SEGMENT 124 AT 17051400
        17095900
ADDC  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 13212000
ADDITIONAL  --  DEFINE  --  DECLARED IN SEGMENT 65 AT 07595000
        07601000  07603000  07610000
ADDOP  --  DEFINE  --  DECLARED IN SEGMENT 108 AT 16005000
        16326000
ADDR  --  REAL  --  DECLARED IN SEGMENT 110 AT 16068000
ADDR  --  REAL  --  DECLARED IN SEGMENT 114 AT 16195000
       *16196000**16201000**16204000* 16208000
ADDR  --  INTEGER  --  DECLARED IN SEGMENT 116 AT 16314000
       *16321000**16344000* 16352000
ADDR  --  INTEGER  --  DECLARED IN SEGMENT 117 AT 16364000
       *16371000**16376000**16378000* 16378500  16379000  16383000  16385000  16389000
ADDR  --  REAL  --  DECLARED IN SEGMENT 118 AT 16405000
       *16408000**16412000**16414000* 16415000  16417000
ADDRES  --  OWN REAL  --  DECLARED IN SEGMENT 69 AT 08010000
       *08055000* 08067000  08068000  08075000  08179000  08212000
```

```
ADDRESS  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01152000
        02850000  02896000  02900000  02923000  04508000  05131000  05217000  05233000  05280000  05281000  05344570
        05344615  05344620  05344660  05344700  05344710  06016000  06072000  06108000  06185000  06320100  06320200
        06324000  06325000  06326000  07071000  07090000  07439000  07745000  07768100  07994000  07995000  08021000
        08029000  08044000  08055000  10277000  12015000  12024000  12036000  12047000  12108000  13205000  13248000
        13259000  13394000  14085000  14115000  14129000  14269520  14269540  14349000  14399000  14430000  14437000
        14461200  14483200  15086400  15086800  15092600  15095600  15100600  15101600  15185300  15185500  15186800
        15267000  15276010  15276060  15276080  15290000  15301200  15304000  15305500  15306200  15306400  16121000
        16124000  16200000  16204000  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  --  INTEGER  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 05199000
        05201000
ADDRESS  --  REAL  --  DECLARED IN SEGMENT 58 AT 07393000
       *07397000* 07397400  07397500  07413000
ADDRSS  --  REAL  --  DECLARED IN SEGMENT 3 AT 01604000
        07397000  07677000 *13205000**13344000**13346000**13347000* 13348000  13373000 *13767000* 13768000  13770000
        14313000  14320000 *16000500*
ADDVALUE  --  INTEGER  --  DECLARED IN SEGMENT 3 AT 01000860
        02193000  02234800 *02592000**09028920*
ADES  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01299010
ADJUST  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 04083000
        04087000  04144000  04172000  06208500  07080000  07459000  07579000  07585100  07586000  07600200  07607100
        07727040  08190000  13605000  13611000  13630000
ADJUST  --  PROCEDURE  --  DECLARED IN SEGMENT 108 AT 16023100
        16137000  16158000
ADOP  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01278000
        05344550  05344670  06013000  07666000  08029000  08031000  12013000  12022000  16326000
ADR  --  INTEGER  --  DECLARED IN SEGMENT 66 AT 07646370
       *07646470* 07646605
ADR  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 95 AT 13633000
        13642000
ADRES  --  INTEGER  --  DECLARED IN SEGMENT 40 AT 05215000
       *05217000**05219000* 05226000  05227000
AEXP  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 06002000  --  FORWARD AT 03001000
        06007000  06057000  06105000  06174000  06202000  06305000  06410000  06416000  06422000  07045000  07062000
        07086000  07487000  07493000  07646455  08101000  08123000  08131000  08135000  08142000  12046000  12051000
        12054000  13759000  15097400  15186600  15276070  15277000  15303700
AGAIN  --  LABEL  --  DECLARED IN SEGMENT 19 AT 02360000  --  OCCURS AT 02372000
        02377000  02390000  02393000  02395000  02397000  02404000  02418000  02419120  02419200  02421100  02431000
        02460000  02470000  02472000  02474000  02480000  02493000  02496000  02501000  02506000  02508000  02513000
        02516000  02520000  02522000  02564000  02567000  02601000
AGAIN  --  LABEL  --  DECLARED IN SEGMENT 47 AT 05439000  --  OCCURS AT 05445000
        05454000  05457000
AGAIN  --  LABEL  --  DECLARED IN SEGMENT 68 AT 07726000  --  OCCURS AT 07727000
        07727100  07741000
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 *14055000**14606000*
AJUMPO  --  BOOLEAN  --  DECLARED IN SEGMENT 98 AT 14034000
       *14054000* 14606000
AKKUM  --  REAL  --  DECLARED IN SEGMENT 3 AT 01693000
```

```
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
ALL -- REAL -- VALUE PARAMETER -- DECLARED IN SEGMENT 69 AT 08052000
      08054000  08055000  08056000  08057000
ALLTHU -- LABEL -- DECLARED IN SEGMENT 31 AT 04168000 -- OCCURS AT 04186000
      04176000
ALONG -- LABEL -- DECLARED IN SEGMENT 20 AT 02325000 -- OCCURS AT 02358000
      02336000  02340000  02354000
ALPHADEC -- LABEL -- DECLARED IN SEGMENT 98 AT 14013000 -- OCCURS AT 14139000
      14018000
ALPHASIZE -- DEFINE -- DECLARED IN SEGMENT 3 AT 01707000
ALPHAWORDS -- STREAM PROCEDURE -- DECLARED IN SEGMENT 15 AT 02264700
      02264900  02265600  02265850
AMPERSAND -- DEFINE -- DECLARED IN SEGMENT 3 AT 01266500
      06034000  06180000  07084000  15086200  15091800  15184850  15302100  15305100
ANDOP -- DEFINE -- DECLARED IN SEGMENT 3 AT 01276000
      02964500  02971000  02983500
ANOTHER -- LABEL -- DECLARED IN SEGMENT 56 AT 07007000 -- OCCURS AT 07009000
      07012000  07015000  07024000
ANOTHER -- LABEL -- DECLARED IN SEGMENT 57 AT 07037000 -- OCCURS AT 07041000
      07095000
ARGH -- LABEL -- DECLARED IN SEGMENT 24 AT 02638000 -- OCCURS AT 02649000
      02685000  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
ARRAE -- PROCEDURE -- DECLARED IN SEGMENT 3 AT 13376000
      13410000  14156000
ARRAYDEC -- LABEL -- DECLARED IN SEGMENT 98 AT 14015000 -- OCCURS AT 14156000
      14020000
ARRAYMONFILE -- DEFINE -- DECLARED IN SEGMENT 3 AT 01526000
ARRAYV -- DEFINE -- DECLARED IN SEGMENT 3 AT 01294000
      14161000
ASKIP -- STREAM VARIABLE -- VALUE PARAMETER -- DECLARED IN SEGMENT 87 AT 10236000
      10237000  10239000
ASSIGNOP -- DEFINE -- DECLARED IN SEGMENT 3 AT 01273000
      06199000  07397210  07401000  07662000  08178000  08229000  12028000  13758000  15085600  15088600  15183800
      15276050  15276160  15300500  16316000  16367000
ASSIGNREF -- DEFINE -- DECLARED IN SEGMENT 3 AT 01007500
      07660600  08178100  15090600  15091200  15184400  15184700  15301000  16318500
ASTOG -- BOOLEAN -- DECLARED IN SEGMENT 3 AT 01300100
      05348000  07600410 *07600430*
ASTRISK -- DEFINE -- DECLARED IN SEGMENT 3 AT 01212200
      07600400  13400000  13404000  15091400  15184750  15253000  15301900
ATSIGN -- LABEL -- DECLARED IN SEGMENT 24 AT 02637000 -- OCCURS AT 02683000
      02641000
ATYPE -- DEFINE -- DECLARED IN SEGMENT 3 AT 01452000
      06005000  06057000
AUXMEMERR -- LABEL -- DECLARED IN SEGMENT 98 AT 14016000 -- OCCURS AT 14136100
      14021000
AUXMEMV -- DEFINE -- DECLARED IN SEGMENT 3 AT 01298500
AWAY -- LABEL -- DECLARED IN SEGMENT 10 AT 02191000 -- OCCURS AT 02196250
```

```
          02191250  02194250  02195250
AXNUM  --  REAL  --  DECLARED IN SEGMENT 3 AT 07034000
          *07086700* 09384000  09384100
B   --  REAL  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 01717000
B   --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 3 AT 02054000
          02055000
B   --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 3 AT 02062500
          02063500
B   --  BOOLEAN  --  DECLARED IN SEGMENT 20 AT 02323000
          *02343000* 02352000  02353000 *02356000* 02357000
B   --  BOOLEAN  --  NAME PARAMETER  --  DECLARED IN SEGMENT 3 AT 02955500
B   --  BOOLEAN  --  DECLARED IN SEGMENT 26 AT 02970000
          *02970500* 02971000  02971500
B   --  BOOLEAN  --  DECLARED IN SEGMENT 27 AT 02973500
          *02976000**02978000**02978500* 02979000
B   --  BOOLEAN  --  NAME PARAMETER  --  DECLARED IN SEGMENT 3 AT 02980000
          *02983500* 02984000  02984500
B   --  BOOLEAN  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 03041000
B   --  BOOLEAN  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 03047100
          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 *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 36 AT 04505000
          04503000  04507000
B   --  OWN REAL  --  DECLARED IN SEGMENT 69 AT 08010000
          08044000  08047000  08089000  08112000  08116000 *08122000**08169000* 08183000  08199000 *08209000* 08213000
B   --  BOOLEAN  --  NAME PARAMETER  --  DECLARED IN SEGMENT 69 AT 08027000
          *08035000**08037000*
B   --  REAL  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 81 AT 09393240
          09393300  09393310  09393330  09393350 *09393370* 09393400 *09393410* 09393430 *09393440*
B   --  BOOLEAN  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 122 AT 17018000
          17016000  17017000  17021000
B   --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 122 AT 17029000
          17032000  17033500
B   --  REAL ARRAY  --  NAME PARAMETER  --  DECLARED IN SEGMENT 122 AT 17042200
          17042100  17042300  17042350
B   --  BOOLEAN  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 124 AT 17089000
          17087000  17088000  17091115  17091300  17096850
B   --  REAL ARRAY  --  NAME PARAMETER  --  DECLARED IN SEGMENT 124 AT 17116000
          17115000  17117000  17117300  17117400  17117700  17117702  17117708
BACK  --  INTEGER  --  DECLARED IN SEGMENT 61 AT 07491000
          *07493000* 07495000
BACK  --  REAL  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 70 AT 08084000
          08082000  08083000  08089000
BACK  --  LABEL  --  DECLARED IN SEGMENT 86 AT 10259000  --  OCCURS AT 10263300
          10276500  10278000  10279000
BACKFIX  --  INTEGER  --  DECLARED IN SEGMENT 70 AT 08091000
          *08097000**08099000**08106000* 08127000 *08146000**08147000* 08157000  08168000  08171000
BAF  --  INTEGER PROCEDURE  --  DECLARED IN SEGMENT 3 AT 07458000  --  FORWARD AT 03046000
          *07459000*
BANA  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 06419000  --  FORWARD AT 03048000
          05225000  06423000  07523000
```

```
BASE -- STREAM VARIABLE -- VALUE PARAMETER -- DECLARED IN SEGMENT 80 AT 09370000
        09371000  09374000
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
BBW -- DEFINE -- DECLARED IN SEGMENT 3 AT 01635000
        06207000  07444000  07495000  07646635  08047000  08078000  08166000  08168000
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  07646456  09252000  13771000  14393000  14479000
BEND -- DEFINE -- DECLARED IN SEGMENT 3 AT 01001462
        02910600
BENDBIT -- DEFINE -- DECLARED IN SEGMENT 3 AT 01001171
        02419200  02910600
BETA1 -- LABEL -- DECLARED IN SEGMENT 94 AT 13379000 -- OCCURS AT 13390000
        13409000
BEXP -- DEFINE -- DECLARED IN SEGMENT 3 AT 03006000
        06410000  07487000  07493000  08135000  08142000
BFC -- DEFINE -- DECLARED IN SEGMENT 3 AT 01636000
        06301000  07496000  07572000  07573000  07575000  07579000  07583000  07585100  08151000
BFW -- DEFINE -- DECLARED IN SEGMENT 3 AT 01637000
        04218000  06307000  07088000  07512000  07516400  07518000  07526000  07529000  07572000  07586000  07646465
        07646555  07646605  07646620  07646665  08125000  08127000  08146000  08148000  08215000  08228000  13263000
        13274000  13606000  14041000  14451000
BIT -- INTEGER -- VALUE PARAMETER -- DECLARED IN SEGMENT 3 AT 02307000
        02311000
BIT -- DEFINE -- DECLARED IN SEGMENT 108 AT 16015000
        16216000
BITEDUST -- STREAM PROCEDURE -- DECLARED IN SEGMENT 38 AT 05067000
        05087000  05088000  05089000  05090000
BITOP -- DEFINE -- DECLARED IN SEGMENT 3 AT 01269000
        06087000
BLANKET -- STREAM PROCEDURE -- DECLARED IN SEGMENT 3 AT 01737350
        01737500  02265100  02265250  02265600  02265700  02265850  02265950  05325450  09034500
BLANKET -- STREAM PROCEDURE -- DECLARED IN SEGMENT 124 AT 17052600
        17053200  17083000  17094650  17095550  17096450
BLKAD -- INTEGER -- DECLARED IN SEGMENT 98 AT 14026000
       *14037000**14044000* 14600000
BLOCK -- PROCEDURE -- DECLARED IN SEGMENT 3 AT 14001000 -- FORWARD AT 03067000
        07750000  07768200  14482000  14613000  17121500
BLOCKCTR -- DEFINE -- DECLARED IN SEGMENT 3 AT 01430000
BNS -- DEFINE -- DECLARED IN SEGMENT 108 AT 16011000
        16122000  16124000
BNSFIX -- REAL -- DECLARED IN SEGMENT 112 AT 16118000
       *16121000* 16141000
BOOARRAYID -- DEFINE -- DECLARED IN SEGMENT 3 AT 01202000
BOOID -- DEFINE -- DECLARED IN SEGMENT 3 AT 01198000
        02967000  02977500  07047000  08057000  08058000  12042000  14140000  17095300  17095310
BOOLCOMP -- PROCEDURE -- DECLARED IN SEGMENT 3 AT 02980000 -- FORWARD AT 02955500
```

```
C -- BOOLEAN -- VALUE PARAMETER -- DECLARED IN SEGMENT 69 AT 08020000
        08021000
C -- STREAM VARIABLE -- VALUE PARAMETER -- DECLARED IN SEGMENT 96 AT 13673100
        13673350
CALL -- DEFINE -- DECLARED IN SEGMENT 3 AT 01721000
        05402100  05405000  05407000  05409000  05428000  05430000  07397400  07397500  14483400
CALL -- PROCEDURE -- DECLARED IN SEGMENT 69 AT 08072000
        08079000  08098000  08108000  08130000  08217000
CALLA -- REAL ARRAY -- DECLARED IN SEGMENT 3 AT 01720000
        05402100  05405000  05407000  05409000  05428000  05430000  07397400 *07397500**14483400*
CALLINFO -- REAL -- DECLARED IN SEGMENT 3 AT 01722000
        07397300 *14481100* 14483300  14483400
CALLSTATEMENT -- LABEL -- DECLARED IN SEGMENT 98 AT 14017000 -- OCCURS AT 14505000
        14132000  14137300
CALLSWITCH -- PROCEDURE -- DECLARED IN SEGMENT 3 AT 05315000
        05316000  13263000
CALLX -- REAL -- DECLARED IN SEGMENT 3 AT 01722000
        07397300 *07397500**14481100* 14483500
CARD -- FILE -- DECLARED IN SEGMENT 3 AT 01557000
        02213750  02215500  02216250  09281500  17002490
CARD -- STREAM VARIABLE -- VALUE PARAMETER -- DECLARED IN SEGMENT 3 AT 02188000
        02188500
CARDCOUNT -- INTEGER -- DECLARED IN SEGMENT 2 AT 00504150
       *02234500*
CARDLAST -- LABEL -- DECLARED IN SEGMENT 12 AT 02210000 -- OCCURS AT 02216000
        02210750
CARDNUMBER -- INTEGER -- DECLARED IN SEGMENT 2 AT 00504100
       *02214260**02234800* 02882200  02910300  13310580  15085100
CARDONLY -- LABEL -- DECLARED IN SEGMENT 12 AT 02210000 -- OCCURS AT 02215250
        02210750
CARDOPTION -- LABEL -- DECLARED IN SEGMENT 19 AT 02363000 -- OCCURS AT 02429000
        02436000
CASEADDRESS -- REAL ARRAY -- DECLARED IN SEGMENT 66 AT 07646335
       *07646510* 07646635
CASESTATEMENT -- PROCEDURE -- DECLARED IN SEGMENT 3 AT 07646325
        07759200
CASEV -- DEFINE -- DECLARED IN SEGMENT 3 AT 01229000
CBUFF -- REAL ARRAY -- DECLARED IN SEGMENT 3 AT 01561056
        02213750  02214000  02215500  02215750  02216250  02216500  07025010  07029000
CD -- STREAM VARIABLE -- NAME PARAMETER -- DECLARED IN SEGMENT 67 AT 07653500
        07654500  07655000
CDC -- DEFINE -- DECLARED IN SEGMENT 3 AT 01638000
        15301400  15306600
CELL -- STREAM VARIABLE -- VALUE PARAMETER -- DECLARED IN SEGMENT 43 AT 05325040
        05325050  05325130  05325150  05325180 *05325220* 05325330  05325350  05325360  05325380  05325390  05325400
        05325410  05325420
CF -- DEFINE -- DECLARED IN SEGMENT 119 AT 17002007
        17025100  17025300  17080000
CHANGFSEQ -- STREAM PROCEDURE -- DECLARED IN SEGMENT 3 AT 01741200
        01741300  02193250
CHAR -- DEFINE -- DECLARED IN SEGMENT 3 AT 01299040
        14461100
CHAR -- REAL -- NAME PARAMETER -- DECLARED IN SEGMENT 86 AT 10234000
        10242000  10252000  10255000
CHARCOUNT -- OWN INTEGER -- DECLARED IN SEGMENT 86 AT 10230000
        10242000  10250000  10251000 *10256000**10261000* 10285000
CHECKBIT -- DEFINE -- DECLARED IN SEGMENT 3 AT 01000920
```

```
        02496000
CHECKER  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 05113000
        05120000  05216000  07070500  07398000  08054000  12044000  15085000
CHECKTOG  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01001210
CHKSOB  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 13209000
        14289000
CHS  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01639000
        06021000  08189000  08219000
CIV  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01238000
CLASS  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01148000
        02705100  02850000  02861000  02886000  02910200  02910400  02920000  02924000  02966500  02968000  04509000
        05350120  07044000  08057000  12120000  12125000  13201000  13226000  13677200  14062000  14091000  14425000
        14519000  15085200  15098000  15128100  15185200  15276020  15276090  15295100  15301500  15304200  15305300
        17095000
CLCR  --  INTEGER  --  DECLARED IN SEGMENT 3 AT 01330000
       *02216500**02217750* 02218000  02218250  02224500
CLCR  --  REAL  --  NAME PARAMETER  --  DECLARED IN SEGMENT 12 AT 02202750
        02202500  02203500  02209000
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
        15093400  15290000  15306600
COCT  --  REAL STREAM PROCEDURE  --  DECLARED IN SEGMENT 67 AT 07653500
       *07656500* 07657000  07671000
CODE  --  FILE  --  DECLARED IN SEGMENT 3 AT 01556900
        04147000  04296000  05312000  07662000  07668500  07669000  07671500  08999200  08999575  08999700  09393350
        09393400  09393430  09400000  09401000  13651000  13746000  16495100
COde  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 04130000
        04134000
CODISK  --  FILE  --  DECLARED IN SEGMENT 3 AT 01561300
        08999150  08999175  08999275  08999300  08999350  08999375  08999550  08999575  08999600  08999700  08999725
        09398000
COL  --  INTEGER  --  DECLARED IN SEGMENT 44 AT 05344000
COLON  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01260000
        05276000  06318000  06320900  07597500  07600100  07600600  07604020  07727010  07727030  07727070  13753100
        13756000  13760000  14269480  16160000
COLON  --  LABEL  --  DECLARED IN SEGMENT 24 AT 02637000  --  OCCURS AT 02663000
        02641000
COM  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01641000
COMCOUNT  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 9 AT 02089500
        02090500  02111500  02118000
COMMA  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01261000
        06114000  07067000  07095000  07604140  07673000  08145000  08150000  08153000  10263700  10282000  12010000
        12020000  12033000  12052000  12056000  12125000  13351000  13403000  13409000  13756000  13762000  14197000
        14223000  14259110  14268000  14269840  14269920  14356000  14404000  15287000  15293000
COMMANTS  --  STREAM LABEL  --  DECLARED IN SEGMENT 9 AT 02092000  --  OCCURS AT 02105000
        02097500
COMMENTS  --  STREAM LABEL  --  DECLARED IN SEGMENT 9 AT 02092000  --  OCCURS AT 02106500
        02107500  02108000
COMMENTV  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01253000
        02886000  02920000  08034000
COMMON  --  LABEL  --  DECLARED IN SEGMENT 57 AT 07037000  --  OCCURS AT 07094000
        07045000  07054000  07058000  07064000  07074000  07076000  07084000
COMPAR  --  LABEL  --  DECLARED IN SEGMENT 12 AT 02210500  --  OCCURS AT 02224250
        02216750  02218500
```

```
COMPARE  -- REAL STREAM PROCEDURE  -- DECLARED IN SEGMENT 3 AT 02188000
        *02190000* 02190250  02203500  02226000  02228150  02232000  02233500  02411000  02486000
COMPARECODE  -- DEFINE  -- DECLARED IN SEGMENT 114 AT 16192000
         16208000
COMPLETE  -- LABEL  -- DECLARED IN SEGMENT 24 AT 02637000  -- OCCURS AT 02909000
         02658000  02667000  02680000  02705200  02715000  02757000  02852000  02877000  02877020  02893000  02894000
COMPOST  -- LABEL  -- DECLARED IN SEGMENT 24 AT 02639000  -- OCCURS AT 02882000
         02880000
COMPOUNDTAIL  -- PROCEDURE  -- DECLARED IN SEGMENT 3 AT 07006000
         07033000  07768140  07768200  13772000  14412000  14512000  14524000  16489000
COMPS1  -- BOOLEAN STREAM PROCEDURE  -- DECLARED IN SEGMENT 122 AT 17029000
        *17034000* 17042300
COMP1  -- BOOLEAN PROCEDURE  -- DECLARED IN SEGMENT 122 AT 17042100
        *17042300**17042350* 17045000
COMP2  -- BOOLEAN PROCEDURE  -- DECLARED IN SEGMENT 124 AT 17115000
        *17117000* 17119000
CONSTANA  -- OWN BOOLEAN  -- DECLARED IN SEGMENT 69 AT 08013000
         08074000  08099000  08181000  08188000 *08211000**08224000**08227000*
CONSTANB  -- OWN BOOLEAN  -- DECLARED IN SEGMENT 69 AT 08013000
         08043000  08112000  08116000 *08121000* 08183000  08194000 *08205000* 08213000
CONSTANC  -- OWN BOOLEAN  -- DECLARED IN SEGMENT 69 AT 08013000
         08185000  08195000 *08204000* 08218000
CONSTANTCLEAN  -- PROCEDURE  -- DECLARED IN SEGMENT 3 AT 04163000  -- FORWARD AT 03034000
         04217000  07459000  07496000  07519000  07585000  08170000  08190000  13744000  14040000  14168500  14604000
CONV  -- REAL STREAM PROCEDURE  -- DECLARED IN SEGMENT 3 AT 02041000
         02043000  02214260  02234900  02251000  02255000  02259000  02502000  02581000  02592000
CONVERT  -- REAL PROCEDURE  -- DECLARED IN SEGMENT 3 AT 02248000
        *02260000* 02758000  02796000  02803000  02823000
COP  -- REAL ARRAY  -- DECLARED IN SEGMENT 3 AT 01371000
        *02550000* 04280000  04283500
CORADR  -- INTEGER  -- DECLARED IN SEGMENT 3 AT 01344000
         05248500 *05250000* 09282600  09388000  13653000
CORE  -- STREAM VARIABLE  -- VALUE PARAMETER  -- DECLARED IN SEGMENT 80 AT 09370000
         09371000  09375000
CORNER  -- STREAM VARIABLE  -- VALUE PARAMETER  -- DECLARED IN SEGMENT 3 AT 02011000
         02012000
COUNT  -- INTEGER  -- DECLARED IN SEGMENT 3 AT 01313000
         02128500  02129000  02251000  02252000 *02327000* 02366200 *02371000* 02373000 *02499000**02590000**02602700*
        *02647000* 02685000  02686000 *02691000* 02695000 *02698000**02699000* 02703000  02705000  02790000  02796000
         02798000  02803000  02810000  02817000  02880000  02881000 *02889000**02958500* 07668000  07670000  07671000
         12109000 *12121000* 13302000  14259010  16392000
COUNT  -- STREAM VARIABLE  -- VALUE PARAMETER  -- DECLARED IN SEGMENT 3 AT 02062500
         02064000
COUNT  -- STREAM VARIABLE  -- NAME PARAMETER  -- DECLARED IN SEGMENT 9 AT 02090000
         02121500
COUNT  -- STREAM VARIABLE  -- VALUE PARAMETER  -- DECLARED IN SEGMENT 37 AT 05016000
         05017000  05028000
COUNT  -- STREAM VARIABLE  -- VALUE PARAMETER  -- DECLARED IN SEGMENT 38 AT 05067000
         05071000  05077000 *05080000*
COUNT  -- STREAM VARIABLE  -- DECLARED IN SEGMENT 43 AT 05325070
        *05325150* 05325160 *05325210* 05325320  05325330
COUNT  -- REAL  -- DECLARED IN SEGMENT 49 AT 06062000
         06074000 *06075000* 06103000 *06112000**06119000* 06120000  06121000  06122000
COUNT  -- STREAM VARIABLE  -- VALUE PARAMETER  -- DECLARED IN SEGMENT 87 AT 10236000
         10237000  10240000
COUNT  -- INTEGER  -- DECLARED IN SEGMENT 87 AT 10241000
        *10242000* 10245000 *10247000* 10249000  10252000  10255000  10256000
```

```
COUNTV  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 9 AT 02089500
        02090500  02121000  02123500
CPLUS1  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01713000
CPLUS2  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01714000
CREL  --  BOOLEAN  --  DECLARED IN SEGMENT 31 AT 04167000
       *04173000**04183000* 04187000
CRF  --  DEFINE  --  DECLARED IN SEGMENT 108 AT 16010000
        16121000  16200000  16370000  16408000
CROSSHATCH  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01262000
        10279000
CROSSHATCH  --  LABEL  --  DECLARED IN SEGMENT 24 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  15090600  15184400  15301000
D  --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 8 AT 02001760
        02001775  02001780
D  --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 3 AT 02001838
        02001842  02001848  02001850
D  --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 3 AT 02001858
        02001890  02001895  02001896
D  --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 15 AT 02264400
        02264500
D  --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 15 AT 02264700
        02264800
D  --  INTEGER  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 03047100
        03047000
D  --  INTEGER  --  DECLARED IN SEGMENT 31 AT 04166000
       *04175000* 04184000
D  --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 67 AT 07658500
        07659500  07660000
D  --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 86 AT 10257100
        10257200
D  --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 89 AT 12103000
        12104000
D  --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 102 AT 14254200
        14254300
D  --  REAL  --  DECLARED IN SEGMENT 109 AT 16031000
       *16032000*
D  --  REAL  --  DECLARED IN SEGMENT 110 AT 16067000
       *16070000* 16079000
D  --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 124 AT 17047200
        17047800
D  --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 124 AT 17051400
        17051700
D  --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 124 AT 17052600
        17052800  17053000
DA  --  INTEGER  --  DECLARED IN SEGMENT 3 AT 01559020
DATE  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 5 AT 01823000
        01825000
DATER  --  ALPHA STREAM PROCEDURE  --  DECLARED IN SEGMENT 5 AT 01823000
        01825000  01827000  01833000
DATIME  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 01820000
        01837000  02195000  02195750  02196590  02214200  02228750  02264950  02421000  05039600  05325490  07025020
        13653500  13676000  17091170
DBLSTMT  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 12002000  --  FORWARD AT 03060000
        07753000  12059000
```

```
DBV -- DEFINE -- DECLARED IN SEGMENT 3 AT 01243000
      16417000
DCINTYPE -- DEFINE -- DECLARED IN SEGMENT 3 AT 01561420
      09430060
DCV -- DEFINE -- DECLARED IN SEGMENT 3 AT 01248000
      16208000  16338000
DDFS -- DEFINE -- DECLARED IN SEGMENT 3 AT 01299000
      07677000
DEBLANK -- STREAM LABEL -- DECLARED IN SEGMENT 9 AT 02091500 -- OCCURS AT 02100000
      02094500
DEBUG -- PROCEDURE -- DECLARED IN SEGMENT 3 AT 04277500
      04286000  04297000
DEBUGBIT -- DEFINE -- DECLARED IN SEGMENT 3 AT 01000930
      02531000  02533000  02602500  04148000  04297000
DEBUGTOG -- DEFINE -- DECLARED IN SEGMENT 3 AT 01001220
      02533000  02602500  04148000  04297000
DEBUGWORD -- STREAM PROCEDURE -- DECLARED IN SEGMENT 3 AT 04130000
      04149000
DECK -- FILE -- DECLARED IN SEGMENT 3 AT 01561500
      09435000  09447000  09447090
DECKBIT -- DEFINE -- DECLARED IN SEGMENT 3 AT 01000940
      02474000  09405000  09414010  09421000  09447010  13651100
DECKTOG -- DEFINE -- DECLARED IN SEGMENT 3 AT 01001230
      09405000  09414010  09421000  09447010  13651100
DECLARATORS -- DEFINE -- DECLARED IN SEGMENT 3 AT 01214000
      07767000  10277000  14126000  14370000  14396000  14516000  16434000
DECLARELABEL -- PROCEDURE -- DECLARED IN SEGMENT 3 AT 16000100
      16253100  16482200
DECLREF -- DEFINE -- DECLARED IN SEGMENT 3 AT 01007490
      13310525  13310580  14312100  17092900
DECLSW -- SWITCH LABEL -- DECLARED IN SEGMENT 98 AT 14018000
      14134000
DEFINEARRAY -- REAL 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 *10262000**10278000* 10280000 *10283000**10284000*
DEFINEDEC -- LABEL -- DECLARED IN SEGMENT 98 AT 14017000 -- OCCURS AT 14254000
      14021000
DEFINEDID -- DEFINE -- DECLARED IN SEGMENT 3 AT 01180000
      02894000  12120000  14259010
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*
DEFINEV -- DEFINE -- DECLARED IN SEGMENT 3 AT 01298000
      10278000
DEFINING -- BOOLEAN -- DECLARED IN SEGMENT 3 AT 01007650
     *02419110* 02910100  13283500  13310900 *14255500**14268500* 14445100  16495300
DEL -- DEFINE -- DECLARED IN SEGMENT 3 AT 01642000
DEPTH -- REAL PROCEDURE -- DECLARED IN SEGMENT 3 AT 05400000
     *05402100**05403000* 05407000 *05409000* 05426000  05431000
DESC -- DEFINE -- DECLARED IN SEGMENT 69 AT 08016000
      08098000  08108000
DEST -- STREAM VARIABLE -- NAME PARAMETER -- DECLARED IN SEGMENT 3 AT 02045000
      02048000
DIA -- DEFINE -- DECLARED IN SEGMENT 3 AT 01680000
```

```
        04315000  04550000
DIALA  --  INTEGER  --  DECLARED IN SEGMENT 3 AT 01502000
       *04315000**08152000*
DIALB  --  INTEGER  --  DECLARED IN SEGMENT 3 AT 01502000
       *04317000**08152000*
DIB  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01681000
        04317000
DIQ  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01237000
DISK  --  FILE  --  DECLARED IN SEGMENT 3 AT 01561400
        01561600  01828500  09425900  09429000  09430075  09430100
DISK  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 80 AT 09370000
        09371000  09376000
DISKADR  --  INTEGER  --  DECLARED IN SEGMENT 3 AT 01344000
        05252000 *05254000**09253100* 09389000  13653000
DOIT  --  STREAM PROCEDURE  --  DECLARED IN SEGMENT 44 AT 05334100
        05350160  05350200
DOIT  --  STREAM PROCEDURE  --  DECLARED IN SEGMENT 96 AT 13673100
        13673450  13677300
DOLLAR  --  LABEL  --  DECLARED IN SEGMENT 24 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 3 AT 01691500
       *02211800**02230100* 02234600  02421000
DOSTMT  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 07482000
        07488000  07759000
DOT  --  LABEL  --  DECLARED IN SEGMENT 24 AT 02637000  --  OCCURS AT 02668000
        02641000
DOT  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 06338000  --  FORWARD AT 03015000
        06178000  06346000
DOTSYNTAX  --  BOOLEAN PROCEDURE  --  DECLARED IN SEGMENT 3 AT 05270000
       *05286000* 06341000  15087800  15183400  15276150  15300100
DOV  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01225000
        07494000  08159000  08186000
DPTOG  --  BOOLEAN  --  DECLARED IN SEGMENT 3 AT 01690000
        02253000  02784000  02794000  02829000 *12012000**12016000*
DSKIP  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 02045000
        02046000  02049000
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 108 AT 16362000
        16398000  16491000
DSV  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01240000
DTYPE  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01452000
DUMMY  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01001470
DUMPDFC  --  LABEL  --  DECLARED IN SEGMENT 98 AT 14014000  --  OCCURS AT 14149000
        14019000
DUMPEF  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01547000
DUMPINFO  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 02264000
        02266100  02468000
DUMPOR  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01552000
DUMPV  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01287000
        07524000  07646610  15093200  15095900  15185800  15302500  15303000
```

```
DUPIT  --  BOOLEAN  --  DECLARED IN SEGMENT 107 AT 15076400
       *15091600* 15092800  15095400  15095700  15096400  15096800 *15184800* 15185100  15185600  15186100  15186300
       *15302000* 15302300  15302800  15303400
DWA  --  INTEGER  --  DECLARED IN SEGMENT 15 AT 02264100
       *02265500* 02265600  02265700
E  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 13293000  --  FORWARD AT 03054000
        12123000  13333500  13342000  13755500  16000600
E  --  LABEL  --  DECLARED IN SEGMENT 29 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 36 AT 04504000
        04503000  04506000  04508000 *04509000* 04522000
E  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 89 AT 12103000
        12105000
E  --  REAL  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 108 AT 16065000
        16070000
E  --  REAL  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 108 AT 16085000
        16089000  16093000
EDITLINE  --  STREAM PROCEDURE  --  DECLARED IN SEGMENT 3 AT 02183500
        02187000  02195000  02195750  02214200  02228750  02421000  05038000  07025020
EDOC  --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 3 AT 13683000
        13685000  13687000  13693000
EDOCINDEX  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01710000
EL  --  LABEL  --  DECLARED IN SEGMENT 49 AT 06061900  --  OCCURS AT 06104300
        06107200
EL  --  REAL  --  DECLARED IN SEGMENT 89 AT 12102000
       *12112000* 12114000 *12125000* 12126000
ELBAT  --  REAL ARRAY  --  DECLARED IN SEGMENT 3 AT 01319000
        02265100  02265250 *02910000* 02916000  02920000  02923000 *02924000* 05089000  05189000  05216000  05217000
        05218000  05222000  05273300  05273400  05274100  05274200  05280000  05281000  05344570  05344615  05344620
        05344660  05344700  05344710  05350120  05350160  05350180  06016000  06044000  06072000  06104300  06108000
        06166000  06185000  06315300  06315400  06316300  06316400  06320100  06320200  06324000  06325000  06326000
       *07031000* 07070000  07070500  07071000  07396000  07430000  07516000  07516300  07518000  07521000  07554000
        07566000  07572000  07583000  07599000 *07646525* 07667000  07745000  07768100  07994000  07995000  08029000
       *08034000* 08037000  08176000  09424000  09425900  09427500  09429000 *09432000* 09433000 *09434000* 09435000
       *09437000* 09447000 *09447010* 09447070 *09447080* 09447090  10277000  12015000  12024000  12036000  12044000
        12125000  13200000  13310350  13341300  13361000  13372000  13394000 *13767000* 13768000 *14062000* 14129000
        14220000  14221000  14269446  14269448  14269520  14269540  14314000  14349000  14352000  14353000  14399000
       *14519000* 15077000  15276010 *16000700* 16121000  16124000  16159000  16200000  16204000  16208000  16211000
        16212500  16254000  16321000  16329000  16342000  16344000  16370000  16376000  16379000  16408000  16412000
        16435000 *16449000*
ELBATWORD  --  REAL  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 05113000
        05116000  05118000
ELBATWORD  --  REAL  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 05207000
        05206000  05207000  05208000  05209000
ELBATWORD  --  DEFINE  --  DECLARED IN SEGMENT 65 AT 07595000
        07599000  07601000
ELBW  --  REAL  --  DECLARED IN SEGMENT 62 AT 07503000
       *07521000* 07527000
ELCLASS  --  INTEGER  --  DECLARED IN SEGMENT 3 AT 01328000
       *05002000**05003000* 05109000  05273800  05275000  05304000  05344550  06004000  06013000  06034000  06036000
        06048000  06052000  06055000  06076000  06084000  06104100  06105000  06106300  06107100  06114000  06115000
        06138000  06139000  06141000  06146000  06151000  06175000  06180000  06198000  06302000  06315100  06316000
        06317300  06320900  06340000  06411000  06416000  06422000  07012000  07013000  07018000  07021000  07026000
       *07032000* 07052000  07053000  07075000  07084000  07095000  07096000  07399000  07404000  07409000  07431000
        07485000  07494000  07506000  07507000  07511000  07550000  07553000 *07556000* 07578000  07600300  07600400
```

```
        07604110  07646490 *07646525* 07646535  07646580  07646603  07665000  07665500  07666000  07666500  07668000
        07727000  07727010  07727030  07727050  07728000 *07749000**07751000**07768130* 08031000  08032000  08038000
        08103000  08113000  08128000  08133000  08139000  08145000  08150000  08153000  08159000  08182000  08184000
        08186000  08229000  12022000  12028000  12033000  12035000  12039000  12042000  12052000  12056000  13331000
        13337000 *13338000* 13391000  13400000  13403000  13409000  13756000  13758000  13760000  13762000  13763000
        14137000  14165000  14197000  14259020  14259120  14260000  14269280  14269442  14269460  14269680  14269840
        14269920  14325000  14326000  14336000  14357000  14370000  14396000  14404000  14479000 *14516000* 15078000
        15083000  15086200  15087600  15088600  15183300  15183800  15276010  15276140  15287000  15294000  15300500
        15305000  16119000  16131000  16132000  16198000  16204000  16210000  16212500  16213000  16230000  16253000
        16315000  16319000  16320000  16326000  16330000  16335000  16336000  16337000  16338000  16339000  16341000
        16374000  16379000  16381000  16410000  16415000  16417000  16437000  16477000  16478000  16481000  16482100
        16482300  16482500
ELCLASS  -- REAL  -- DECLARED IN SEGMENT 86 AT 10258100
       *10263400* 10263700  10264100  10264300  10264410  10277000  10279000 *10282000*
ELCLASS  -- DEFINE  -- DECLARED IN SEGMENT 91 AT 13220000
        13226000  13246000  13278000
ELSEBRANCH  -- INTEGER  -- DECLARED IN SEGMENT 53 AT 06295000
       *06300000* 06307000
ELSEV  -- DEFINE  -- DECLARED IN SEGMENT 3 AT 01227000
        06302000  07567000  07578000  16220000  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  06054000  06054100  06082000  06087000  06089000  06099100
        06104600  06148000  06166000  06191000  06308000  07093100  07431000  07436000  07544000  07604070  07646460
        07646550  07995000  08049000  08050000  08122000  08148000  08151000  08227000  14451100  15376700  16080000
        16122000  16133000  16219000  16232000  16266000  16387000  16394000  16453000
EMITB  -- PROCEDURE  -- DECLARED IN SEGMENT 3 AT 04114000  -- FORWARD AT 03036000
        04125000  04218000  06207000  06301000  06307000  07088000  07444000  07487000  07495000  07496000  07540000
        07579000  07585100  07586000  07608000  07646605  07646635  07646665  08047000  08078000  08086000  08127000
        08146000  08168000  08215000  08221000  13606000  14041000
EMITC  -- PROCEDURE  -- DECLARED IN SEGMENT 3 AT 04010000
        04014000  14451000  16032000  16079000  16121000  16124000  16200000  16208000  16211000  16212500  16213000
        16216000  16342000  16352000  16370000  16378500  16379000  16383000  16408000  16415000  16417000
EMITD  -- PROCEDURE  -- DECLARED IN SEGMENT 3 AT 04311000  -- FORWARD AT 03050000
        04322000  04556000  05203000  06332000  07093000  07093200  15097600  15186700  15276070  15303800
EMITI  -- PROCEDURE  -- DECLARED IN SEGMENT 3 AT 04500000
        06344000  15096000  15103400  15185900  15187300  15276210  15303100  15307200
EMITJUMP  -- PROCEDURE  -- DECLARED IN SEGMENT 108 AT 16065000
        16093000  16255000
EMITL  -- PROCEDURE  -- DECLARED IN SEGMENT 3 AT 04003000
        04032000  04205000  04544000  05222000  06110000  06206000  08046000  08067000  08077000  13264000  13267000
        13773000  14506500
EMITLNG  -- PROCEDURE  -- DECLARED IN SEGMENT 3 AT 04098000
        04112000  06148000  07573000  07575000  08150000  08220000
EMITN  -- PROCEDURE  -- DECLARED IN SEGMENT 3 AT 04022000
        04024000  04212000  04524000  05202000  07092000  08075000  08179000  08212000  12036000  15086400  15092600
        15185500  15276080  15301200  15305500  15306400  16261000
EMITNUM  -- PROCEDURE  -- DECLARED IN SEGMENT 3 AT 04203000  -- FORWARD AT 03049000
        04121000  04225000  06168000  07089000  07442000  08021000  08089000  12014000  12015000  13229000
EMITO  -- PROCEDURE  -- DECLARED IN SEGMENT 3 AT 04006000
        04018000  04023000  04033000  04034000  04109000  04112000  04122000  04212100  04223100  04311090  04507000
        04531000  04537000  04545000  05226000  05316000  06021000  06077000  06110300  06172500  06187000  06203000
        07078000  07087000  07091000  07408000  07434000  07512000  07516400  07524000  07525000  07526000  07646465
        07646555  07646610  07646615  07646620  08044000  08049000  08050000  08065000  08068000  08070000  08079000
        08094000  08123000  08125000  08126000  08166000  08189000  08214000  08219000  08228000  12024000  12038000
        12054000  13263000  13264000  13268000  13740000  13743000  13752000  13761000  13765000  13766100  14167000
        14389000  14451000  15093200  15093400  15095900  15100400  15100800  15101000  15185800  15276090  15290000
        15301400  15301600  15302500  15302600  15303000  15303900  15304100  15304200  15305600  15305800  15306600
```

```
EMITPAIR  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 04028000
        04034000  04523000  05201000  05227000  07071000  07090000  08090000  13230000  15101600  15185300  15186800
        15267000
EMITSTORE  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 13214000
        13230000
EMITV  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 04017000
        04019000  04184000  04184500  04521000  05219000  05226000  05316000  07413000  07439000  07516300  08021000
        08044000  08075000  08166000  08212000  12047000  13236000  13739000  15086800  15095600  15276060  15290000
        15306200
EMITWORD  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 04142000
        04152000  04187000  07604120
EMIT21  --  PROCEDURE  --  DECLARED IN SEGMENT 36 AT 04503000
        04526200  04530000  04536000  04543000  04549000  04555000
ENDOFIT  --  INTEGER  --  DECLARED IN SEGMENT 66 AT 07646405
       *07646625* 07646640
ENDOFITALL  --  LABEL  --  DECLARED IN SEGMENT 3 AT 05101100  --  OCCURS AT 16495210
ENDREADTAPE  --  LABEL  --  DECLARED IN SEGMENT 13 AT 02201510  --  OCCURS AT 02202080
        02202010
ENDTOG  --  BOOLEAN  --  DECLARED IN SEGMENT 3 AT 01415000
      *02653000* 02680000  02884000 *06021000**06022000**07016000* 07019000 *07020000**07646601* 07646603 *07646604*
ENDV  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01228000
        02910400  05109000  07013000  07018000  07646475  07646580  07646603  07748000  07749000  16477000  16133000
ENTER  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 13714000  --  FORWARD AT 03069000
        13390000  14138000  14139000  14140000  14141000  14162000
ENTRY  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 13314000  --  FORWARD AT 03055000
        13729000  14165000  14194000  14207000  14259010  14269240  14319000  14333000  14403000  14409000
EOF  --  LABEL  --  DECLARED IN SEGMENT 12 AT 02210250  --  OCCURS AT 02217000
        02216250
EOF  --  LABEL  --  DECLARED IN SEGMENT 71 AT 08999050  --  OCCURS AT 08999400
        08999150  08999375
EOF  --  LABEL  --  DECLARED IN SEGMENT 81 AT 09394000  --  OCCURS AT 09413000
        09399000
EOF  --  LABEL  --  DECLARED IN SEGMENT 123 AT 17009000  --  OCCURS AT 17012000
        17010000
EOF  --  LABEL  --  DECLARED IN SEGMENT 127 AT 17073000  --  OCCURS AT 17082000
        17076000  17079000
EOFT  --  LABEL  --  DECLARED IN SEGMENT 13 AT 02201510  --  OCCURS AT 02202020
        02201750
EOF2  --  LABEL  --  DECLARED IN SEGMENT 129 AT 17091100  --  OCCURS AT 17096800
        17093950
EPART  --  LABEL  --  DECLARED IN SEGMENT 24 AT 02639000  --  OCCURS AT 02810000
EQL  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01644000
EQUAL  --  BOOLEAN STREAM PROCEDURE  --  DECLARED IN SEGMENT 3 AT 02062500
       *02064000* 02064500  02881000
EQVOP  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01274000
        02965500  02971000  06036000  15091600  15184800  15302000
ERR  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 05105000  --  FORWARD AT 01718000
        02211500  04145000  04300000  05109000  05285000  06005000  06086000  06095000  06107000  06109000  06116000
        06142000  06151000  06154000  06176000  06302000  06316700  06334000  06411000  06415000  06417000  06421000
        06423000  07015000  07053000  07097000  07099000  07405000  07412000  07437000  07485000  07494000  07513000
        07517000  07598000  07600700  07604130  07646458  07646585  07667500  07672500  07674000  07727080  07732000
        07994000  08059000  08134000  08160000  08178000  08229000  12007000  12033000  12039000  12052000  12056000
        14259090  14259120  15081000  15089800  15102600  15184300  15256000  15261000  15276170  15276190  15294000
        15297000  15300900  15306800  16125000  16132000  16160000  16205000  16206000  16207000  16214000  16217000
        16253100  16316000  16347000  16367000  16384500  16397000  16418000  16483000
ERREXIT  --  DEFINE  --  DECLARED IN SEGMENT 107 AT 15076140
        15089800  15102600  15184300  15300900  15306800
```

```
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  05091000
ERRNUM  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 37 AT 05016000
        05017000  05026000
ERRNUM  --  INTEGER  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 05105000
        05106000  05107100  05107200
ERROR  --  STREAM LABEL  --  DECLARED IN SEGMENT 9 AT 02091500  --  OCCURS AT 02114000
        02119000
ERROR  --  DEFINE  --  DECLARED IN SEGMENT 25 AT 02958000
        02959000
ERROR  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 05110000
        07667500  07672500  07674000
ERRORCOUNT  --  INTEGER  --  DECLARED IN SEGMENT 2 AT 00501000
        02211500 *05035000* 05087000  09028000 *09028900* 09386000  09391000
ERRORTOG  --  BOOLEAN  --  DECLARED IN SEGMENT 3 AT 01412000
        05032000 *05047000**05119000**07009000**07063000**07645000**07646485**07750000**13282000**14130000**14443000*
        14511000 *16130000*
EXAMIN  --  REAL STREAM PROCEDURE  --  DECLARED IN SEGMENT 2 AT 00511000
        00512000  00529000  00531000  00533000  00536000  02214200  02225750  02228250  02230000  02230100  02665000
        02680000  02697000  02745000  02750000  02753000  02786000  02791000  02807000  02813000  02888000
EXIT  --  STREAM LABEL  --  DECLARED IN SEGMENT 3 AT 01758000  --  OCCURS AT 01782000
        01763000  01764500
EXIT  --  STREAM LABEL  --  DECLARED IN SEGMENT 9 AT 02091500  --  OCCURS AT 02119500
        02104500  02112000  02114500  02115500  02118500
EXIT  --  LABEL  --  DECLARED IN SEGMENT 12 AT 02210000  --  OCCURS AT 02225000
        02215750
EXIT  --  LABEL  --  DECLARED IN SEGMENT 19 AT 02360000  --  OCCURS AT 02602000
        02375000  02411000  02413000  02486000  02488000
EXIT  --  LABEL  --  DECLARED IN SEGMENT 25 AT 02957000  --  OCCURS AT 02967500
        02959000  02962500
EXIT  --  LABEL  --  DECLARED IN SEGMENT 35 AT 04311010  --  OCCURS AT 04322100
        04311100
EXIT  --  LABEL  --  DECLARED IN SEGMENT 36 AT 04501000  --  OCCURS AT 04558000
        04526200  04532000  04538000  04546000  04552000
EXIT  --  LABEL  --  DECLARED IN SEGMENT 38 AT 05049000  --  OCCURS AT 05100000
        05094000
EXIT  --  LABEL  --  DECLARED IN SEGMENT 41 AT 05273000  --  OCCURS AT 05286000
        05273500  05274400  05284000
EXIT  --  LABEL  --  DECLARED IN SEGMENT 43 AT 05325030
EXIT  --  LABEL  --  DECLARED IN SEGMENT 44 AT 05334000  --  OCCURS AT 05375000
        05344760  05372000
EXIT  --  LABEL  --  DECLARED IN SEGMENT 47 AT 05439000  --  OCCURS AT 05458000
        05454000  05457000
EXIT  --  LABEL  --  DECLARED IN SEGMENT 49 AT 06061000  --  OCCURS AT 06125000
        06069000  06086000  06096000  06107000  06109000  06116000
EXIT  --  LABEL  --  DECLARED IN SEGMENT 50 AT 06137000  --  OCCURS AT 06182000
        06142000  06149000  06154000  06176000
EXIT  --  LABEL  --  DECLARED IN SEGMENT 52 AT 06195000  --  OCCURS AT 06209000
        06200000  06204000
EXIT  --  LABEL  --  DECLARED IN SEGMENT 54 AT 06315000  --  OCCURS AT 06335000
        06316800  06333000
EXIT  --  LABEL  --  DECLARED IN SEGMENT 55 AT 06339000  --  OCCURS AT 06346000
        06341000
```

```
EXIT  --  LABEL  --  DECLARED IN SEGMENT 57 AT 07037000  --  OCCURS AT 07101000
          07097000  07099000
EXIT  --  LABEL  --  DECLARED IN SEGMENT 58 AT 07394000  --  OCCURS AT 07425000
          07402000  07405000
EXIT  --  LABEL  --  DECLARED IN SEGMENT 62 AT 07504000  --  OCCURS AT 07531000
          07514000  07516600  07517000  07519000
EXIT  --  LABEL  --  DECLARED IN SEGMENT 63 AT 07549000  --  OCCURS AT 07557000
          07555000
EXIT  --  LABEL  --  DECLARED IN SEGMENT 64 AT 07562000  --  OCCURS AT 07587000
          07573000  07574000  07576000  07579100  07584000
EXIT  --  LABEL  --  DECLARED IN SEGMENT 65 AT 07594000  --  OCCURS AT 07646000
          07598000  07600700
EXIT  --  LABEL  --  DECLARED IN SEGMENT 68 AT 07726000  --  OCCURS AT 07771000
          07727080  07732000  07733000  07735000  07737000  07739000  07743000  07746000  07749000  07751000  07753000
          07755000  07757000  07759000  07759200  07761000  07763000  07765000  07768160
EXIT  --  LABEL  --  DECLARED IN SEGMENT 69 AT 08017000  --  OCCURS AT 08232000
          08178000  08222000  08229000
EXIT  --  LABEL  --  DECLARED IN SEGMENT 70 AT 08092000  --  OCCURS AT 08172000
          08134000  08160000
EXIT  --  LABEL  --  DECLARED IN SEGMENT 86 AT 10259000  --  OCCURS AT 10284000
          10263900  10275000
EXIT  --  LABEL  --  DECLARED IN SEGMENT 88 AT 12005000  --  OCCURS AT 12058000
          12033000  12040000  12052000  12056000
EXIT  --  LABEL  --  DECLARED IN SEGMENT 103 AT 14269100  --  OCCURS AT 14269860
          14269710  14269760
EXIT  --  LABEL  --  DECLARED IN SEGMENT 107 AT 15076000  --  OCCURS AT 15377000
          15081000  15087000  15088000  15089800  15102600  15183500  15184300  15187400  15256000  15261000  15275000
          15276120  15276150  15276170  15276190  15276230  15294000  15297000  15300200  15300900  15301800  15306800
EXIT  --  LABEL  --  DECLARED IN SEGMENT 112 AT 16117000  --  OCCURS AT 16144000
          16125000  16132000
EXIT  --  LABEL  --  DECLARED IN SEGMENT 114 AT 16193000  --  OCCURS AT 16239000
          16205000  16206000  16207000  16214000  16217000
EXIT  --  LABEL  --  DECLARED IN SEGMENT 115 AT 16251000  --  OCCURS AT 16270000
          16253100
EXIT  --  LABEL  --  DECLARED IN SEGMENT 116 AT 16313000  --  OCCURS AT 16355000
          16316000  16342000  16347000
EXIT  --  LABEL  --  DECLARED IN SEGMENT 117 AT 16365000  --  OCCURS AT 16398000
          16367000  16384500
EXIT  --  LABEL  --  DECLARED IN SEGMENT 108 AT 16475000  --  OCCURS AT 16494000
          16484000  16487000  16490000  16491000  16492000  16493000
EXPECT  --  REAL  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 03009000
EXPECT  --  REAL  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 06060000
          06119000
EXPRSS  --  INTEGER PROCEDURE  --  DECLARED IN SEGMENT 3 AT 06057000  --  FORWARD AT 03007000
          06299000
F  --  INTEGER  --  DECLARED IN SEGMENT 3 AT 01688000
         *14077000**14078000**14115000*
F  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 02262000
          02263000
F  --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 21 AT 02441000
          02443000
F  --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 21 AT 02446000
          02448000
F  --  STREAM VARIABLE  --  DECLARED IN SEGMENT 38 AT 05068000
          05069000  05076000
F  --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 71 AT 08999075
          08999100
```

```
F  --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 81 AT 09393700
       09393710  09393720  09393730
F  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 96 AT 13673100
       13673350
F  --  REAL  --  DECLARED IN SEGMENT 109 AT 16031000
FACTOR  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01299100
       15253000
FCR  --  INTEGER  --  DECLARED IN SEGMENT 3 AT 01330000
       02194000  02195000  02195750 *02214000* 02214200 *02225250* 02225750  02228250  02228750  02230000  02230100
       02421000  02731000  05038000 *07025010* 07025020 *07025030*
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 21 AT 02441000
       02444000  02445000  02452000
FETCH  --  REAL STREAM PROCEDURE  --  DECLARED IN SEGMENT 3 AT 02262000
       02263000  08999175  09400000  09401000
FF  --  DEFINE  --  DECLARED IN SEGMENT 119 AT 17002008
       17025100  17080000
FIEL  --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 3 AT 02056000
       02058000  02059000
FIEL  --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 80 AT 09364000
       09365000  09366000  09367000
FIEL  --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 95 AT 13633000
       13636000  13637000
FIEL  --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 96 AT 13661000
       13665000  13667000
FIELDDEC  --  LABEL  --  DECLARED IN SEGMENT 98 AT 14016000  --  OCCURS AT 14269020
       14021000
FIELDID  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01278700
       05273100  05273900  06315100  06316100  14269240  14269442  17095100
FIELDV  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01298600
FIL  --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 80 AT 09370000
       09372000
FILEDEC  --  LABEL  --  DECLARED IN SEGMENT 98 AT 14015000  --  OCCURS AT 14158000
       14020000
FILEID  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01299300
       14430000
FILEV  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01296000
FILLIT  --  STREAM PROCEDURE  --  DECLARED IN SEGMENT 43 AT 05325040
       05325440  05325460  05325470  05325480
FILLSTMT  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 07647000
       07677500  13396000
FINAL  --  LABEL  --  DECLARED IN SEGMENT 86 AT 10258300  --  OCCURS AT 10263900
       10264400  10264420
FINALAX  --  STREAM PROCEDURE  --  DECLARED IN SEGMENT 80 AT 09378000
       09384100
FINDOPTION  --  BOOLEAN PROCEDURE  --  DECLARED IN SEGMENT 3 AT 02307000
      *02317000* 02318000  02327000  02967000
FINI  --  LABEL  --  DECLARED IN SEGMENT 108 AT 16475000  --  OCCURS AT 16495000
       16476000  16478000  16483000  16486000  16489000
FINIS  --  STREAM LABEL  --  DECLARED IN SEGMENT 9 AT 02091500  --  OCCURS AT 02125500
       02095500  02096500  02099500  02110000
FINISHED  --  LABEL  --  DECLARED IN SEGMENT 32 AT 04204000  --  OCCURS AT 04225000
       04220000
FINISHNUMBER  --  LABEL  --  DECLARED IN SEGMENT 24 AT 02638000  --  OCCURS AT 02847000
```

```
FIRST  --  INTEGER  --  NAME PARAMETER  --  DECLARED IN SEGMENT 3 AT 05271000
       05270000 *05273300**05274100**05280000* 05282000
FIRST  --  INTEGER  --  DECLARED IN SEGMENT 54 AT 06314000
      *06315300**06316300**06320100**06324000* 06327000  06332000
FIRST  --  INTEGER  --  DECLARED IN SEGMENT 55 AT 06339000
       06341000  06344000
FIRSTTIME  --  LABEL  --  DECLARED IN SEGMENT 12 AT 02210000  --  OCCURS AT 02213500
       02210750
FIRSTTIME  --  BOOLEAN  --  DECLARED IN SEGMENT 124 AT 17069400
       17091155 *17091162**17117920*
FIRSTX  --  REAL  --  DECLARED IN SEGMENT 3 AT 01621000
       13608000 *13612000* 13623000  14051000 *14052000* 14600000 *14608000*
FIRSTXO  --  INTEGER  --  DECLARED IN SEGMENT 98 AT 14033000
      *14051000* 14608000
FIX  --  STREAM PROCEDURE  --  DECLARED IN SEGMENT 21 AT 02446000
       02451000  02454000
FIX  --  REAL  --  DECLARED IN SEGMENT 59 AT 07427000
      *07430000**07433000* 07439000  07440000
FIX  --  PROCEDURE  --  DECLARED IN SEGMENT 70 AT 08082000
       08090000  08157000  08171000
FIXC  --  PROCEDURE  --  DECLARED IN SEGMENT 108 AT 16029000
       16057000  16141000  16233000  16236000  16238000
FIXDEFINEINFO  --  REAL PROCEDURE  --  DECLARED IN SEGMENT 3 AT 12101000  --  FORWARD AT 01717900
       02896000 *12108000*
FIXHDR  --  STREAM PROCEDURE  --  DECLARED IN SEGMENT 81 AT 09393700
       09393750  09430075
FIX1  --  REAL  --  DECLARED IN SEGMENT 114 AT 16195000
      *16218000* 16233000  16238000
FIX2  --  REAL  --  DECLARED IN SEGMENT 114 AT 16195000
      *16232000* 16236000
FL  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01420000
       06140000  07050000  07058000  07073000  07083000  12037000  15085800  15276210
FLAG  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 05012000  --  FORWARD AT 01716000
       02226500  02316000  02421000  02501000  02593000  02599000  02650000  02686500  02696000  02824000  02828000
       02899000  02959000  02962000  02976500  02977500  04013000  04176000  05101000  05106000  05119000  05344610
       05344661  05347000  05348000  05370000  06104520  06110200  06200000  07024000  07025000  07061000  07600000
       07600420  07600900  07604000  07604068  07671500  07728000  07747000  09282600  10243000  10264420  12025000
       12030000  12115000  12126000  13210000  13282000  13332000  13338000  13340000  13386000  13391000  13395000
       13400000  13404000  13651100  13729000  13753000  13754000  13758000  13761000  13763000  13766000  14066000
       14135000  14136000  14136100  14137500  14146000  14153000  14161080  14165000  14168000  14187000  14220000
       14263000  14269820  14286000  14295000  14311000  14326000  14333300  14337000  14339000  14346000  14351000
       14358000  14371000  14442000  14486000  14508000  15376600  16033000  16071000  16163000  16171000  16265000
       16434100
FLIPFLOP  --  BOOLEAN  --  DECLARED IN SEGMENT 97 AT 13750500
      *13752000**13755500*
FOOT  --  INTEGER  --  DECLARED IN SEGMENT 70 AT 08091000
      *08162000* 08166000
FORLIST  --  PROCEDURE  --  DECLARED IN SEGMENT 69 AT 08080000
       08156000  08172000  08231000
FORMAL  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01149000
       05209000  05218000  05236000  07070000  09133800  13202000  13244000  14427000
FORMALF  --  BOOLEAN  --  DECLARED IN SEGMENT 3 AT 01591000
       07400000  07411000 *13202000**13328000**13341100**13343000* 13372000 *16000400*
FORMALNAME  --  DEFINE  --  DECLARED IN SEGMENT 107 AT 15076110
       15092200  15100000
FORMALV  --  DEFINE  --  DECLARED IN SEGMENT 69 AT 08014000
       08056000  08065000  08098000  08108000  08130000  08179000  08212000  08217000  08227000
```

```
FORMATBIT -- DEFINE -- DECLARED IN SEGMENT 3 AT 01000950
        02567000  14461500  14493500
FORMATOG -- DEFINE -- DECLARED IN SEGMENT 3 AT 01001240
        14461500  14493500
FORSTMT -- PROCEDURE -- DECLARED IN SEGMENT 3 AT 08008000 -- FORWARD AT 03052000
        07755000  08232000
FORV -- DEFINE -- DECLARED IN SEGMENT 3 AT 01223000
FORWARDBRANCH -- INTEGER -- DECLARED IN SEGMENT 70 AT 08091000
       *08152000* 08157000  08169000  08171000
FORWARDREF -- DEFINE -- DECLARED IN SEGMENT 3 AT 01007485
        14469100  17092000
FORWARDV -- DEFINE -- DECLARED IN SEGMENT 3 AT 01254000
        14467000
FORWART -- REAL -- VALUE PARAMETER -- DECLARED IN SEGMENT 70 AT 08084000
        08082000  08083000  08086000
FOULED -- REAL -- DECLARED IN SEGMENT 3 AT 01583100
       *04119500**04151000**04207500**07483500**07491500**07604010**14505000*
FOUND -- LABEL -- DECLARED IN SEGMENT 18 AT 02309000 -- OCCURS AT 02315000
        02313000
FOUND -- LABEL -- DECLARED IN SEGMENT 32 AT 04204000 -- OCCURS AT 04221000
        04209000
FOUNDLB -- BOOLEAN -- DECLARED IN SEGMENT 103 AT 14269080
       *14269360**14269440* 14269660
FP -- DEFINE -- DECLARED IN SEGMENT 3 AT 01420000
        06164000  15102400  15305000
FPART -- LABEL -- DECLARED IN SEGMENT 24 AT 02639000 -- OCCURS AT 02790000
        02682000
FR -- DEFINE -- DECLARED IN SEGMENT 3 AT 01420000
        08228000  15301700
FR -- INTEGER -- VALUE PARAMETER -- DECLARED IN SEGMENT 3 AT 03042000
FR -- INTEGER -- VALUE PARAMETER -- DECLARED IN SEGMENT 3 AT 13657000
        13677100 *13677250* 13677300
FRMTID -- DEFINE -- DECLARED IN SEGMENT 3 AT 01182000
FROM -- BOOLEAN -- VALUE PARAMETER -- DECLARED IN SEGMENT 3 AT 03011000
FROM -- BOOLEAN -- VALUE PARAMETER -- DECLARED IN SEGMENT 3 AT 03029000
FROM -- INTEGER -- VALUE PARAMETER -- DECLARED IN SEGMENT 3 AT 03037000
        03036000
FROM -- INTEGER -- 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 06194000
        06198000
FROM -- BOOLEAN -- VALUE PARAMETER -- DECLARED IN SEGMENT 3 AT 07391000
        07402000  07404000
FRONT -- INTEGER -- DECLARED IN SEGMENT 61 AT 07491000
       *07493000* 07496000
FRSTLEVEL -- INTEGER -- DECLARED IN SEGMENT 3 AT 01402000
        05116000 *14478000*
FS -- DEFINE -- DECLARED IN SEGMENT 3 AT 01420000
        07739000  15089600  15098200  15099200  15184200  15186800  15276100  15276110  15276160  15276190  15300800
        15304300  15304400  15306700
FUNCMONFILE -- DEFINE -- DECLARED IN SEGMENT 3 AT 01543000
FUNCTOG -- BOOLEAN -- DECLARED IN SEGMENT 3 AT 01586000
        13736000  14054000  14187000 *14275000**14288000**14296000* 14411100  14415100  14452000  14478000 *14599000*
FUNCTOGO -- BOOLEAN -- DECLARED IN SEGMENT 98 AT 14034000
       *14054000* 14599000
FWDSEQNO -- STREAM VARIABLE -- VALUE PARAMETER -- DECLARED IN SEGMENT 124 AT 17047300
        17047400  17050700  17051000
```

```
FWDSEQNO  --  OWN REAL  --  DECLARED IN SEGMENT 129 AT 17091150
      *17092600**17093400* 17094900
FWDTOG  --  BOOLEAN  --  DECLARED IN SEGMENT 104 AT 14274000
      *14276000**14312000* 14333100  14373000
FWDTOG  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 124 AT 17047200
       17047400  17050600
FWDTOG  --  OWN BOOLEAN  --  DECLARED IN SEGMENT 129 AT 17091110
      *17093300* 17094800 *17095400*
FZERO  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01692000
       05344592  05344662  05375100  14078000
G  --  REAL  --  DECLARED IN SEGMENT 3 AT 01610000
      *13231000* 13236000 *13718000* 13723000 *13724000* 13726000 *13727000* 13729000 *14084000* 14085000  14091000
      *14115000**14116000* 14117000 *14278000**14281000* 14284000  14292000  14294000 *14296000* 14298000  14299000
      *14310000* 14315000 *14349000**14352000* 14353000 *14446000* 14449000 *14470000*
GENERATE  --  LABEL  --  DECLARED IN SEGMENT 116 AT 16313000  --  OCCURS AT 16345000
       16322000
GEQ  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01647000
       08050000
GET  --  INTEGER PROCEDURE  --  DECLARED IN SEGMENT 3 AT 05307000  --  FORWARD AT 03019000
       04102000  04105000  04182000 *05312000* 07604050  07607000  07608000  07646660  08086000  13235000  13258000
       13267000  13273000  14168000  16032000  16092000
GETF  --  INTEGER STREAM PROCEDURE  --  DECLARED IN SEGMENT 2 AT 00523000
       00524000 *00526000* 00527000  00529000
GETLP  --  LABEL  --  DECLARED IN SEGMENT 99 AT 14076000  --  OCCURS AT 14084000
       14089000
GETSPACE  --  INTEGER PROCEDURE  --  DECLARED IN SEGMENT 3 AT 05331000  --  FORWARD AT 03051000
       05132000 *05375000* 05378000  08162000  13347000  14044000
GETSYL  --  INTEGER STREAM PROCEDURE  --  DECLARED IN SEGMENT 42 AT 05309000
       05310000  05312000
GETVOID  --  STREAM PROCEDURE  --  DECLARED IN SEGMENT 3 AT 01756000
       01784000  02410000  02485000
GIT  --  INTEGER PROCEDURE  --  DECLARED IN SEGMENT 3 AT 05123000  --  FORWARD AT 01719000
       02901000 *05124000* 06104300  07410000  07412000  07438000  07441000  07527000  07539000  07601000  13249000
       13257000  14388100  14438000  14483400  15376300  16070000  16089000  16138000  16163000  16254000  16257000
GNAT  --  INTEGER PROCEDURE  --  DECLARED IN SEGMENT 3 AT 05128000  --  FORWARD AT 03020000
      *05131000**05132000* 05133000  05316000  07516300  13267000
GNC  --  STREAM LABEL  --  DECLARED IN SEGMENT 9 AT 02091500  --  OCCURS AT 02104000
       02097000
GOGEN  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 07535000  --  FORWARD AT 03023000
       07518000  07529000  07545000  07572000  07573000  07575000  07583000  13274000
GOMCP  --  LABEL  --  DECLARED IN SEGMENT 62 AT 07504000
GOSTMT  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 07501000
       07531000  07763000
GOT  --  REAL ARRAY  --  NAME PARAMETER  --  DECLARED IN SEGMENT 81 AT 09393060
       09393050  09393130  09393170  09393190
GOTOG  --  BOOLEAN  --  DECLARED IN SEGMENT 66 AT 07646430
      *07646520* 07646540
GOTOS  --  PROCEDURE  --  DECLARED IN SEGMENT 108 AT 16249000
       16270000  16454000  16487000
GOTSCHK  --  LABEL  --  DECLARED IN SEGMENT 98 AT 14016000  --  OCCURS AT 14160000
GOTSTORAGE  --  BOOLEAN  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 13733000
       13731000  13732000
GOTSTORAGE  --  BOOLEAN  --  DECLARED IN SEGMENT 98 AT 14025000
      *14160000*
GOV  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01232000
       07550000  07556000  07646525
GRINCH  --  REAL ARRAY  --  NAME PARAMETER  --  DECLARED IN SEGMENT 81 AT 09393060
```

```
        09393050  09393130  09393170  09393190
GRINCH  --  REAL ARRAY  --  NAME PARAMETER  --  DECLARED IN SEGMENT 81 AT 09393250
        09393240  09393350  09393400  09393430
GS  --  INTEGER  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 05325010
        05325460  05325470  05325480
GS  --  INTEGER  --  DECLARED IN SEGMENT 44 AT 05344000
       *05344592**05344600**05344615**05344620**05344660* 05344661 *05344662**05344700* 05344720 *05344740* 05344750
       *05350000* 05350160  05350200 *05371000**05373000* 05375000  05375100 *05376000* 05376100
GTA1  --  REAL ARRAY  --  DECLARED IN SEGMENT 3 AT 01697000
        13210000 *13351000* 13381000 *13408500* 13718000  13724000  13727000 *14069000**14129000* 14134000  14161000
        14161020  14164000  14259000  14259010 *14259015* 14259080  14278000  14281000  14296000
GTI1  --  INTEGER  --  DECLARED IN SEGMENT 3 AT 01384500
       *02192500**02193000* 02193250
GTR  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01650000
GTR  --  DEFINE  --  DECLARED IN SEGMENT 119 AT 17002000
        17079000
GT1  --  REAL  --  DECLARED IN SEGMENT 3 AT 01384000
       *02203500* 02205000 *02452000* 02456000 *02652000* 02653000  02655000 *02658000* 02659000  02716000 *02719000*
        02720000 *02723000**02823000* 02830000 *02832000* 02833000  02834000  02838000  02839000 *02875000* 02876000
        02877010  02878000  02879000  02881000  02882000  02882100 *02886000* 02894000 *04105000* 04107000  04109000
        04110000 *04120100**04120500* 04121000  04122000 *04508000* 04521000  04523000  04524000 *05116000* 05117000
       *05231000* 05233000 *05235000**05447000* 05449000 *07042000* 07043000  07044000 *07066000* 07067000 *07436000*
       *07539000* 07540000  07544000  07545000 *07600950**07727020* 07727070 *07994000* 07995000 *08058000* 08060000
       *09034100* 09034200  09133800 *09133900**09134000* 09134100  09134200 *09253000* 09253500  09254000  09255000
       *09414040* 09414080  09414090 *09415000* 09415500 *09416000**09417000* 09418000 *10250000* 10251000 *13226000*
        13232000  13240000  13244000  13248000 *13251000**13256000* 13259000 *13273000* 13274000 *13280000* 13282000
       *13371000* 13373000 *14039000* 14041000  14116000 *14333300**14425000* 14427000  14429000  14430000  14437000
        14447000 *14483200* 14483300 *15098000* 15101000  15101600 *15295100* 15296000 *16254000* 16255000  16257000
        16258000  16260000  16265000 *16452000*
GT1  --  INTEGER  --  DECLARED IN SEGMENT 101 AT 14203000
       *14206000**14221000* 14224000
GT1  --  REAL  --  DECLARED IN SEGMENT 113 AT 16157100
       *16159000* 16163000  16164000  16166000  16171000  16173000
GT2  --  REAL  --  DECLARED IN SEGMENT 3 AT 01384000
       *02721000* 02723000 *02834000**02839000**02876000* 02877010  02878000  02879000  02881000  02882000 *05233000*
        05234000 *05448000* 05450000 *07539000* 07545000 *07601000* 07604040  07604050  07604060  07604070 *07604090*
        07605000  07607000  07607100  07608000 *07609000**09133800* 09133900  09134000 *09134100**13284000* 13285000
        13286000 *14430000* 14431000 *14446000* 14448000 *16163000* 16167000  16172000 *16254000* 16255000  16266000
GT2  --  INTEGER  --  DECLARED IN SEGMENT 101 AT 14203000
       *14217000* 14224000
GT3  --  REAL  --  DECLARED IN SEGMENT 3 AT 01384000
       *05402100**05405000**05407000**05409000**05428000**05430000**07397400**07397500**07527000* 07528000 *07601000*
        07610000 *09134100* 09134200 *13285000* 13287000 *14438000**14483400*
GT4  --  REAL  --  DECLARED IN SEGMENT 3 AT 01384000
        02904000 *06104300**06104500**06104510**06104520* 06104600 *07527000**07528000* 07529000 *07601000* 07603000
        07610000 *13235000* 13238000
GT4  --  INTEGER  --  DECLARED IN SEGMENT 101 AT 14203000
GT5  --  REAL  --  DECLARED IN SEGMENT 3 AT 01384000
       *07028000* 07029000 *07604050* 07604080  07604090 *07607000* 07609000
GT5  --  INTEGER  --  DECLARED IN SEGMENT 101 AT 14203000
       *01828000* 01833000  01835000
H  --  STREAM VARIABLE  --  DECLARED IN SEGMENT 38 AT 05068000
        05071100  05071200  05085000
H  --  REAL  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 05315000
        05316000
HEXIZE  --  BOOLEAN STREAM PROCEDURE  --  DECLARED IN SEGMENT 3 AT 02001858
       *02001897* 02001898
HEXOP  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01268000
```

```
        06071000  06089000
HF  --  LABEL  --  DECLARED IN SEGMENT 98 AT 14017000  --  OCCURS AT 14382000
        14123000  14508000
HOLE  --  REAL  --  DECLARED IN SEGMENT 58 AT 07393000
       *07396000* 07398000  07410000  07412000
HTTEOAP  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 13731000
        13747000  14489000  14600000
HVS1  --  STREAM PROCEDURE  --  DECLARED IN SEGMENT 122 AT 17036000
        17042600
HV1  --  PROCEDURE  --  DECLARED IN SEGMENT 122 AT 17042400
        17045000
HV2  --  PROCEDURE  --  DECLARED IN SEGMENT 124 AT 17112000
        17119000
I  --  INTEGER  --  DECLARED IN SEGMENT 3 AT 01319000
       *02917000**05002000**05003000**05107000**05107100**05107200* 05189000  05216000  05217000  05218000  05221000
        05222000  05224000 *05225000* 05273300  05273400  05274100  05274200  05280000  05281000  05344570 *05344600*
        05344615  05344620  05344660 *05344670* 05344700  05344710 *05344770* 05350120  05350160  05350180  06016000
        06044000  06072000  06104300  06108000  06166000  06185000  06315300  06315400  06316300  06316400  06320100
        06320200  06324000  06325000  06326000 *07008000**07031000* 07066000  07070000  07070500  07071000  07396000
        07397210  07401000  07430000  07516000  07516300  07518000  07521000  07554000 *07556000* 07566000  07572000
        07583000  07599000 *07600200**07600500* 07600950  07604020 *07604130**07604150**07646525* 07667000 *07727040*
       *07727060* 07745000  07748000 *07749000**07750000**07751000* 07768100 *07768130**07768170**07768180* 07994000
        07995000  08029000  08034000  08037000 *08110000* 08111000 *08120000**08174000* 08176000 *08223000**09391000*
       *09414030* 09414040 *09414100**09415000* 09418000 *09419000**09420020**09424000* 09426000 *09427000* 09427100
       *09427500**09432000* 09433000  09434000 *09436000**09447020* 09447030  09447040  12015000  12020000  12024000
        12036000  12044000  13200000  13310350 *13324000* 13341300  13361000  13372000  13394000  13753100 *13757000*
        13761100  13767000  13768000  14062000  14063000 *14067000* 14129000 *14137200**14189000* 14220000  14221000
       *14255000**14264000**14269140* 14269446  14269448  14269520  14269540  14314000 *14332000* 14349000  14352000
        14353000 *14360000* 14399000 *14408000* 14516000  14519000  15077000  15091400  15091600  15091800  15184750
        15184800  15184850  15276010 *15276040* 15301900  15302000  15302100  16000700  16121000  16124000  16159000
        16200000  16204000  16208000  16211000  16212500 *16252000* 16254000  16321000  16329000  16342000  16344000
        16370000  16376000  16379000  16408000  16412000  16435000 *16449000**16450000* 16482300 *16489000* 16495100
       *17004700* 17004710
I  --  INTEGER  --  DECLARED IN SEGMENT 33 AT 04237000
       *04239000* 04241000 *04242000* 04243000
I  --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 44 AT 05334100
        05334400  05334600
I  --  REAL  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 05400000
        05402000  05402200  05402300  05404000  05409200  05409500
I  --  REAL  --  DECLARED IN SEGMENT 46 AT 05414000
       *05424000* 05425000  05425100 *05425400* 05426000 *05430000* 05430500  05431000
I  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 46 AT 05415000
        05419000
I  --  REAL  --  DECLARED IN SEGMENT 47 AT 05438000
       *05444000* 05445000  05447000  05455000 *05456000**05458000* 05459000
I  --  INTEGER  --  DECLARED IN SEGMENT 71 AT 08999050
       *08999175* 08999225  08999350
I  --  INTEGER  --  DECLARED IN SEGMENT 72 AT 08999500
       *08999525* 08999550  08999650
I  --  REAL  --  DECLARED IN SEGMENT 83 AT 09393270
       *09393300* 09393310
I  --  DEFINE  --  DECLARED IN SEGMENT 86 AT 10258200
        10277000  10282000
I  --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 96 AT 13673100
        13673300  13673400
I  --  INTEGER  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 119 AT 17002020
        17002015  17002020  17002040
```

```
I  --  INTEGER  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 119 AT 17002055
       17002050  17002055  17002075
I  --  INTEGER  --  DECLARED IN SEGMENT 125 AT 17054400
       *17054600*  17054800  17055010
I  --  DEFINE  --  DECLARED IN SEGMENT 127 AT 17073100
       17079000  17080000
I  --  REAL  --  DECLARED IN SEGMENT 129 AT 17091120
       *17094900*  17095000  17095100  17095300  17095310
ID  --  REAL  --  DECLARED IN SEGMENT 18 AT 02310000
       *02312000*  02313000
ID  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 38 AT 05067000
       05070000
ID  --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 43 AT 05325040
       05325320
IDBIT  --  BOOLEAN  --  DECLARED IN SEGMENT 57 AT 07039000
       *07047000*  07050000  07068000  07082000
IDBLDR  --  STREAM LABEL  --  DECLARED IN SEGMENT 9 AT 02091500  --  OCCURS AT 02110500
       02095000  02103000
IDENT  --  LABEL  --  DECLARED IN SEGMENT 24 AT 02639000  --  OCCURS AT 02865000
       02643000
IDMAX  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01479000
       05344520  06106300  07664000  13337000  13754000  16213000  16253000  16482100  17047100  17095000
IDNOF  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01007410
       02001660  02001795  17025400  17042350  17091900  17093850  17094050  17094100
IDTYPE  --  REAL ARRAY  --  DECLARED IN SEGMENT 124 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
IFFXP  --  INTEGER PROCEDURE  --  DECLARED IN SEGMENT 3 AT 06294000  --  FORWARD AT 03013000
       06005000 *06299000*  06311000
IFS  --  PROCEDURE  --  DECLARED IN SEGMENT 108 AT 16191000
       16239000  16486000
IFSB  --  LABEL  --  DECLARED IN SEGMENT 114 AT 16193000  --  OCCURS AT 16216000
       16194000
IFSC  --  LABEL  --  DECLARED IN SEGMENT 114 AT 16193000  --  OCCURS AT 16207000
       16194000
IFSTMT  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 07561000  --  FORWARD AT 03022000
       07587000  07761000
IFSW  --  SWITCH LABEL  --  DECLARED IN SEGMENT 114 AT 16194000
       16197000
IFTOG  --  LABEL  --  DECLARED IN SEGMENT 114 AT 16193000  --  OCCURS AT 16217000
       16194000  16215000
IFV  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01231000
       06004000
IMPFUN  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 06183000  --  FORWARD AT 03039000
       06158000
INCR  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01153000
       02658000  05124000  05189000  05222000  06080000  06189000  13206000  13240000  13307000  14116000  15255000
       15295100
INCRF  --  REAL  --  DECLARED IN SEGMENT 3 AT 01607000
       *13206000*  13769000  14314000
INDEC  --  LABEL  --  DECLARED IN SEGMENT 98 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 03053000
INDEX  --  INTEGER  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 05005000
        05006000
INDEX  --  REAL  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 05008000
        05009000
INDEX  --  REAL  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 07036000
        07035000  07042000  07098000
INDEX  --  DEFINE  --  DECLARED IN SEGMENT 65 AT 07595000
        07601000  07610000
INDEX  --  INTEGER  --  DECLARED IN SEGMENT 116 AT 16314000
       *16320000**16328000**16334000* 16346000  16349000
INDEXS  --  PROCEDURE  --  DECLARED IN SEGMENT 108 AT 16311000
        16355000  16490000
INFO  --  REAL ARRAY  --  DECLARED IN SEGMENT 3 AT 01007000
        02001805  02212250  02214250  02214260  02226000  02234750  02234900  02265600  02265700  02265850  02265950
        02368000  02410000  02485000  02602800  02877010  02878000  02879000  02881000  02923000  04172000  04187000
        04209000 *04210000**04211000* 04221000 *04223000* 05006000 *05009000* 05039500  05045000  05088000 *05233000*
        05325470  05350160  05350200  05426000  05431000  05449000  05450000  07086400 *08035000* 09034200  09034500
       *09085000**09121650* 09253500  09254000  09255000  09384100  09386000  10251000  10254000  10264800  12117000
        13281000  13310000 *13593000* 13677300  14440000 *17025100**17025300* 17080000
INFO  --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 3 AT 02005000
        02006000
INFO  --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 19 AT 02320200
        02320600
INFO  --  REAL ARRAY  --  DECLARED IN SEGMENT 81 AT 09392500
        09406000  09427500  09447060
INFO  --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 87 AT 10236000
        10238000
INLINE  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 13748000  --  FORWARD AT 03010000
        07746000  13774000
INLINETOG  --  BOOLEAN  --  DECLARED IN SEGMENT 3 AT 01695500
        13307500 *13752000**13770500*
INPUT1  --  BOOLEAN PROCEDURE  --  DECLARED IN SEGMENT 122 AT 17006000
       *17012000* 17045000
INPUT2  --  BOOLEAN PROCEDURE  --  DECLARED IN SEGMENT 124 AT 17070000
       *17082000* 17119000
INT  --  OWN BOOLEAN  --  DECLARED IN SEGMENT 69 AT 08012000
       *08060000* 08070000
INTARRAYID  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01205000
        04522000  08057000  15185200  15304200
INTBIT  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01000960
        02577000  09253050  09253100  09394100  09414010  09430050  13651100
INTEGERDEC  --  LABEL  --  DECLARED IN SEGMENT 98 AT 14013000  --  OCCURS AT 14141000
        14018000
INTID  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01201000
        04509000  08038000  08061000  12035000  12042000  14141000  15085200  15098000
INTNAMEID  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01205400
INTOG  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01001250
        02577000  09253050  09253100  09394100  09414010  09430050  13651100
INTPROCID  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01197000
        13278000  14294000  15078000
INTRINSICADR  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01299500
        09253100
INTRNSICPROCID  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01188000
INTSTRPROCID  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01193000
        14284000
```

```
INTV  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01285000
INV  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01290000
INX  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01652000
        04531000  07091000  15301600  15305600
IOCLASS  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01233000
IOSTMT  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 07993000  --  FORWARD AT 03012000
        07765000
IPART  --  LABEL  --  DECLARED IN SEGMENT 24 AT 02639000  --  OCCURS AT 02758000
        02643000
IS  --  LABEL  --  DECLARED IN SEGMENT 36 AT 04501000  --  OCCURS AT 04548000
        04527000  04540000
ISD  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01653000
        15098400  15276100  15304300
ISKIP  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 87 AT 10236000
        10237000  10238000  15098800  15276100  15304300
ISOLATE  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01270000
        06092000
J  --  REAL  --  DECLARED IN SEGMENT 3 AT 01618000
        *13210000**13324000**13333500**13342000* 13351000 *13381000**13408500**13718000**13724000**13727000**14069000*
        *14129000* 14132000  14134000 *14135000**14136000**14136100**14137600**14161000**14161020**14164000**14278000*
        *14281000**14296000**14421000* 14425000  14438000  14440000  14445100  14446000  14448000 *14449000*
J  --  INTEGER  --  DECLARED IN SEGMENT 31 AT 04166000
        *04170000* 04172000  04187000
J  --  REAL  --  DECLARED IN SEGMENT 45 AT 05401000
        *05405000**05406000* 05407000
J  --  REAL  --  DECLARED IN SEGMENT 46 AT 05414000
        *05428000**05429000* 05430000
J  --  REAL  --  DECLARED IN SEGMENT 47 AT 05438000
        *05444000* 05446000  05448000  05452000 *05453000*
J  --  REAL  --  DECLARED IN SEGMENT 58 AT 07393100
        *07397300* 07397400
J  --  INTEGER  --  DECLARED IN SEGMENT 66 AT 07646410
        *07646630* 07646635 *07646645* 07646650 *07646660*
J  --  INTEGER  --  DECLARED IN SEGMENT 71 AT 08999050
        *08999125* 08999225 *08999275* 08999350
J  --  INTEGER  --  DECLARED IN SEGMENT 72 AT 08999500
        *08999625* 08999650  08999700 *08999750*
J  --  INTEGER  --  DECLARED IN SEGMENT 81 AT 09393010
        *09414022* 09414024 *09414026**09414030**09414060* 09414070  09414110 *09420020**09425000**09428000**09447050*
         09447060  09447080
J  --  REAL  --  DECLARED IN SEGMENT 83 AT 09393270
        *09393290* 09393350 *09393360* 09393400 *09393420* 09393430 *09393450*
J  --  REAL  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 10228000
         10263110  10264600  10264900
J  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 86 AT 10257100
         10257500 *10257700*
J  --  REAL  --  DECLARED IN SEGMENT 89 AT 12102000
        *12116000* 12117000  12126000
J  --  REAL  --  DECLARED IN SEGMENT 91 AT 13221000
        *13232000* 13239000 *13257000**13258000* 13264000 *13265000* 13270000 *13271000*
J  --  INTEGER  --  DECLARED IN SEGMENT 97 AT 13750000
        *13765000*
J  --  REAL  --  DECLARED IN SEGMENT 102 AT 14254100
        *14259010**14259015**14259080* 14259150  14266000
J  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 102 AT 14254200
         14254500
J  --  REAL  --  DECLARED IN SEGMENT 107 AT 15075000
        *15234000* 15255000  15266000 *15280000* 15290000  15296000  15301100  15305200
```

```
J  --  INTEGER  --  DECLARED IN SEGMENT 116 AT 16314000
       *16348000**16349000*
J  --  INTEGER  --  DECLARED IN SEGMENT 117 AT 16364000
       *16387000* 16389000 *16392000* 16393000
J  --  INTEGER  --  DECLARED IN SEGMENT 120 AT 17002030
       *17002035* 17002040
J  --  INTEGER  --  DECLARED IN SEGMENT 121 AT 17002065
       *17002070* 17002075
J  --  INTEGER  --  DECLARED IN SEGMENT 125 AT 17054400
       *17054900* 17055010  17055020
JEDEN  --  INTEGER  --  DECLARED IN SEGMENT 15 AT 02264100
       *02265000* 02265100 *02265350* 02265450  02265600  02265700  02265850  02265950
JFC  --  DEFINE  --  DECLARED IN SEGMENT 108 AT 16016000
       16219000
JFW  --  DEFINE  --  DECLARED IN SEGMENT 108 AT 16007000
       16079000  16232000
JNS  --  DEFINE  --  DECLARED IN SEGMENT 108 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 112 AT 16118000
       *16127000* 16143000
JRV  --  DEFINE  --  DECLARED IN SEGMENT 108 AT 16009000
       16079000
JUMPCHAIN  --  PROCEDURE  --  DECLARED IN SEGMENT 108 AT 16085000
       16097000  16139000  16173000
JUMPCHKNX  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 13616000  --  FORWARD AT 03058000
       14165000  14387000
JUMPCHKX  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 13596000  --  FORWARD AT 03059000
       14506000
JUMPCTR  --  INTEGER  --  DECLARED IN SEGMENT 3 AT 01459000
       14497000 *14499000*
JUMPLEVEL  --  REAL  --  DECLARED IN SEGMENT 3 AT 01485000
       16260000  16265000 *16269000**16433000**16435000* 16452000
JUMPS  --  PROCEDURE  --  DECLARED IN SEGMENT 108 AT 16431000
       16455000  16493000
JUMPV  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01242000
JUNK  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01430000
K  --  REAL  --  DECLARED IN SEGMENT 3 AT 01693000
       *14085000*
K  --  STREAM LABEL  --  DECLARED IN SEGMENT 9 AT 02091500  --  OCCURS AT 02099000
       02099000
K  --  REAL  --  DECLARED IN SEGMENT 45 AT 05401000
       *05405000* 05406000
K  --  REAL  --  DECLARED IN SEGMENT 46 AT 05414000
       *05428000* 05429000
K  --  REAL  --  DECLARED IN SEGMENT 47 AT 05438000
       *05444000* 05452000 *05454000* 05455000 *05457000*
K  --  INTEGER  --  DECLARED IN SEGMENT 66 AT 07646415
       *07646635* 07646640
K  --  DEFINE  --  DECLARED IN SEGMENT 69 AT 08016000
       08033000  08034000  08035000  08094000  08180000  08223000  08232000
K  --  INTEGER  --  DECLARED IN SEGMENT 71 AT 08999050
       *08999200**08999275* 08999350
K  --  INTEGER  --  DECLARED IN SEGMENT 81 AT 09393010
       *09447030* 09447060
K  --  REAL  --  DECLARED IN SEGMENT 86 AT 10258100
```

```
        *10263200* 10263800 *10264200**10264400*
K   --  REAL  --  DECLARED IN SEGMENT 89 AT 12102000
        *12109000* 12117000  12118000  12121000
K   --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 89 AT 12103000
         12104000
K   --  REAL  --  DECLARED IN SEGMENT 102 AT 14254100
        *14259010* 14259080  14259090
K   --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 102 AT 14254200
         14254400
K   --  INTEGER  --  DECLARED IN SEGMENT 117 AT 16364000
        *16389000*
K   --  INTEGER  --  DECLARED IN SEGMENT 125 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
         05350100  05350200 *13201000**13226000* 13246000  13278000 *13333500**13342000* 13372000 *13677200* 13677300
        *16000300*
KNOT --  BOOLEAN  --  DECLARED IN SEGMENT 27 AT 02973500
        *02974500* 02978500
L   --  INTEGER  --  DECLARED IN SEGMENT 3 AT 01428000
         02195000  02195750  02214200  02228750  02421000  04086000  04102000 *04105000**04109000* 04119000 *04120000*
        *04124000**04145000* 04147000  04149000 *04151000**04172000* 04175000  04182000 *04185000**04186000* 04207500
         04210000 *04216000* 04218000  04223000  04279000  04282500  04293000  04296000 *04298000**04300000* 05038000
        *06054100* 06104400  06104500  06104510 *06148000* 06206000 *06207000**06297000**06300000* 06301000  06307000
        *06308000* 07025020  07079000  07081000  07088000 *07411000**07433000**07440000**07443000**07444000**07459000*
         07483500  07485000 *07487000* 07491500 *07493000**07495000* 07496000 *07540000* 07545000 *07577000* 07579000
        *07583000**07585000* 07585100  07586000  07597000 *07600300**07600440* 07600800  07604010  07604040 *07604060*
         07604067 *07604085* 07604090 *07604100* 07607100  07608000  07610000  07646470  07646510  07646560  07646605
         07646625 *07646635* 07646665  07727020 *07727050* 07727070  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* 09282600  09388000 *09395000* 09396000  09397000  09416000 *13232000* 13235000
        *13238000**13239000* 13258000 *13263000**13264000* 13267000 *13270000* 13273000 *13277000* 13606000  13612000
        *13628000* 13746000  13752000 *13765000**14039000* 14041000  14054000 *14055000* 14167000  14168000  14388100
         14451200  14461100  14461200 *14461300**14483700* 14487000  14505000  14602000  14603000  15376400  15376500
         15376600 *16032000**16034000* 16070000  16088000 *16089000* 16090000  16092000 *16094000**16096000* 16121000
         16138000  16167000  16172000  16218000  16232000  16257000
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 9 AT 02091500  --  OCCURS AT 02101000
         02101000
L   --  LABEL  --  DECLARED IN SEGMENT 9 AT 02127500  --  OCCURS AT 02128000
         02135500
L   --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 02183500
         02183750  02186000
L   --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 11 AT 02196530
         02196550
L   --  REAL  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 03020000
L   --  INTEGER  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 03039500
L   --  INTEGER  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 03051001
         03051000
L   --  LABEL  --  DECLARED IN SEGMENT 33 AT 04232000  --  OCCURS AT 04243000
         04241000
L   --  LABEL  --  DECLARED IN SEGMENT 38 AT 05049000  --  OCCURS AT 05092000
```

```
        05094000  05098000
L  --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 38 AT 05050000
        05051000  05052000
L  --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 38 AT 05059000
        05060000
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  --  REAL  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 05411100
L  --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 46 AT 05415000
        05418000
L  --  REAL  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 05437000
        05440000  05441000  05442000  05444000  05458000
L  --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 49 AT 06064000
        06065000
L  --  INTEGER  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 08999450
        08999525
L  --  LABEL  --  DECLARED IN SEGMENT 93 AT 13360000  --  OCCURS AT 13364000
        13366000
L  --  LABEL  --  DECLARED IN SEGMENT 116 AT 16313000  --  OCCURS AT 16347000
        16319000  16327000  16340000
L  --  INTEGER  --  DECLARED IN SEGMENT 117 AT 16364000
       *16391000* 16393000
L  --  LABEL  --  DECLARED IN SEGMENT 108 AT 16475000  --  OCCURS AT 16483000
        16476000
L  --  LABEL  --  DECLARED IN SEGMENT 123 AT 17009000  --  OCCURS AT 17014000
        17011000
L  --  LABEL  --  DECLARED IN SEGMENT 127 AT 17073000  --  OCCURS AT 17085000
        17081000
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
LABELDEC  --  LABEL  --  DECLARED IN SEGMENT 98 AT 14014000  --  OCCURS AT 14187000
        14019000
LABELID  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01206000
        05344520  05350120  05350160  06104100  06106300  06107100  07047000  07075000  07506000  07553000  07645000
        07664000  13246000  13337000  13754000  14194000  14220000  15128100  15376900  16213000  16253000  16482100
        17047100  17095000
LABELR  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 07593000
        07646000  07741000
LABELS  --  PROCEDURE  --  DECLARED IN SEGMENT 108 AT 16156000
        16177000  16485000
LABELV  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01286000
        14399000
LABLMONFILE  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01535000
LAMPER  --  LABEL  --  DECLARED IN SEGMENT 50 AT 06137000  --  OCCURS AT 06179000
        06164000  06166000  06168000
LASS  --  LABEL  --  DECLARED IN SEGMENT 107 AT 15076000  --  OCCURS AT 15276060
        15276160
LASTADDRESS  --  REAL  --  DECLARED IN SEGMENT 124 AT 17069600
       *17079000* 17080000 *17084000* 17091800  17095750 *17096850*
```

```
LASTCRDPATCH  --  BOOLEAN  --  DECLARED IN SEGMENT 3 AT 01001500
LASTENTRY  --  REAL  --  DECLARED IN SEGMENT 3 AT 01376000
        04170000 *04189000* 04208000  04210000  04211000 *04214000* 14038000
LASTGT  --  DEFINE  --  DECLARED IN SEGMENT 106 AT 14420000
        14438000
LASTGT  --  DEFINE  --  DECLARED IN SEGMENT 108 AT 16003000
        16089000  16266000
LASTINFO  --  REAL  --  DECLARED IN SEGMENT 3 AT 01619000
        02724000  02725000 *02726000* 05350200 *09033000* 10261000  10263100  10263110  10264800  13222000 *13290000*
        13309000 *13311000* 13347000  13348000  13408000  13755500  14079000  14083000  14259150  14269260  14320000
       *14330000**14373000* 14421000 *14463000* 16000700  16445000 *16447000*
LASTINFOT  --  REAL  --  DECLARED IN SEGMENT 98 AT 14029000
       *14330000* 14373000  14463000
LASTRESULT  --  INTEGER  --  DECLARED IN SEGMENT 86 AT 10258000
       *10262000* 10272000 *10276000*
LASTSEQROW  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01570000
        02214250  02214260  02226000  02234750  02234900  02368000  02410000  02485000  02602800  05039500  05045000
        07086400  09033000  09034200  09034500  09253500  09254000  09255000  09384100  09386000
LASTSEQUENCE  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01569000
        02214250  02214260  02226000  02234750  02234900  02368000  02410000  02485000  02602800  05039500  05045000
        07086400  09033000  09034200  09034500  09253500  09254000  09255000  09384100  09386000
LASTUSED  --  INTEGER  --  DECLARED IN SEGMENT 3 AT 01391000
        02194500 *02204000**02209250* 02212000  02212250 *02212750* 02228050  02234000 *02332000**02430000**02438000*
       *02458000**02722000* 02900000 *02901000**07030000**09029000**09037000* 09282000
LBC  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01645000
LBLREF  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01007486
        07598100  16159100  17092300
LBLTOG  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 124 AT 17047200
        17047400  17050900
LBLTOG  --  OWN BOOLEAN  --  DECLARED IN SEGMENT 129 AT 17091110
       *17092500* 17094800 *17095400*
LBP  --  DEFINE  --  DECLARED IN SEGMENT 98 AT 14023000
LCARD  --  LABEL  --  DECLARED IN SEGMENT 10 AT 02191000  --  OCCURS AT 02194750
        02191250
LCR  --  INTEGER  --  DECLARED IN SEGMENT 3 AT 01330000
        02129500  02193250 *02202000**02204000**02209000* 02213000 *02214000* 02214250  02214500 *02215750* 02225250
        02226000  02228150  02232000  02233500  02234750  02235750  02368000  02369000  02410000  02485000  02602800
        02716000  02717000 *02719000* 02902000 *02903000* 02904000  02905000 *07025010**07025030*
LCR  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 01756000
        01767000  01774000
LCR  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 02005000
        02006000
LCR  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 19 AT 02320200
        02320600
LDES  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01299030
        05248000  05250000  09415500  13252500
LDOT  --  LABEL  --  DECLARED IN SEGMENT 50 AT 06137000  --  OCCURS AT 06178000
        06144000  06156000  06158000  06160000  06162000  06170000  06172500
LEFTPAREN  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01212000
        02975000  06069000  06415000  07409000  07437000  07768110  10264100  12007000  12114000  13753000  14137000
        14259020  14326000  16125000  16482400
LENGTH1  --  LABEL  --  DECLARED IN SEGMENT 19 AT 02360000  --  OCCURS AT 02374000
        02364000
LENGTH2  --  LABEL  --  DECLARED IN SEGMENT 19 AT 02360000  --  OCCURS AT 02380000
LENGTH3  --  LABEL  --  DECLARED IN SEGMENT 19 AT 02360000  --  OCCURS AT 02381000
        02364000
LENGTH4  --  LABEL  --  DECLARED IN SEGMENT 19 AT 02360000  --  OCCURS AT 02399000
```

```
        02364000
LENGTH5  --  LABEL  --  DECLARED IN SEGMENT 19 AT 02361000  --  OCCURS AT 02476000
        02364000
LENGTH6  --  LABEL  --  DECLARED IN SEGMENT 19 AT 02361000  --  OCCURS AT 02518000
        02365000
LENGTH7  --  LABEL  --  DECLARED IN SEGMENT 19 AT 02361000  --  OCCURS AT 02569000
        02365000
LENGTH8  --  LABEL  --  DECLARED IN SEGMENT 19 AT 02361000  --  OCCURS AT 02574000
LENGTH9  --  LABEL  --  DECLARED IN SEGMENT 19 AT 02361000  --  OCCURS AT 02575000
        02365000
LEQ  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01655000
        08050000
LEQ  --  DEFINE  --  DECLARED IN SEGMENT 119 AT 17002000
LEVEL  --  INTEGER  --  DECLARED IN SEGMENT 3 AT 01402000
        05208000  05344400  05350100  05350120  13339000  13371000  13677100 *14053000* 14220000  14307000  14318500
       *14323000* 14387000  14389000  14411300  14415300  14451000  14461100  14477000  14478000  14497000  14499000
       *14500000* 14603000 *14604000*
LEVEL  --  DEFINE  --  DECLARED IN SEGMENT 108 AT 16005000
        16171000  16265000
LEVELF  --  REAL  --  DECLARED IN SEGMENT 3 AT 01605000
       *13204000* 13339000  13371000  14307000
LFC  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01648000
LFTBRKET  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01263000
        05273800  06139000  06316000  06421000  10264100  12114000  13391000  14259020  14269320  15183100
LFTPAREN  --  DEFINE  --  DECLARED IN SEGMENT 108 AT 16003000
        16125000  16482400  14167000
LGTFLD  --  DEFINE  --  DECLARED IN SEGMENT 108 AT 16004000
        16172000
LIB  --  BOOLEAN  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 12 AT 02202500
LIN  --  REAL ARRAY  --  DECLARED IN SEGMENT 3 AT 01559010
        02194000  02195000  02195750  02196580  02196590  02214200  02226750  02228750  02421000  04149000  04150000
        05038000  05039500  05039600  05044000  05046000  05096000  05097500  05099000  05325450  05325460  05325470
        05325480  05325490  06122000  06123000  07025020  07086300  07086500  09384100  09384500  09386000  09387000
        09388000  09389500  13653000  13653500  13675000  13676000
LIN  --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 43 AT 05325040
        05325090
LINE  --  FILE  --  DECLARED IN SEGMENT 3 AT 01558000
        01829000  01835700  02195000  02195750  02196590  02214200  02228750  02264950  02265150  02265300  02265450
        02265650  02265750  02265900  02266000  02421000  02464000  04150000  04279000  04282500  05039600  05046000
        05087000  05088000  05089000  05090000  05097500  05099000  05325490  05427000  05432000  05434000  06123000
        07025020  07086500  09384500  09387000  09389500  09390000  13653500  13676000  14461500  14493500  17002520
        17054500  17054800  17055020  17091510  17094300  17094400  17094550  17095500  17096300
LINE  --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 3 AT 02183500
        02184250
LINE  --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 37 AT 05016000
        05019000  05021000  05024000
LINE  --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 38 AT 05067000
        05069000  05071200  05076000  05085000
LINE  --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 57 AT 07038100
        07038200  07038300
LINE  --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 80 AT 09378000
        09379000  09380000
LINECOUNT  --  DEFINE  --  DECLARED IN SEGMENT 129 AT 17091140
        17091520  17094410  17095510  17096350
LINK  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01154000
        02725000  02882200  05124000  05132000  05189000  05233000  05234000  05447000  05448000  06079000  06189000
        06190000  09133800  09134000  13207000  13232000  13286000  13308000  13361000  13362000  13364000  14448000
```

```
        14470000  15080000  16443000
LINK  -- INTEGER  -- DECLARED IN SEGMENT 31 AT 04166000
        *04182000* 04185000
LINK  -- DEFINE  -- DECLARED IN SEGMENT 65 AT 07595000
        07601000  07604040  07604050  07604060  07604070  07604090  07605000  07607000  07607100  07608000  07609000
LINK  -- INTEGER  -- DECLARED IN SEGMENT 66 AT 07646355
        07646550 *07646560* 07646645 *07646650* 07646660  07646665
LINK  -- REAL  -- DECLARED IN SEGMENT 111 AT 16087000
        *16092000* 16094000
LINKC  -- DEFINE  -- DECLARED IN SEGMENT 3 AT 01156000
        02001660  02001695  02001795  02001805  02001815  02001822  02212250  02876000  05006000  05009000  05325470
        05350180  05350200  05402100  05405000  05407000  05409000  05426000  05428000  05430000  05431000  05449000
        05450000  07397400  07397500  09393110  09393330  09414070  09418000  09419000  09433000  09447060  10251000
        10264800  12117000  13281000  13310000  13310200  13310350  13310950  13593000  13677300  14483400  17025100
        17025300  17080000
LINKF  -- REAL  -- DECLARED IN SEGMENT 3 AT 01606000
        *13207000* 13768000  13769000  14310000  14314000  14315000
LINKP  -- REAL  -- DECLARED IN SEGMENT 93 AT 13359000
        *13361000* 13362000  13364000
LINKR  -- DEFINE  -- DECLARED IN SEGMENT 3 AT 01155000
        02001660  02001695  02001795  02001805  02001815  02001822  02212250  02875000  05006000  05009000  05325470
        05350160  05350200  05402100  05405000  05407000  05409000  05426000  05428000  05430000  05431000  05449000
        05450000  07397400  07397500  09414070  09418000  09419000  09433000  09447060  10251000  10254000  10264800
        12117000  13281000  13302000  13303000  13310000  13310200  13310350  13310950  13593000  13677300  14483400
LINKR  -- DEFINE  -- DECLARED IN SEGMENT 82 AT 09393090
        09393130  09393170  09393190
LINKR  -- DEFINE  -- DECLARED IN SEGMENT 83 AT 09393280
        09393350  09393400  09393430
LINKT  -- REAL  -- DECLARED IN SEGMENT 93 AT 13359000
        *13361000* 13362000  13364000  13365000 *13366000*
LINKTOG  -- BOOLEAN  -- DECLARED IN SEGMENT 3 AT 01399000
        04100000 *04213000**04222000**04295000* 06021000 *06022000*
LISTABIT  -- DEFINE  -- DECLARED IN SEGMENT 3 AT 01000970
        02194250  02511000  02602500
LISTATOG  -- DEFINE  -- DECLARED IN SEGMENT 3 AT 01001260
        02194250  02602500
LISTBIT  -- DEFINE  -- DECLARED IN SEGMENT 3 AT 01000980
        02402000  02602500  09028000
LISTER  -- BOOLEAN  -- DECLARED IN SEGMENT 3 AT 01001570
        02195000  02195750  02214200  02464000 *02602500* 02910600  05036000  07025020  07086100 *09028000* 13652000
        13674000  14461500  14493500
LISTID  -- DEFINE  -- DECLARED IN SEGMENT 3 AT 01181000
LISTOG  -- DEFINE  -- DECLARED IN SEGMENT 3 AT 01001270
        02602500  09028000
LISTPBIT  -- DEFINE  -- DECLARED IN SEGMENT 3 AT 01000990
        02195000  02228750  02480000
LISTPTOG  -- DEFINE  -- DECLARED IN SEGMENT 3 AT 01001280
        02195000  02228750
LITERAL  -- INTEGER  -- VALUE PARAMETER  -- DECLARED IN SEGMENT 3 AT 04003000
        04004000
LITNO  -- DEFINE  -- DECLARED IN SEGMENT 3 AT 01210000
        02850000  05275000  05277000  05344610  06085000  06094000  06317300  06319000  06321000  07600300  07665000
        07666500  07727050  07994000  13392000  14269460  14269500  15276010  16119000  16204000  16212500  16326000
        16341000  16374000  16410000  16435000  16482300
LITV  -- DEFINE  -- DECLARED IN SEGMENT 3 AT 01250000
        16372000  16381000
LN  -- INTEGER  -- DECLARED IN SEGMENT 97 AT 13750000
       *13755500* 13773000
```

```
LND  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01656000
        04545000
LNG  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01657000
        04112000
LO  --  REAL  --  DECLARED IN SEGMENT 98 AT 14029000
       *14304000* 14496000
LOC  --  DEFINE  --  DECLARED IN SEGMENT 106 AT 14420000
        14438000
LOc  --  DEFINE  --  DECLARED IN SEGMENT 108 AT 16003000
        16070000  16163000  16255000
LOCAL  --  BOOLEAN PROCEDURE  --  DECLARED IN SEGMENT 3 AT 05206000
       *05210000* 07516000  07554000  07599000
LOCALV  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01249000
        14396000
LOcBEGIN  --  DEFINE  --  DECLARED IN SEGMENT 98 AT 14022000
LOCFLD  --  DEFINE  --  DECLARED IN SEGMENT 108 AT 16004000
        16138000
LOCLID  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01179000
        13755000  14409000  14411200  14415200  14425000  16198000  16317000  16326000  16330000  16335000  16340000
        16368000  16406000  16482100  16482300  16482500  17095310
LOCV  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01247000
        16336000  16339000
LOD  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01658000
        04523000  05201000  05227000  05316000  06172500  07071000  07090000  08079000  15185300  15267000  15302600
        15305800
LOLD  --  INTEGER  --  DECLARED IN SEGMENT 98 AT 14030000
       *14054000*
LONGID  --  DEFINE  --  DECLARED IN SEGMENT 107 AT 15076120
        15301500  15305300
LOR  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01659000
LOWER  --  REAL  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 05299000
        05297000  05298000  05304000
LPRT  --  REAL  --  DECLARED IN SEGMENT 3 AT 01483000
LS  --  INTEGER  --  DECLARED IN SEGMENT 97 AT 13750000
       *13752000* 13765000
LSQ  --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 80 AT 09364000
        09366000
LSS  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01661000
LSS  --  DEFINE  --  DECLARED IN SEGMENT 119 AT 17002000
        17117400
LSTRTN  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01430000
LSTSEQ  --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 37 AT 05016000
        05020000  05023000
LTAPE  --  LABEL  --  DECLARED IN SEGMENT 10 AT 02191000  --  OCCURS AT 02195500
        02191250
LVL  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01151000
        05116000  05208000  07432000  13204000  14220000  16171000  16265000
L1  --  LABEL  --  DECLARED IN SEGMENT 38 AT 05049000  --  OCCURS AT 05096000
        05099100
L1  --  LABEL  --  DECLARED IN SEGMENT 44 AT 05334000
L1  --  LABEL  --  DECLARED IN SEGMENT 50 AT 06128000  --  OCCURS AT 06152000
        06133000
L1  --  LABEL  --  DECLARED IN SEGMENT 67 AT 07658000  --  OCCURS AT 07675000
        07667500  07672500  07673000
L1  --  LABEL  --  DECLARED IN SEGMENT 68 AT 07713000  --  OCCURS AT 07729000
        07720000
L1  --  LABEL  --  DECLARED IN SEGMENT 73 AT 09025000
```

```
L1  --  LABEL  --  DECLARED IN SEGMENT 88 AT 12005000  --  OCCURS AT 12009000
        12057000
L1  --  STREAM LABEL  --  DECLARED IN SEGMENT 95 AT 13635000  --  OCCURS AT 13641000
        13639000
L1  --  LABEL  --  DECLARED IN SEGMENT 97 AT 13751000  --  OCCURS AT 13754000
        13761100  13762000
L1  --  LABEL  --  DECLARED IN SEGMENT 116 AT 16313000  --  OCCURS AT 16349000
        16353000
L1  --  LABEL  --  DECLARED IN SEGMENT 117 AT 16365000  --  OCCURS AT 16397000
        16372000
L1  --  LABEL  --  DECLARED IN SEGMENT 108 AT 16475000  --  OCCURS AT 16484000
        16482400
L10  --  LABEL  --  DECLARED IN SEGMENT 50 AT 06128000  --  OCCURS AT 06152000
        06133000
L10  --  LABEL  --  DECLARED IN SEGMENT 68 AT 07713000  --  OCCURS AT 07734000
        07720000
L10  --  LABEL  --  DECLARED IN SEGMENT 108 AT 16475000  --  OCCURS AT 16493000
        16476000
L11  --  LABEL  --  DECLARED IN SEGMENT 50 AT 06129000  --  OCCURS AT 06157000
        06134000
L11  --  LABEL  --  DECLARED IN SEGMENT 68 AT 07714000  --  OCCURS AT 07729000
        07721000
L12  --  LABEL  --  DECLARED IN SEGMENT 50 AT 06129000  --  OCCURS AT 06152000
        06134000
L12  --  LABEL  --  DECLARED IN SEGMENT 68 AT 07714000  --  OCCURS AT 07736000
        07721000
L13  --  LABEL  --  DECLARED IN SEGMENT 50 AT 06129000  --  OCCURS AT 06152000
        06134000
L13  --  LABEL  --  DECLARED IN SEGMENT 68 AT 07714000  --  OCCURS AT 07729000
        07721000
L14  --  LABEL  --  DECLARED IN SEGMENT 50 AT 06129000  --  OCCURS AT 06159000
        06134000
L14  --  LABEL  --  DECLARED IN SEGMENT 68 AT 07714000  --  OCCURS AT 07729000
        07721000
L15  --  LABEL  --  DECLARED IN SEGMENT 50 AT 06129000  --  OCCURS AT 06159000
        06134000
L15  --  LABEL  --  DECLARED IN SEGMENT 68 AT 07714000  --  OCCURS AT 07729000
        07721000
L16  --  LABEL  --  DECLARED IN SEGMENT 50 AT 06129000  --  OCCURS AT 06152000
        06134000
L16  --  LABEL  --  DECLARED IN SEGMENT 68 AT 07714000  --  OCCURS AT 07729000
        07721000
L17  --  LABEL  --  DECLARED IN SEGMENT 50 AT 06129000  --  OCCURS AT 06152000
        06134000
L17  --  LABEL  --  DECLARED IN SEGMENT 68 AT 07714000  --  OCCURS AT 07729000
        07721000
L18  --  LABEL  --  DECLARED IN SEGMENT 50 AT 06129000  --  OCCURS AT 06161000
        06134000
L18  --  LABEL  --  DECLARED IN SEGMENT 68 AT 07714000  --  OCCURS AT 07734000
        07721000
L19  --  LABEL  --  DECLARED IN SEGMENT 50 AT 06129000  --  OCCURS AT 06161000
        06134000
L19  --  LABEL  --  DECLARED IN SEGMENT 68 AT 07714000  --  OCCURS AT 07734000
        07721000
L2  --  LABEL  --  DECLARED IN SEGMENT 44 AT 05334000  --  OCCURS AT 05373000
L2  --  LABEL  --  DECLARED IN SEGMENT 50 AT 06128000  --  OCCURS AT 06152000
        06133000
L2  --  LABEL  --  DECLARED IN SEGMENT 68 AT 07713000  --  OCCURS AT 07729000
```

07720000
L2 -- LABEL -- DECLARED IN SEGMENT 88 AT 12005000 -- OCCURS AT 12055000
12018000 12048000
L2 -- LABEL -- DECLARED IN SEGMENT 97 AT 13751000 -- OCCURS AT 13760000
13753100 13754000
L2 -- LABEL -- DECLARED IN SEGMENT 108 AT 16475000 -- OCCURS AT 16485000
16481000 16482200
L20 -- LABEL -- DECLARED IN SEGMENT 50 AT 06129000 -- OCCURS AT 06152000
06134000
L20 -- LABEL -- DECLARED IN SEGMENT 68 AT 07714000 -- OCCURS AT 07729000
07721000
L21 -- LABEL -- DECLARED IN SEGMENT 50 AT 06130000 -- OCCURS AT 06152000
06135000
L21 -- LABEL -- DECLARED IN SEGMENT 68 AT 07715000 -- OCCURS AT 07729000
07722000
L22 -- LABEL -- DECLARED IN SEGMENT 50 AT 06130000 -- OCCURS AT 06163000
06135000
L22 -- LABEL -- DECLARED IN SEGMENT 68 AT 07715000 -- OCCURS AT 07738000
07722000
L23 -- LABEL -- DECLARED IN SEGMENT 50 AT 06130000 -- OCCURS AT 06163000
06135000
L23 -- LABEL -- DECLARED IN SEGMENT 68 AT 07715000 -- OCCURS AT 07738000
07722000
L24 -- LABEL -- DECLARED IN SEGMENT 50 AT 06130000 -- OCCURS AT 06152000
06135000
L24 -- LABEL -- DECLARED IN SEGMENT 68 AT 07715000 -- OCCURS AT 07729000
07722000
L25 -- LABEL -- DECLARED IN SEGMENT 50 AT 06130000 -- OCCURS AT 06152000
06135000
L25 -- LABEL -- DECLARED IN SEGMENT 68 AT 07715000 -- OCCURS AT 07729000
07722000
L26 -- LABEL -- DECLARED IN SEGMENT 50 AT 06130000 -- OCCURS AT 06163000
06135000
L26 -- LABEL -- DECLARED IN SEGMENT 68 AT 07715000 -- OCCURS AT 07738000
07722000
L27 -- LABEL -- DECLARED IN SEGMENT 50 AT 06130000 -- OCCURS AT 06163000
06135000
L27 -- LABEL -- DECLARED IN SEGMENT 68 AT 07715000 -- OCCURS AT 07738000
07722000
L28 -- LABEL -- DECLARED IN SEGMENT 50 AT 06130000 -- OCCURS AT 06152000
06135000
L28 -- LABEL -- DECLARED IN SEGMENT 68 AT 07715000 -- OCCURS AT 07729000
07722000
L29 -- LABEL -- DECLARED IN SEGMENT 50 AT 06130000 -- OCCURS AT 06152000
06135000
L29 -- LABEL -- DECLARED IN SEGMENT 68 AT 07715000 -- OCCURS AT 07729000
07722000
L3 -- LABEL -- DECLARED IN SEGMENT 50 AT 06128000 -- OCCURS AT 06152000
06133000
L3 -- LABEL -- DECLARED IN SEGMENT 68 AT 07713000 -- OCCURS AT 07729000
07720000
L3 -- LABEL -- DECLARED IN SEGMENT 88 AT 12005000 -- OCCURS AT 12056000
12026000
L3 -- LABEL -- DECLARED IN SEGMENT 97 AT 13751000 -- OCCURS AT 13767000
13770000
L3 -- LABEL -- DECLARED IN SEGMENT 108 AT 16475000 -- OCCURS AT 16486000
16476000

```
L30  --  LABEL  --  DECLARED IN SEGMENT 50 AT 06130000  --  OCCURS AT 06163000
         06135000
L30  --  LABEL  --  DECLARED IN SEGMENT 68 AT 07715000  --  OCCURS AT 07738000
         07722000
L31  --  LABEL  --  DECLARED IN SEGMENT 50 AT 06131000  --  OCCURS AT 06163000
         06136000
L31  --  LABEL  --  DECLARED IN SEGMENT 68 AT 07716000  --  OCCURS AT 07738000
         07723000
L32  --  LABEL  --  DECLARED IN SEGMENT 50 AT 06131000  --  OCCURS AT 06153000
         06136000
L32  --  LABEL  --  DECLARED IN SEGMENT 68 AT 07716000  --  OCCURS AT 07740000
         07723000
L33  --  LABEL  --  DECLARED IN SEGMENT 50 AT 06131000  --  OCCURS AT 06165000
         06136000
L33  --  LABEL  --  DECLARED IN SEGMENT 68 AT 07716000  --  OCCURS AT 07730000
         07723000
L34  --  LABEL  --  DECLARED IN SEGMENT 50 AT 06131000  --  OCCURS AT 06167000
         06136000
L34  --  LABEL  --  DECLARED IN SEGMENT 68 AT 07716000  --  OCCURS AT 07730000
         07723000
L35  --  LABEL  --  DECLARED IN SEGMENT 50 AT 06131000  --  OCCURS AT 06165000
         06136000
L35  --  LABEL  --  DECLARED IN SEGMENT 68 AT 07716000  --  OCCURS AT 07730000
         07723000
L36  --  LABEL  --  DECLARED IN SEGMENT 50 AT 06131000  --  OCCURS AT 06167000
         06136000
L36  --  LABEL  --  DECLARED IN SEGMENT 68 AT 07716000  --  OCCURS AT 07730000
         07723000
L37  --  LABEL  --  DECLARED IN SEGMENT 50 AT 06131000  --  OCCURS AT 06173000
         06136000
L37  --  LABEL  --  DECLARED IN SEGMENT 68 AT 07716000  --  OCCURS AT 07730000
         07723000
L38  --  LABEL  --  DECLARED IN SEGMENT 50 AT 06131000  --  OCCURS AT 06169000
         06136000
L38  --  LABEL  --  DECLARED IN SEGMENT 68 AT 07716000  --  OCCURS AT 07742000
         07723000
L39  --  LABEL  --  DECLARED IN SEGMENT 50 AT 06131000  --  OCCURS AT 06171000
         06136000
L39  --  LABEL  --  DECLARED IN SEGMENT 68 AT 07716000  --  OCCURS AT 07730000
         07723000
L4   --  LABEL  --  DECLARED IN SEGMENT 50 AT 06128000  --  OCCURS AT 06152000
         06133000
L4   --  LABEL  --  DECLARED IN SEGMENT 68 AT 07713000  --  OCCURS AT 07729000
         07720000
L4   --  LABEL  --  DECLARED IN SEGMENT 108 AT 16475000  --  OCCURS AT 16487000
         16476000
L40  --  LABEL  --  DECLARED IN SEGMENT 68 AT 07716000  --  OCCURS AT 07744000
         07723000
L41  --  LABEL  --  DECLARED IN SEGMENT 68 AT 07717000  --  OCCURS AT 07770000
         07724000
L42  --  LABEL  --  DECLARED IN SEGMENT 68 AT 07717000  --  OCCURS AT 07752000
         07724000
L43  --  LABEL  --  DECLARED IN SEGMENT 68 AT 07717000  --  OCCURS AT 07754000
         07724000
L44  --  LABEL  --  DECLARED IN SEGMENT 68 AT 07717000  --  OCCURS AT 07756000
         07724000
L45  --  LABEL  --  DECLARED IN SEGMENT 68 AT 07717000  --  OCCURS AT 07758000
```

07724000
L46 -- LABEL -- DECLARED IN SEGMENT 68 AT 07717000 -- OCCURS AT 07769000
07724000
L47 -- LABEL -- DECLARED IN SEGMENT 68 AT 07717000 -- OCCURS AT 07769000
07724000
L48 -- LABEL -- DECLARED IN SEGMENT 68 AT 07717000 -- OCCURS AT 07769000
07724000
L49 -- LABEL -- DECLARED IN SEGMENT 68 AT 07717000 -- OCCURS AT 07759100
07724000
L5 -- LABEL -- DECLARED IN SEGMENT 50 AT 06128000 -- OCCURS AT 06152000
06133000
L5 -- LABEL -- DECLARED IN SEGMENT 68 AT 07713000 -- OCCURS AT 07729000
07720000
L5 -- LABEL -- DECLARED IN SEGMENT 108 AT 16475000 -- OCCURS AT 16488000
16476000
L50 -- LABEL -- DECLARED IN SEGMENT 68 AT 07717000 -- OCCURS AT 07769000
07724000
L51 -- LABEL -- DECLARED IN SEGMENT 68 AT 07718000 -- OCCURS AT 07760000
07725000
L52 -- LABEL -- DECLARED IN SEGMENT 68 AT 07718000 -- OCCURS AT 07762000
07725000
L53 -- LABEL -- DECLARED IN SEGMENT 68 AT 07718000 -- OCCURS AT 07764000
07725000
L54 -- LABEL -- DECLARED IN SEGMENT 68 AT 07718000 -- OCCURS AT 07766000
07725000
L6 -- LABEL -- DECLARED IN SEGMENT 50 AT 06128000 -- OCCURS AT 06152000
06133000
L6 -- LABEL -- DECLARED IN SEGMENT 68 AT 07713000 -- OCCURS AT 07729000
07720000
L6 -- LABEL -- DECLARED IN SEGMENT 108 AT 16475000 -- OCCURS AT 16489000
16476000
L7 -- LABEL -- DECLARED IN SEGMENT 50 AT 06128000 -- OCCURS AT 06155000
06133000
L7 -- LABEL -- DECLARED IN SEGMENT 68 AT 07713000 -- OCCURS AT 07732000
07720000
L7 -- LABEL -- DECLARED IN SEGMENT 108 AT 16475000 -- OCCURS AT 16490000
16476000 16482500
L8 -- LABEL -- DECLARED IN SEGMENT 50 AT 06128000 -- OCCURS AT 06152000
06133000
L8 -- LABEL -- DECLARED IN SEGMENT 68 AT 07713000 -- OCCURS AT 07732000
07720000
L8 -- LABEL -- DECLARED IN SEGMENT 108 AT 16475000 -- OCCURS AT 16491000
16476000
L9 -- LABEL -- DECLARED IN SEGMENT 50 AT 06128000 -- OCCURS AT 06152000
06133000
L9 -- LABEL -- DECLARED IN SEGMENT 68 AT 07713000 -- OCCURS AT 07729000
07720000
L9 -- LABEL -- DECLARED IN SEGMENT 108 AT 16475000 -- OCCURS AT 16492000
16476000
M -- BOOLEAN -- DECLARED IN SEGMENT 44 AT 05343000
05350100 05350140
M -- REAL -- DECLARED IN SEGMENT 45 AT 05401000
*05404000**05407000**05409000* 05409300 05409400
M -- REAL -- DECLARED IN SEGMENT 47 AT 05438000
05441000 05442000 05443000 05444000 05445000
M -- INTEGER -- DECLARED IN SEGMENT 81 AT 09393010
09447040 09447050

```
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  13307500 *14265900**14266100*
MAKEUPACCUM  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 13368000  --  FORWARD AT 01627000
       13333500  13342000  16000600
MARK  --  REAL  --  DECLARED IN SEGMENT 3 AT 01615000
       14117000 *14314000**14319000* 14333300  14334000  14374000  14430000  14451200  14463000
MATCH  --  DEFINE  --  DECLARED IN SEGMENT 129 AT 17091115
       17091800
MAXINTRINSIC  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01299400
       09253100  09414120
MAXROW  --  INTEGER  --  DECLARED IN SEGMENT 3 AT 01684000
MAXSAVE  --  INTEGER  --  DECLARED IN SEGMENT 3 AT 01598000
MAXSTACK  --  REAL  --  DECLARED IN SEGMENT 3 AT 01683000
       13230000 *13231000* 14277000 *14502000*
MAXSTACK0  --  REAL  --  DECLARED IN SEGMENT 98 AT 14029000
      *14277000* 14502000
MAXTLCR  --  INTEGER  --  DECLARED IN SEGMENT 3 AT 01331000
       02439000 *02456000*
MCPBIT  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01001000
       02397000
MCPTOG  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01001290
MCPTYPE  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01561410
       09430060
MDESC  --  STREAM PROCEDURE  --  DECLARED IN SEGMENT 73 AT 09005000
       09034200  09253500  09254000  09255000  09418000  09419100
MEDIUM  --  ALPHA  --  DECLARED IN SEGMENT 3 AT 01001580
       02195000  02195750 *02203250**02204250**02205500**02214100* 02214200  02228750  02421000  05038000  07025020
MERGEBIT  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01001010
       01835600  02425000  02425500  02426000  02427000  02428000  02435000  02436000
MERGEOPTION  --  LABEL  --  DECLARED IN SEGMENT 19 AT 02363000  --  OCCURS AT 02437000
       02428000
MERGETOG  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01001300
       01835600  02425500  02427000  02428000  02436000
MIN  --  INTEGER  --  DECLARED IN SEGMENT 5 AT 01822000
      *01828000* 01834000
MKABS  --  REAL STREAM PROCEDURE  --  DECLARED IN SEGMENT 3 AT 02022000
      *02023000* 02202000  02202050  02214000  02215750  02216500  02217750  02226000  02410500  02411000  02456000
       02485500  02486000  02903000  05091000  07025010  07025030  08999175  08999200  09400000  09401000
MKS  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01662000
       07408000  07434000  13761000  13765000
MKST  --  BOOLEAN  --  DECLARED IN SEGMENT 97 AT 13750000
      *13752000**13761000* 13764000
MODE  --  INTEGER  --  DECLARED IN SEGMENT 3 AT 01402000
       04175500  04184000  04212100  04223100  05115000  05345000  07397200  07600410  07748000  08093000 *14303000*
       14478000 *14501000*
MON  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01147000
       16254000
MONITORDEC  --  LABEL  --  DECLARED IN SEGMENT 98 AT 14014000  --  OCCURS AT 14142000
       14019000
MONITORV  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01291000
MOVE  --  STREAM PROCEDURE  --  DECLARED IN SEGMENT 3 AT 02054000
       02212250  02386100  02412000  02487000  02602600  02916000  04147000  05039500  07029000  07669000  08999550
       08999575  08999700  09393130  09393170  09393190  09393350  09393400  09393430  09433000  09447060  13281000
       13310000  13651000  13746000  14259000  14440000  16495100
```

```
MOVE  --  STREAM PROCEDURE  --  DECLARED IN SEGMENT 71 AT 08999075
        08999100  08999275  08999350
MOVECHARACTERS  --  STREAM PROCEDURE  --  DECLARED IN SEGMENT 3 AT 02045000
        02705000  07668500  16349000  16392000
MOVECODE  --  STREAM PROCEDURE  --  DECLARED IN SEGMENT 3 AT 13683000
MOVEIT  --  LABEL  --  DECLARED IN SEGMENT 24 AT 02638000  --  OCCURS AT 02704000
MOVEXREFINFO  --  STREAM PROCEDURE  --  DECLARED IN SEGMENT 8 AT 02001760
        02001785  02001805
MRCLEAN  --  BOOLEAN  --  DECLARED IN SEGMENT 3 AT 01379000
        04164000 *09040000* 08049000
MULOP  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01278500
        12022000  13337000  15091800  15184850  15302100
MYCLASS  --  INTEGER  --  DECLARED IN SEGMENT 3 AT 01001590
       *02968000* 02971000 *02974500* 02975000  02976500  02977500 *02978500* 02982000  02983000
M0  --  STREAM LABEL  --  DECLARED IN SEGMENT 43 AT 05325080  --  OCCURS AT 05325320
        05325240
M1  --  STREAM LABEL  --  DECLARED IN SEGMENT 43 AT 05325080  --  OCCURS AT 05325350
        05325250
M2  --  STREAM LABEL  --  DECLARED IN SEGMENT 43 AT 05325080  --  OCCURS AT 05325360
        05325260
M3  --  STREAM LABEL  --  DECLARED IN SEGMENT 43 AT 05325080  --  OCCURS AT 05325380
        05325270
M4  --  STREAM LABEL  --  DECLARED IN SEGMENT 43 AT 05325080  --  OCCURS AT 05325390
        05325280
M5  --  STREAM LABEL  --  DECLARED IN SEGMENT 43 AT 05325080  --  OCCURS AT 05325400
        05325290
M6  --  STREAM LABEL  --  DECLARED IN SEGMENT 43 AT 05325080  --  OCCURS AT 05325410
        05325300
M7  --  STREAM LABEL  --  DECLARED IN SEGMENT 43 AT 05325080  --  OCCURS AT 05325420
        05325310
N  --  REAL  --  DECLARED IN SEGMENT 3 AT 01693000
N  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 01737350
        01737450
N  --  STREAM VARIABLE  --  DECLARED IN SEGMENT 3 AT 01759000
       *01781000*
N  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 8 AT 02001760
        02001765  02001780
N  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 02041000
        02043000
N  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 02045000
        02046000  02050000
N  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 11 AT 02196530
        02196560
N  --  INTEGER  --  DECLARED IN SEGMENT 14 AT 02249000
       *02251000**02252000* 02255000  02259000
N  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 15 AT 02264400
        02264550
N  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 15 AT 02264700
        02264850
N  --  REAL  --  DECLARED IN SEGMENT 32 AT 04204000
       *04208000* 04209000  04221000  04223000
N  --  INTEGER  --  DECLARED IN SEGMENT 33 AT 04237000
       *04238000* 04239000 *04240000*
N  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 38 AT 05067000
        05075000
N  --  INTEGER  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 05325010
        05325460  05325470  05325480
```

```
N  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 43 AT 05325040
       05325050  05325230
N  --  STREAM VARIABLE  --  DECLARED IN SEGMENT 44 AT 05334200
       05334400  05334600
N  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 46 AT 05415000
       05421000
N  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 49 AT 06064000
       06067000
N  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 57 AT 07038100
       07038500
N  --  INTEGER  --  DECLARED IN SEGMENT 66 AT 07646395
       *07646495* 07646510 *07646515* 07646625  07646630
N  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 67 AT 07653500
       07655500
N  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 71 AT 08999075
       08999100
N  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 80 AT 09378000
       09381000
N  --  INTEGER  --  DECLARED IN SEGMENT 81 AT 09393000
       *09400000* 09402100  09403000  09405500  09406000  09407000 *09413000* 09417000  09420000 *09430050* 09430075
N  --  REAL  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 81 AT 09393240
       09393300
N  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 81 AT 09393700
       09393730
N  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 86 AT 10257100
       10257300
N  --  REAL  --  DECLARED IN SEGMENT 86 AT 10258100
       *10263110* 10264700  10264900
N  --  REAL  --  DECLARED IN SEGMENT 91 AT 13221000
       *13258000* 13265000  13271000
N  --  STREAM VARIABLE  --  DECLARED IN SEGMENT 96 AT 13673150
       13673300  13673400
N  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 124 AT 17051400
       17051500  17051900
NAMEDEC  --  LABEL  --  DECLARED IN SEGMENT 98 AT 14015000  --  OCCURS AT 14161000
       14020000
NAMEID  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01205200
       07046000  07048000  14161010  14161120  15276020  15276090  15295100  15301500  15305300
NAMEV  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01295000
NB  --  REAL  --  DECLARED IN SEGMENT 103 AT 14269060
       *14269448**14269540* 14269560  14269600
NBITE  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01154300
       05273400  05274200  06315400  06316400  14269448  14269600
NCII  --  INTEGER  --  DECLARED IN SEGMENT 3 AT 01624000
       *13228000* 14056000 *14057000*
NCIIO  --  INTEGER  --  DECLARED IN SEGMENT 98 AT 14031000
       *14056000*
NCR  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 2 AT 00511000
       00512000
NCR  --  INTEGER  --  DECLARED IN SEGMENT 3 AT 01330000
       02128500  02129000  02129500 *02213000**02214000**02225250**02229000**02230750* 02410000  02485000  02665000
       02680000  02697000 *02720000* 02731000  02745000  02750000  02753000  02786000  02791000  02807000  02813000
       02888000  02902000 *02903000*
NCR  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 01756000
       01761000 *01768000**01773000* 01780000
NCR  --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 9 AT 02089500
       02126000
```

```
NCR  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 02183500
        02183750  02184500
NCRV  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 9 AT 02090000
        02090500  02092500 *02102000* 02124500
NEQ  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01664000
NEQ  --  DEFINE  --  DECLARED IN SEGMENT 119 AT 17002000
NER  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 80 AT 09364000
        09368000
NESTBIT  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01001020
        02472000  07397100  14483100  14601000
NESTCTR  --  REAL  --  DECLARED IN SEGMENT 3 AT 01722000
       *05375100**14481200* 14483400
NESTCUR  --  REAL  --  DECLARED IN SEGMENT 3 AT 01722000
       *05402200* 05409100 *05409300*
NESTFORM  --  STREAM PROCEDURE  --  DECLARED IN SEGMENT 46 AT 05415000
        05426000  05431000
NESTLEVEL  --  REAL  --  DECLARED IN SEGMENT 3 AT 01484000
       *16126000**16142000* 16166000  16171000  16260000  16265000
NESTOG  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01001310
        07397100  14483100  14601000
NESTPRT  --  REAL ARRAY  --  DECLARED IN SEGMENT 3 AT 01724000
        05402000  05402200 *05402300**05404000* 05409200 *05409500* 05425000  05425400  05430500  05447000  05448000
       *14483300*
NESTS  --  PROCEDURE  --  DECLARED IN SEGMENT 108 AT 16115000
        16144000  16484000
NESTSORT  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 05437000  --  FORWARD AT 05411100
        05425200  05442000  05443000
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  02388000  09282500
NEWID  --  DEFINE  --  DECLARED IN SEGMENT 119 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
NEWTAPBUF  --  REAL ARRAY  --  DECLARED IN SEGMENT 3 AT 01490510
NEWTAPE  --  FILE  --  DECLARED IN SEGMENT 3 AT 01560000
        02194000  09282500  17002510
NEWTOG  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01001320
        02193750  09282500
NEXT  --  INTEGER PROCEDURE  --  DECLARED IN SEGMENT 3 AT 02956000
       *02968000* 02968500  02974500  02978500
NEXT  --  LABEL  --  DECLARED IN SEGMENT 107 AT 15076000  --  OCCURS AT 15253000
        15292000
NEXTCHK  --  LABEL  --  DECLARED IN SEGMENT 54 AT 06315000  --  OCCURS AT 06327000
        06315500  06317000  06320500
NEXTINFO  --  REAL  --  DECLARED IN SEGMENT 3 AT 01620000
        02265350 *02726000**09033000* 10250000  10254000  10261000  10263000 *10263100**10285000**13291000* 13302000
        13303000  13304000 *13305000* 13308000  13309000  13310000  13310200  13310580  13310950  13311000 *13312000*
        13593000 *13594000* 13770600  14055000  14166500  14217000  14319000  14329000 *14330000**14373000* 14375000
       *14463000* 16442000
NEXTLINK  --  DEFINE  --  DECLARED IN SEGMENT 65 AT 07596000
        07604050  07604080  07604090  07607000  07609000
NHI  --  REAL  --  DECLARED IN SEGMENT 3 AT 01689000
```

```
      *02681000**02785000* 02800000  02801000  02833000  02835000  02838000  02840000  12015000
NINFOO  --  INTEGER  --  DECLARED IN SEGMENT 98 AT 14030000
      *14055000* 14600000  14601000
NLAB  --  REAL  --  DECLARED IN SEGMENT 106 AT 14418000
      *14422000**14437000*
NLO  --  REAL  --  DECLARED IN SEGMENT 3 AT 01689000
      *02681000**02785000* 02800000  02833000  02835000  02838000  02840000  12014000
NLOC  --  REAL  --  DECLARED IN SEGMENT 106 AT 14418000
      *14422000**14433000* 14451200
NLOCS  --  DEFINE  --  DECLARED IN SEGMENT 98 AT 14022000
NODIMPART  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01532000
       07098000  07412000
NOHEADING  --  BOOLEAN  --  DECLARED IN SEGMENT 3 AT 01001480
      *01836000* 02195000  02195750  02196590  02214200  02228750  02264950  02421000  05039600  05046000  05325490
       07025020 *09028050* 09362000  13653500  13676000
NONLITNO  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01209000
       02851000  02924000  07604120  07665000  07666500  08032000  13226000
NONSAVNDX  --  INTEGER  --  DECLARED IN SEGMENT 81 AT 09393000
       09405500  09406000 *09407000* 09427100
NOP  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01665000
       04212100  04223100  07078000  08094000  13752000
NOP  --  DEFINE  --  DECLARED IN SEGMENT 108 AT 16012000
NORMAL  --  LABEL  --  DECLARED IN SEGMENT 35 AT 04311010  --  OCCURS AT 04311120
       04311060  04311080
NORMALREF  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01007495
       02882200  13310580
NOTOP  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01272000
       02964000  02974500  06146000
NUMBEREND  --  LABEL  --  DECLARED IN SEGMENT 24 AT 02640000  --  OCCURS AT 02845000
       02687000
NUMBERS  --  STREAM LABEL  --  DECLARED IN SEGMENT 9 AT 02091500  --  OCCURS AT 02117000
       02096000  02102500
NUMLE  --  BOOLEAN  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 69 AT 08080000
       08095000  08162000  08165000
NXTELBT  --  INTEGER  --  DECLARED IN SEGMENT 3 AT 01319000
       02644000  02910000 *02912000**02917000**08110000**08174000**09038000* 10263400 *10263500* 10277000  10282000
       12112000 *12113000* 12125000
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 91 AT 13221000
      *13249000* 13250000  13252500 *13257000* 13262000  13263000
OCTALWORDS  --  STREAM PROCEDURE  --  DECLARED IN SEGMENT 15 AT 02264400
       02264650  02265100  02265250  02265700  02265950
OCTIZE  --  BOOLEAN STREAM PROCEDURE  --  DECLARED IN SEGMENT 3 AT 02001838
      *02001852* 02001854  02686000
OK  --  LABEL  --  DECLARED IN SEGMENT 58 AT 07393100  --  OCCURS AT 07397600
       07397400
OLDL  --  REAL  --  DECLARED IN SEGMENT 65 AT 07596500
      *07597000* 07600800 *07604040* 07604067  07604070  07604100
OLDSEQ  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 01741200
       01741300
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  02228750  02235250  02421000  02421100  07025020
OMITTING  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01001340
```

```
        02194250  02195000  02195750  02214200  02228750  02235250  02421000  07025020
OPARSIZE  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01000902
        01000904  02316000  02331000  09414022  09414026
OPCLASS  --  REAL  --  DECLARED IN SEGMENT 28 AT 02981000
       *02982000* 02983000  02983500  02984000
OPCLASS  --  INTEGER  --  DECLARED IN SEGMENT 48 AT 06042000
       *06048000* 06052000  06055000
OPCODE  --  DEFINE  --  DECLARED IN SEGMENT 117 AT 16366000
        16379000
OPDC  --  DEFINE  --  DECLARED IN SEGMENT 69 AT 08016000
        08108000  08130000  08217000
OPERATOR  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01271000
        06076000
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 48 AT 06042000
       *06044000* 06054000
OPERATORS  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01299200
        06071000
OPINX  --  INTEGER  --  DECLARED IN SEGMENT 3 AT 01000800
       *02311000**02312000* 02314000  02316000  02317000  02334000  02338000  02342000  02356000
OPTIONLENGTH  --  SWITCH LABEL  --  DECLARED IN SEGMENT 19 AT 02364000
        02373000
OPTIONS  --  REAL ARRAY  --  DECLARED IN SEGMENT 3 AT 01000904
        02312000 *02314000* 02317000 *02331000**02334000**02338000**02352000**02356000**02426000**09251208* 09414024
        09414026
OPTIONWORD  --  BOOLEAN  --  DECLARED IN SEGMENT 3 AT 01000910
        01835600  02191500  02193750  02194250  02195000  02195750  02196590  02214200  02228100  02228750  02231500
        02232500  02234800  02235250 *02330000**02335000**02339000**02353000**02357000* 02421000 *02425500* 02427000
        02428000  02436000  02533000 *02577000* 02602500  02882200  02910600  04148000  04150000  04278500  04297000
        05039600  05046000  05048000  05097500  05099000  05325490  05350100  05376100  06123000  07025020  07086500
        07397100  07598100  07660600  08178100 *09028000**09253050* 09253100  09282500  09384500  09387000  09389500
        09394100  09405000  09414010  09421000  09430050  09447010  13310050  13310525  13310580  13341200  13651100
        13652000  13653500  13674000  13676000  13677100  14312100  14461500  14469100  14483100  14493500  14601000
        15090600  15091200  15184400  15184700  15301000  16159100  16318500  16495300  17094250  17094500
OROP  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01275000
        02965000  02984000
OUTDEC  --  LABEL  --  DECLARED IN SEGMENT 98 AT 14014000  --  OCCURS AT 14158000
        14019000
OUTPUTSOURCE  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 02190500
        02196500  02230500  02235000
OUTPUT1  --  PROCEDURE  --  DECLARED IN SEGMENT 122 AT 17016000
        17045000
OUTPUT2  --  PROCEDURE  --  DECLARED IN SEGMENT 124 AT 17087000
        17096900  17119000
OUTV  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01289000
OWNERR  --  LABEL  --  DECLARED IN SEGMENT 98 AT 14013000  --  OCCURS AT 14135000
        14018000
OWNTOG  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 124 AT 17047300
        17047400  17048100
OWNV  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01280000
        13383000  13723000  14161040
P  --  REAL  --  DECLARED IN SEGMENT 3 AT 01489000
```

```
        *14335000* 14387000  14430000  14451200  14463000
P  --  INTEGER  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 02635000
        02644000 *02917000* 02920000  02923000  02924000
P  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 38 AT 05050000
        05052000  05055000
P  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 38 AT 05059000
P  --  REAL  --  DECLARED IN SEGMENT 46 AT 05414100
        *05425300* 05425400
P  --  REAL  --  DECLARED IN SEGMENT 89 AT 12102000
        *12108000* 12126000
P  --  INTEGER  --  DECLARED IN SEGMENT 97 AT 13750000
        *13752000* 13755000 *13761000**13761200* 13767000
PACK  --  STREAM PROCEDURE  --  DECLARED IN SEGMENT 3 AT 04270000
        04276000  04296000
PACKIN  --  LABEL  --  DECLARED IN SEGMENT 86 AT 10258300  --  OCCURS AT 10264910
        10264000  10264200  10264400  10264420
PACKINFO  --  STREAM PROCEDURE  --  DECLARED IN SEGMENT 87 AT 10236000
        10240000  10251000  10254000
PAN  --  STREAM PROCEDURE  --  DECLARED IN SEGMENT 80 AT 09364000
        09386000
PANA  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 06413000  --  FORWARD AT 03021000
        06186000  06417000
PARAMTOG  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 124 AT 17047300
        17047400  17049410
PARM  --  BOOLEAN STREAM PROCEDURE  --  DECLARED IN SEGMENT 102 AT 14254200
        *14254700* 14259080
PARSE  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 06313000  --  FORWARD AT 03014000
        06035000  06181000  06335000
PARTIALWORD  --  DEFINE  --  DECLARED IN SEGMENT 107 AT 15076130
        15089200  15092800  15095400  15095700  15096400  15184000  15185100  15185600  15186100  15300700  15302300
        15302800  15305000  15307000
PASSFORMAT  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 05214000
        05227000
PAY  --  REAL ARRAY  --  DECLARED IN SEGMENT 124 AT 17069500
        17083000  17091510  17094300  17094400  17094650  17094700  17095500  17095550  17096000  17096300  17096450
PCTR  --  REAL  --  DECLARED IN SEGMENT 57 AT 07038000
        *07040000* 07042000 *07094000* 07098000
PDES  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01299020
        13746000
PEN  --  STREAM PROCEDURE  --  DECLARED IN SEGMENT 80 AT 09370000
        09377000  09388000
PERCENT  --  LABEL  --  DECLARED IN SEGMENT 24 AT 02637000  --  OCCURS AT 02731000
        02641000  02906000
PERIOD  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01264000
        06105000  06106100  06340000  07021000  07026000  15087600  15183300  15276140  15300000
PERMANENT  --  BOOLEAN  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 05333000
        05331000  05345000
PINFOO  --  INTEGER  --  DECLARED IN SEGMENT 98 AT 14026000
        *14375000* 14471000  14483000  14489000
PJ  --  REAL  --  DECLARED IN SEGMENT 3 AT 01617000
        *13344000* 14073000  14085000  14117000 *14120000**14323000* 14333300  14334000  14335000  14349000  14364000
        14374000 *14387000* 14411300  14415300
PLUG  --  PROCEDURE  --  DECLARED IN SEGMENT 69 AT 08020000
        08116000  08188000  08213000  08218000
PN  --  INTEGER  --  DECLARED IN SEGMENT 97 AT 13750600
        *13751100**13761110* 13770000 *13770600* 13773000
POINTER  --  INTEGER  --  DECLARED IN SEGMENT 91 AT 13219000
```

```
       *13222000* 13223000  13226000  13229000 *13240000* 13249000  13255000  13257000  13263000  13279000  13281000
        13283500  13284000  13286000 *13287000* 13290000
POL  --  LABEL  --  DECLARED IN SEGMENT 57 AT 07037000  --  OCCURS AT 07052000
        07059000
POLISHER  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 06060000  --  FORWARD AT 03009000
        06170000  07052000  07512000  07743000
POLISHV  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01212100
        07052000  07511000  13338000
PORS  --  INTEGER  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 05325010
        05325460  05325470  05325480
PORS  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 43 AT 05325040
        05325050  05325090
POSITION  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 04270000
        04271000  04273000
PRIMARY  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 06126000  --  FORWARD AT 03005000
        06016000  06018000  06023000  06035000  06049000  06147000  06172000  06181000  06182000  07076000
PRINT  --  STREAM PROCEDURE  --  DECLARED IN SEGMENT 95 AT 13633000
        13643000  13653000
PRINT  --  STREAM PROCEDURE  --  DECLARED IN SEGMENT 96 AT 13661000
        13673000  13675000
PRINTCARD  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 02182500
        02195000  02195750  02214200  02228750  02421000  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 129 AT 17091000
PRINTXREFSTATISTICS  --  PROCEDURE  --  DECLARED IN SEGMENT 124 AT 17053300
        17055200  17119200
PRL  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01666000
PROAD  --  INTEGER  --  DECLARED IN SEGMENT 98 AT 14032000
       *14313000**14320000* 14461100  14476000  14489000
PROADO  --  REAL  --  DECLARED IN SEGMENT 3 AT 01613000
        14037000 *14476000* 14602000
PROCEDUREDEC  --  LABEL  --  DECLARED IN SEGMENT 98 AT 14015000  --  OCCURS AT 14270000
        14020000  14137700
PROCID  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01187000
        07399000  07404000  13278000  14284000  14292000
PROCSTMT  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 07391000  --  FORWARD AT 03029000
        06162000  07425000  07735000
PROCV  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01293000
PROGDFSCBLDR  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 05245000  --  FORWARD AT 03047000
        05255000  07677000  13252500  13746000  14461100
PROGRAM  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 09003000
        09448000  16495200
PROINFO  --  REAL  --  DECLARED IN SEGMENT 3 AT 01608000
        07090000  13677200  13677300  14304000 *14314000**14320000* 14388100  14411200  14411300  14415200  14415300
        14457000  14461200  14469100  14470000  14483200  14483300  14483400 *14496000* 15080000
PRT  --  REAL ARRAY  --  DECLARED IN SEGMENT 3 AT 01343000
       *05248000**05252000* 09414040  09415000
PRTAD  --  REAL  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 13734000
        13731000  13732000  13746000
PRTADR  --  INTEGER  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 05247000
        05245000  05246000 *05247500* 05248000  05252000
PRTBASE  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01342000
        01343000  01724000  01725000  05424000  09039000  09414030  09415000  09419100
PRTBIT  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01001070
```

```
        02395000  05376100
PRTE  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01667000
        04018000  04023000  04033000  05316000  06110300  08068000  15100800  15304100
PRTI  --  INTEGER  --  DECLARED IN SEGMENT 3 AT 01703000
       *05349000* 07600440 *09039000* 14054000
PRTIMAX  --  INTEGER  --  DECLARED IN SEGMENT 3 AT 01703000
        05347000 *05350000**09039000* 09388000  09414100  09419000
PRTIO  --  INTEGER  --  DECLARED IN SEGMENT 98 AT 14030000
       *14054000*
PRTOG  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01001360
        05376100
PRTOP  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01342000
        01343000  01724000  01725000  05424000
PRTSIZ  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 80 AT 09370000
        09371000  09372000
PTOG  --  BOOLEAN  --  DECLARED IN SEGMENT 3 AT 01592000
        13310580  13341100  13343000  13370000 *14332000**14373000**14388000*
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
        13745000  13773000  14471000  14483000  14601000
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
        02726000  13285000  13309000  14088000  14118000  14446000
PUSHER  --  PROCEDURE  --  DECLARED IN SEGMENT 81 AT 09393240
        09393480  09396000  09402100  09406000
PUSHER  --  REAL PROCEDURE  --  DECLARED IN SEGMENT 81 AT 09393050
       *09393140**09393200* 09393230  09424000  09427500
PUT  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 05008000
        05132000  06104400  07545000  07610000  09133900  09134100  13304000  13348000  13365000  13408000  13768000
        14116000  14117000  14168000  14224000  14259150  14269600  14315000  14334000  14353000  14388100  14411200
        14415200  14430000  14451200  14457000  14470000  15376400  16138000  16166000  16167000  16172000  16257000
        16260000  17004710
PUTNBUMP  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 13589000  --  FORWARD AT 03057000
        09034000  14167000  14195000  14217000  14221000  14319000  14331000  14403000  16000600  16443000  16445000
        16446000
PUTOGETHER  --  PROCEDURE  --  DECLARED IN SEGMENT 86 AT 10234000
        10257000  10263900  10268000  10269000  10270000  10273000  10274000
PUTSEQNO  --  STREAM PROCEDURE  --  DECLARED IN SEGMENT 3 AT 02005000
        02006000  02202060  02214250  02218000  02234750  02368000  02716000  02904000
P1  --  REAL  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 03057000
P1  --  STREAM PROCEDURE  --  DECLARED IN SEGMENT 38 AT 05050000
        05096000
P1  --  INTEGER  --  NAME PARAMETER  --  DECLARED IN SEGMENT 3 AT 15070000
        15085800  15089600  15098200  15099200  15102400  15184200  15186800  15276100  15276110  15276160  15276190
        15276210  15300800  15301700  15304300  15304400 *15304600*  15305000  15306700 *15307300*
P2  --  BOOLEAN  --  DECLARED IN SEGMENT 3 AT 01588000
        13341000  13343000  13347000 *13385000* 13722000 *13723000**14133000**14161070**14318000**14321000*
P2  --  STREAM PROCEDURE  --  DECLARED IN SEGMENT 38 AT 05059000
        05096000
P3  --  BOOLEAN  --  DECLARED IN SEGMENT 3 AT 01589000
        13725000 *13726000**14133000**14138000**14139000**14140000**14141000*
Q  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 2 AT 00523000
        00524000  00526000
```

```
Q  --  ALPHA  --  DECLARED IN SEGMENT 3 AT 01316000
        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  02576000
        02588000 *02590000**02650000**02686500**02696000**02698000* 02700000 *02754000**02822000**02846000**02865000*
        02879000 *02958500* 02960500  02961000  02964000  02964500  02965000  02965500  05043000  05044000  07086200
        07086300 *13282000**14441000* 16213000  16378500
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 35 AT 04311020
       *04311050**04311070* 04311090
Q  --  BOOLEAN  --  DECLARED IN SEGMENT 44 AT 05343000
       *05344750**05371000**05374000* 05376100
Q  --  REAL  --  DECLARED IN SEGMENT 46 AT 05414100
       *05425100**05425200* 05425300
Q  --  DEFINE  --  DECLARED IN SEGMENT 69 AT 08015000
        08093000  08094000  08185000  08200000  08210000  08218000
Q  --  INTEGER  --  DECLARED IN SEGMENT 81 AT 09393010
       *09395700**09405500* 09447020
Q  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 86 AT 10257100
        10257700
QUOTE  --  LABEL  --  DECLARED IN SEGMENT 24 AT 02637000  --  OCCURS AT 02690000
        02641000
R  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 02183500
        02183750  02185750
RANGE  --  BOOLEAN PROCEDURE  --  DECLARED IN SEGMENT 3 AT 05297000
       *05304000*
RCA  --  DEFINE  --  DECLARED IN SEGMENT 108 AT 16008000
RDV  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01668000
READACARD  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 02196750  --  FORWARD AT 02065000
        02130500  02238000  02731000  09035000
READTAPE  --  PROCEDURE  --  DECLARED IN SEGMENT 12 AT 02198500
        02202250  02208500  02219500
REALARRAYID  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01203000
        07056000  07057000  13380000  13389000
REALDEC  --  LABEL  --  DECLARED IN SEGMENT 98 AT 14013000  --  OCCURS AT 14138000
        14018000  14135000  14136000  14136100
REALID  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01199000
        07060000  07069000  08038000  12035000  14138000  15276210
REALPROCID  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01195000
REALSTRPROCID  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01191000
        14294000
REALSUBID  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01184000
        14164000  14167000
REALV  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01283000
        13389000  14161120  14164000
RECOV  --  LABEL  --  DECLARED IN SEGMENT 91 AT 13220000  --  OCCURS AT 13281000
        13251000  13256000
RED  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 02011000
REED  --  REAL PROCEDURE  --  DECLARED IN SEGMENT 3 AT 08999000
       *08999175* 08999425  09399000
REFCOUNT  --  DEFINE  --  DECLARED IN SEGMENT 124 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 69 AT 08010000
       *08093000* 08094000 *08111000* 08120000 *08160000**08170000* 08185000  08200000 *08210000* 08218000  08231000
RELAD  --  REAL  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 13734000
        13731000  13732000
RELAD  --  REAL  --  DECLARED IN SEGMENT 98 AT 14029000
       *14487000* 14489000
RELOP  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01277000
        05344500  14260000  14269280  16207000
REMCOUNT  --  OWN INTEGER  --  DECLARED IN SEGMENT 86 AT 10230000
        10245000  10247000 *10248000**10249000**10263000*
REMEMBERSEQNO  --  REAL  --  DECLARED IN SEGMENT 107 AT 15075550
       *15085100* 15090600  15184400  15301000
REPEAT  --  INTEGER  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 04011000
        04010000  04013000  04014000
RESIZE  --  STREAM PROCEDURE  --  DECLARED IN SEGMENT 3 AT 02056000
RESTORESEQNUM  --  STREAM PROCEDURE  --  DECLARED IN SEGMENT 19 AT 02320200
        02602800
RESULT  --  INTEGER  --  DECLARED IN SEGMENT 3 AT 01386000
        02128500  02129000 *02327000* 02366200 *02371000**02499000* 02500000  02579000  02586000 *02590000* 02591000
        02598000 *02602700**02647000* 02648000 *02663000**02666000**02667000**02684000**02694000**02698000**02700000*
       *02744000**02747000**02749000**02751000**02752000**02754000**02757000**02788000**02793000**02806000**02809000*
       *02812000**02816000**02820000**02846000**02889000**02890000**02958500* 02959000  02959500  10265000  10272000
        10276000
RESULT  --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 9 AT 02089500
        02092500
RESULTSWITCH  --  SWITCH LABEL  --  DECLARED IN SEGMENT 24 AT 02643000
        02648000
RESULTV  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 9 AT 02089500
        02090500  02094000
RETURNSTORE  --  OWN REAL  --  DECLARED IN SEGMENT 69 AT 08010000
        08087000  08090000 *08166000**08168000**08190000* 08193000 *08207000* 08215000
RIGHT  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 08999450  --  FORWARD AT 03039500
        07675500  08999800  14461300  14603000
RINX  --  REAL  --  DECLARED IN SEGMENT 92 AT 13301000
       *13302000* 13304000  13305000
ROSE  --  LABEL  --  DECLARED IN SEGMENT 24 AT 02639000  --  OCCURS AT 02875000
        02879000  02881000
ROUND  --  LABEL  --  DECLARED IN SEGMENT 65 AT 07594000  --  OCCURS AT 07645000
        07600000  07600900  07604000
ROW  --  INTEGER  --  DECLARED IN SEGMENT 44 AT 05344000
RP  --  LABEL  --  DECLARED IN SEGMENT 50 AT 06137000
RRB1  --  BOOLEAN  --  DECLARED IN SEGMENT 3 AT 01522000
RRB2  --  BOOLEAN  --  DECLARED IN SEGMENT 3 AT 01525000
RR1  --  REAL  --  DECLARED IN SEGMENT 2 AT 00507000
       *00529000**00530000* 01557000 *14329000* 14373000
RR10  --  REAL  --  DECLARED IN SEGMENT 2 AT 00507000
RR11  --  REAL  --  DECLARED IN SEGMENT 2 AT 00507000
       *00529000* 00531000  00533000  00536000
RR2  --  REAL  --  DECLARED IN SEGMENT 2 AT 00507000
       *00530000* 01557000
RR3  --  REAL  --  DECLARED IN SEGMENT 2 AT 00507000
       *00531000**00532000* 01558000
RR4  --  REAL  --  DECLARED IN SEGMENT 2 AT 00507000
       *00532000* 01558000
```

```
RR5  --  REAL  --  DECLARED IN SEGMENT 2 AT 00507000
        *00534000**00535000* 01560000
RR6  --  REAL  --  DECLARED IN SEGMENT 2 AT 00507000
        *00534000**00535000* 01560000
RR7  --  REAL  --  DECLARED IN SEGMENT 2 AT 00507000
        *00534000**00535000* 01560000
RR8  --  REAL  --  DECLARED IN SEGMENT 2 AT 00507000
        *00537000**00538000* 01561000
RR9  --  REAL  --  DECLARED IN SEGMENT 2 AT 00507000
        *00537000**00538000* 01561000 *07599000* 07601000
RSA  --  DEFINE  --  DECLARED IN SEGMENT 108 AT 16018000
RTBRKFT  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01265000
        05274300  05278000  06141000  06316500  06320000  06322000  06422000  10264300  12126000  13395000  13402000
        14259120  14269680  15260000  15276010  15294000
RTN  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01669000
        13740000
RTPARFN  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01266000
        02976500  06115000  06175000  06416000  07067000  07096000  10264300  12039000  12126000  13756000  13761100
        13763000  14259120  14336000  16132000  16478000
RTPARFN  --  LABEL  --  DECLARED IN SEGMENT 24 AT 02639000  --  OCCURS AT 02744000
        02642000
RTS  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01670000
        07087000  13264000  13268000
S  --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 8 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  --  NAME PARAMETER  --  DECLARED IN SEGMENT 15 AT 02264400
        02264500
S  --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 15 AT 02264700
        02264800
S  --  BOOLEAN  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 03051001
        03051000
S  --  BOOLEAN  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 03067000
S  --  REAL  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 04277500
        04280000  04280500  04281500  04282000  04283500  04284000  04285000  04285500
S  --  REAL  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 04291000
        04296000  04297000
S  --  INTEGER  --  DECLARED IN SEGMENT 36 AT 04502000
        *04548000* 04551000
S  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 42 AT 05309000
        05311000
S  --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 44 AT 05334100
        05334300  05334500
S  --  REAL  --  DECLARED IN SEGMENT 45 AT 05401000
        *05407000*
S  --  STREAM VARIABLE  --  DECLARED IN SEGMENT 46 AT 05416000
        05418000  05420000
S  --  BOOLEAN  --  DECLARED IN SEGMENT 49 AT 06063000
        *06074000**06103000*
S  --  SWITCH LABEL  --  DECLARED IN SEGMENT 50 AT 06132000
        06138000
S  --  SWITCH LABEL  --  DECLARED IN SEGMENT 68 AT 07719000
        07727000
S  --  BOOLEAN  --  NAME PARAMETER  --  DECLARED IN SEGMENT 69 AT 08027000
```

```
        *08029000*
S   --  BOOLEAN  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 69 AT 08063000
        *08065000* 08069000
S   --  BOOLEAN  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 69 AT 08072000
         08075000  08079000
S   --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 86 AT 10257100
         10257200  10257800
S   --  REAL  --  DECLARED IN SEGMENT 88 AT 12004000
        *12006000**12025000**12030000**12038000**12055000*
S   --  REAL  --  DECLARED IN SEGMENT 89 AT 12102000
        *12110000*
S   --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 89 AT 12103000
         12104000
S   --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 96 AT 13673100
         13673200  13673350
S   --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 102 AT 14254200
         14254300
S   --  REAL  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 108 AT 16029000
         16032000
S   --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 124 AT 17047200
         17047700
SAF  --  BOOLEAN  --  DECLARED IN SEGMENT 3 AT 01300200
         13677400 *14299000*
SAN  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01677100
SAV  --  BOOLEAN  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 05247000
        05245000  05246000  05248000
SAVECODE  --  BOOLEAN  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 13632000
        13651000  13651100  13653000
SAVECODE  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 95 AT 13633000
         13638000
SAVEINFO  --  INTEGER  --  DECLARED IN SEGMENT 94 AT 13378000
       *13394000* 13396000 *13397000**13401000**13405000* 13408000
SAVEINFO  --  REAL  --  DECLARED IN SEGMENT 103 AT 14269060
       *14269260* 14269600  14269620
SAVEINX  --  INTEGER  --  DECLARED IN SEGMENT 20 AT 02324000
       *02331000**02342000* 02352000
SAVEIT  --  LABEL  --  DECLARED IN SEGMENT 103 AT 14269100  --  OCCURS AT 14269590
         14269450
SAVEL  --  REAL  --  DECLARED IN SEGMENT 3 AT 01623000
         13606000 *13628000* 14055000 *14609000*
SAVELO  --  INTEGER  --  DECLARED IN SEGMENT 98 AT 14030000
       *14055000* 14609000
SAVERR  --  LABEL  --  DECLARED IN SEGMENT 98 AT 14013000  --  OCCURS AT 14136000
         14018000
SAVETIME  --  INTEGER  --  DECLARED IN SEGMENT 2 AT 00503000
SAVETIMES  --  PROCEDURE  --  DECLARED IN SEGMENT 119 AT 17002015
         17002525  17091180  17117910
SAVEV  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01281000
         13726000  14292000  14298000  14299000
SAVINFO  --  REAL ARRAY  --  DECLARED IN SEGMENT 81 AT 09392300
       *09395100* 09396000  09402100 *09414070**09414110* 09418000 *09419000* 09419100  09424000  09433000
SAVL  --  REAL  --  DECLARED IN SEGMENT 109 AT 16031000
       *16032000* 16034000
SAVL  --  REAL  --  DECLARED IN SEGMENT 111 AT 16087000
       *16088000* 16096000
SAVNDX  --  INTEGER  --  DECLARED IN SEGMENT 81 AT 09393000
       *09397000* 09402100 *09403000* 09413000 *09414120**09420000* 09426000  09436000
```

```
SB  --  REAL  --  DECLARED IN SEGMENT 103 AT 14269060
       *14269446**14269520* 14269560  14269600
SBIT  --  BOOLEAN  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 07036000
        07035000  07046000
SBITF  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01154200
        05273300  05274100  06315300  06316300  14269446  14269600
SBV  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01244000
        16197000  16415000
SCAN  --  STREAM PROCEDURE  --  DECLARED IN SEGMENT 9 AT 02089500
        02127000  02128500
SCAN  --  STREAM PROCEDURE  --  DECLARED IN SEGMENT 86 AT 10257100
        10264800
SCANAGAIN  --  LABEL  --  DECLARED IN SEGMENT 24 AT 02639000  --  OCCURS AT 02646000
        02650000  02686500  02696000  02728000  02737000  02890000
SCANNER  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 02066000
        02137000  02327000  02371000  02499000  02590000  02647000  02663000  02666000  02684000  02694000  02698000
        02744000  02747000  02749000  02751000  02752000  02754000  02788000  02793000  02806000  02809000  02812000
        02816000  02820000  02889000  02890000  02958500
SCATTERFLBAT  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 13197000  --  FORWARD AT 03035000
        07395000  13208000  13327000  13767000  14206000  14305000
SCI  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01677050
SCLASS  --  REAL  --  DECLARED IN SEGMENT 57 AT 07038000
       *07044000**07046000* 07048000  07056000  07060000 *07079000* 07088000
SCOUNT  --  INTEGER  --  DECLARED IN SEGMENT 19 AT 02365100
       *02366200* 02602700
SCRAM  --  INTEGER  --  DECLARED IN SEGMENT 3 AT 01301000
       *02865000**09385000* 09386000  12110000 *12122000* 13308000  13361000  13362000  13755000
SCS  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01677150
SCV  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01246000
        16206000  16337000
SEARCH  --  ALPHA PROCEDURE  --  DECLARED IN SEGMENT 3 AT 04231000
       *04243000* 04244000  04280000  04280500  04283500  04284000
SEC  --  DEFINE  --  DECLARED IN SEGMENT 108 AT 16022000
        16342000
SECOND  --  INTEGER  --  NAME PARAMETER  --  DECLARED IN SEGMENT 3 AT 05271000
        05270000 *05273400**05274200**05281000* 05282000
SECOND  --  INTEGER  --  DECLARED IN SEGMENT 54 AT 06314000
       *06315400**06316400**06320400**06325000* 06328000  06332000
SECOND  --  INTEGER  --  DECLARED IN SEGMENT 55 AT 06339000
        06341000  06344000
SECRET  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01628000
        13331000  14333000  14345000
SED  --  DEFINE  --  DECLARED IN SEGMENT 106 AT 14419000
SED  --  DEFINE  --  DECLARED IN SEGMENT 108 AT 16017000
SEG  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 124 AT 17047200
        17047400
SEGMENT  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 13657000  --  FORWARD AT 03042000
        07676500  13681000  14461200  14602000
SEGMENTSTART  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 13632000  --  FORWARD AT 03041000
        07661500  13655000  14298000  14387000
SEGNOF  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01007405
        13310450  17094800
SEGSBIT  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01001100
        02470000  13652000  13674000
SEGSIZEMAX  --  INTEGER  --  DECLARED IN SEGMENT 3 AT 01687000
       *13678000*
SEGSTOG  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01001390
```

```
        13652000  13674000
SEMICOLON  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01230000
        05109000  07012000  07032000  07646490  07646535  10264410  13766000  14062000  14063000  14165000  14269840
        14325000  14338000  14357000  14519000  16131000
SEQ  --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 3 AT 01756000
SEQ  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 04130000
        04132000
SEQ  --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 57 AT 07038100
        07038300  07038600
SEQ  --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 80 AT 09378000
        09382000
SEQBIT  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01001110
        02191500  02234800  02393000
SEQCOMPARE  --  PROCEDURE  --  DECLARED IN SEGMENT 12 AT 02202500
        02209750  02224500
SEQERRBIT  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01001120
        02520000
SEQERRTOG  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01001420
SEQNO  --  REAL  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 02001650
        02001640  02001645  02001695
SEQNO  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 124 AT 17047200
        17047400  17050400
SEQNO  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 124 AT 17051400
        17051500  17052100
SEQNOF  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01007430
        17092600  17093400  17094800  17095750  17095900  17117702  17117708
SEQTOG  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01001400
        02191500  02234800
SEQUENCEERROR  --  STREAM PROCEDURE  --  DECLARED IN SEGMENT 3 AT 01742100
        01742110  02226750
SEQXEQTOG  --  BOOLEAN  --  DECLARED IN SEGMENT 3 AT 01001560
        *02525000*
SES  --  DEFINE  --  DECLARED IN SEGMENT 106 AT 14419000
SET  --  STREAM PROCEDURE  --  DECLARED IN SEGMENT 89 AT 12103000
        12117000
SETTING  --  BOOLEAN  --  DECLARED IN SEGMENT 3 AT 01000802
       *02327000* 02338000  02352000  02356000  02427000
SETUPHEADING  --  STREAM PROCEDURE  --  DECLARED IN SEGMENT 124 AT 17047200
        17051200  17094700
SGAVL  --  INTEGER  --  DECLARED IN SEGMENT 3 AT 01369000
SGNO  --  INTEGER  --  DECLARED IN SEGMENT 3 AT 01370000
        13310450  14054000
SGNOO  --  INTEGER  --  DECLARED IN SEGMENT 98 AT 14030000
       *14054000*
SIGNA  --  OWN BOOLEAN  --  DECLARED IN SEGMENT 69 AT 08012000
       *08056000* 08065000  08098000  08108000  08130000  08179000  08181000  08189000  08212000  08217000 *08227000*
SIGNB  --  OWN BOOLEAN  --  DECLARED IN SEGMENT 69 AT 08012000
        08050000  08112000 *08121000* 08126000  08183000  08196000 *08203000* 08214000
SIGNC  --  OWN BOOLEAN  --  DECLARED IN SEGMENT 69 AT 08012000
        08044000 *08112000* 08185000  08197000 *08202000* 08219000 *08220000*
SIMPARITH  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 06032000  --  FORWARD AT 03003000
        06006000  07085000  15097000  15186400  15303500
SIMPGO  --  BOOLEAN PROCEDURE  --  DECLARED IN SEGMENT 3 AT 07548000  --  FORWARD AT 03026000
       *07555000* 07557000  07564000  07570000  07581000  07646520
SIMPI  --  BOOLEAN PROCEDURE  --  DECLARED IN SEGMENT 69 AT 08052000
       *08061000* 08176000  08211000  08224000
SIMPLE  --  BOOLEAN PROCEDURE  --  DECLARED IN SEGMENT 69 AT 08027000
       *08033000**08038000* 08039000  08112000  08181000  08183000  08185000
```

```
SIMPLFB  --  DEFINE  --  DECLARED IN SEGMENT 69 AT 08014000
        08044000  08112000  08220000
SIMPLFV  --  DEFINE  --  DECLARED IN SEGMENT 69 AT 08015000
        08074000  08099000  08211000  08224000  08227000
SINGLBIT  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01001130
        02195000  02195750  02196590  02214200  02228750  02421000  02522000  04150000  04278500  05039600  05046000
        05097500  05099000  05325490  06123000  07025020  07086500  09384500  09387000  09389500  13653500  13676000
        17094250  17094500
SINGLTOG  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01001430
        02195000  02195750  02196590  02214200  02228750  02421000  04150000  04278500  05039600  05046000  05097500
        05099000  05325490  06123000  07025020  07086500  09384500  09387000  09389500  13653500  13676000  17094250
        17094500
SIV  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01236000
        16319000  16320000  16328000  16334000
SIZE  --  INTEGER  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 03042000
SIZE  --  INTEGER  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 05247000
        05245000  05246000  05248500  05249000  05250000  05252000  05254000
SIZE  --  INTEGER  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 07647000
        07663000  07675500  07676500  07677000
SIZE  --  INTEGER  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 13657000
        13675000  13677400  13678000
SIZE  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 96 AT 13661000
        13670000
SKAN  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 02277000
        02327000  02371000  02499000  02590000  02958500
SKANAGAIN  --  LABEL  --  DECLARED IN SEGMENT 19 AT 02360000  --  OCCURS AT 02370000
        02378000  02383000  02385000  02389000  02465000  02468000  02478000  02503000  02527000  02577250  02584000
        02595000  02599000
SKBIT  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 67 AT 07653500
        07655000
SKIP  --  LABEL  --  DECLARED IN SEGMENT 129 AT 17091100  --  OCCURS AT 17096850
        17094150
SKIPCOUNT  --  INTEGER  --  DECLARED IN SEGMENT 87 AT 10241000
       *10247000* 10253000  10254000  10256000
SKIPIT  --  DEFINE  --  DECLARED IN SEGMENT 27 AT 02974000
        02974500  02978500
SKIPS  --  PROCEDURE  --  DECLARED IN SEGMENT 108 AT 16403000
        16419000  16492000
SKIPV  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01241000
SKP  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 02001838
        02001842
SKP  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 02001858
        02001890
SKP  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 02041000
        02043000
SKSC  --  LABEL  --  DECLARED IN SEGMENT 86 AT 10259000  --  OCCURS AT 10263500
        10283000 *13755500* 13767000  13768000 *13769000*
SND  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01671000
        15099800  15186900  15276110  15304400
SOP  --  BOOLEAN  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 14003000
        14001000  14002000  14036000  14506000  14600000
SORCE  --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 3 AT 02045000
        02048000
SORTNEST  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 05412000
        14601000
SORTPRT  --  REAL ARRAY  --  DECLARED IN SEGMENT 3 AT 01725000
       *05425100* 05425400  05447000  05448000 *05452000**05455000**05459000*
```

```
SPACEITDOWN  --  DEFINE  --  DECLARED IN SEGMENT 98 AT 14023100
       14461500  14493500
SPCLMON  --  BOOLEAN  --  DECLARED IN SEGMENT 107 AT 15076300
      *15305000* 15305700  15306100  15306600
SPECIAL  --  REAL ARRAY  --  DECLARED IN SEGMENT 3 AT 01003000
       02655000  02666000  02755000  02961500  02963000  07031000 *09198000*
SPECIALCHAR  --  LABEL  --  DECLARED IN SEGMENT 24 AT 02637000  --  OCCURS AT 02651000
       02643000
SPECIALSWITCH  --  SWITCH LABEL  --  DECLARED IN SEGMENT 24 AT 02641000
       02659000
SPECTOG  --  BOOLEAN  --  DECLARED IN SEGMENT 3 AT 01593000
      *05034000* 13305100  13310300  13310500  13328000  13341200  13385000  13601000  13621000  13719000  14070000
      *14122000* 14142000  14149000  14160000  14161070  14187000  14205000  14275000  14287000 *14364000**14368000*
       14377000  14507000
SPRT  --  BOOLEAN ARRAY  --  DECLARED IN SEGMENT 3 AT 01698000
SRESULT  --  INTEGER  --  DECLARED IN SEGMENT 19 AT 02365100
      *02366200* 02602700
SSKIP  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 02045000
       02046000  02049000
SSP  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01672000
       05226000
SSS  --  REAL  --  DECLARED IN SEGMENT 49 AT 06063500
      *06068500* 06125000
STACKCT  --  REAL  --  DECLARED IN SEGMENT 3 AT 01566010
       04527000 *06054150**06158000**06172000**06174500**06179500**06296500**06299500**06305000**06332500**07041200*
      *07100500**07726990**14125500**15089000**15096400**15097800**15186100**15187000**15234500**15276115**15278000*
      *15304500**15306900*
STACKCTR  --  REAL  --  DECLARED IN SEGMENT 3 AT 01683000
       05370000 *05371000* 06068500 *06125000* 14058000  14477000 *14478000* 14481200 *14493000* 14506000 *14610000*
STACKCTR0  --  REAL  --  DECLARED IN SEGMENT 98 AT 14029000
      *14058000**14477000* 14493000  14610000
STACKHEAD  --  REAL ARRAY  --  DECLARED IN SEGMENT 3 AT 01310000
      *02724000* 02865000  05090000  05233000 *05234000* 09134100 *09134200**09405500* 09447030  09447040 *13286000*
      *13308000* 13361000 *13362000* 13768000 *13769000**14448000* 16443000
STARS  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 124 AT 17051400
       17051500  17052000  17052400
START  --  REAL  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 70 AT 08084000
       08082000  08083000  08086000
START  --  LABEL  --  DECLARED IN SEGMENT 98 AT 14017000  --  OCCURS AT 14063000
       14138000  14139000  14140000  14141000  14156000  14160000  14162000  14170000  14199000  14253000  14269000
       14269980  14379000  14504000  14521000
START  --  LABEL  --  DECLARED IN SEGMENT 101 AT 14202000
START  --  LABEL  --  DECLARED IN SEGMENT 102 AT 14254050  --  OCCURS AT 14269000
       14259090
START  --  LABEL  --  DECLARED IN SEGMENT 104 AT 14272000  --  OCCURS AT 14373000
START  --  LABEL  --  DECLARED IN SEGMENT 105 AT 14384000  --  OCCURS AT 14503000
START  --  LABEL  --  DECLARED IN SEGMENT 108 AT 16475000  --  OCCURS AT 16477000
       16485000
STARTINTRSC  --  DEFINE  --  DECLARED IN SEGMENT 73 AT 09024000
START1  --  LABEL  --  DECLARED IN SEGMENT 104 AT 14272000  --  OCCURS AT 14375000
       14325000  14364000
START2  --  LABEL  --  DECLARED IN SEGMENT 104 AT 14273000  --  OCCURS AT 14376000
STATS  --  SWITCH FORMAT  --  DECLARED IN SEGMENT 126 AT 17053500
       17054500  17054800  17055020
STD  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01673000
       08090000  12038000  13230000  15099400  15186800  15276110  15304400
STEPI  --  INTEGER PROCEDURE  --  DECLARED IN SEGMENT 3 AT 05003000
```

```
        05108000  05273100  05273900  05274300  05276000  05277000  05278000  05344500  05344520  05344610  05344670
        06069000  06071000  06085000  06094000  06106100  06199000  06316100  06316500  06318000  06319000  06320000
        06321000  06322000  06415000  06421000  07018000  07041000  07437000  07505000  07552000  07567000  07597500
        07600100  07600600  07604110  07604140  07645000  07646456  07646475  07646603  07662000  07664000  07666500
        07673000  07727070  07767000  07768100  07994000  08029000  08178000  09252000  12007000  12010000  12013000
        13351000  13392000  13395000  13402000  13404000  13753000  13754000  13756000  13766000  13771000  14126000
        14220000  14223000  14259110  14268000  14269320  14269480  14269500  14338000  14341000  14345000  14356000
        14393000  14396000  14467000  15085600  15183100  15253000  15260000  15276050  15276160  15300000  16125000
        16160000  16197000  16206000  16207000  16208000  16217000  16220000  16252000  16253000  16316000  16319000
        16324000  16326000  16340000  16367000  16368000  16372000  16384000  16406000  16434000  16435000
STEPIT  -- PROCEDURE -- DECLARED IN SEGMENT 3 AT 05002000
        05344690  06015000  06035000  06049000  06102000  06104700  06106200  06111000  06117000  06140000  06143000
        06147000  06166000  06168000  06172000  06174000  06177000  06181000  06201000  06208000  06303000  06331000
        06345000  06410000  06411000  06416000  06417000  06422000  06423000  07010000  07072000  07100000  07407000
        07485000  07487000  07493000  07495000  07505000  07516500  07518000  07552000  07569000  07573000  07580000
        07584000  07604100  07646450  07727020  07727090  07996000  08031000  08039000  08120000  08131000  08135000
        08142000  08155000  08161000  08175000  08192000  08231000  12013000  12017000  12025000  12030000  12034000
        12036000  12039000  12045000  12054000  13327000  13406000  13409000  13753100  13757000  13758000  13761100
        13772000  14166000  14193000  14259000  14259070  14259140  14269220  14269380  14269640  14269705  14269840
        14314000  14359000  14403000  14472000  15088400  15092000  15096600  15183700  15185000  15186200  15274000
        15276070  15300400  15302200  15303300  15376800  16130000  16176000  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  16000300  16253000  16437000  16443000  16481000
STMT  -- PROCEDURE -- DECLARED IN SEGMENT 3 AT 07711000  -- FORWARD AT 03027000
        05108000  07011000  07485000  07495000  07574000  07577000  07585100  07646530  07771000  08163000  08201000
        09275000  14167000  14488000  14515000
STMTSTART  -- OWN REAL  -- DECLARED IN SEGMENT 69 AT 08010000
        08157000 *08169000**08191000* 08193000 *08206000* 08221000
STOP  -- LABEL  -- DECLARED IN SEGMENT 105 AT 14384000  -- OCCURS AT 14491000
        14484000
STOPDEFINE  -- BOOLEAN  -- DECLARED IN SEGMENT 3 AT 01597000
        02893000 *02911000**02926000**07017000**07646602**10263300**10281000**12111000**13327000**14130000**14192000*
       *14258000**14259060**14259100**14259130**14269200**14402000**14403000*
STOPENTRY  -- BOOLEAN  -- DECLARED IN SEGMENT 3 AT 01594000
        13351000 *14188000**14198000**14205000**14251000**14255000**14269000**14269120**14269940**14318000**14321000*
       *14402000**14404000*
STOPGSP  -- BOOLEAN  -- DECLARED IN SEGMENT 3 AT 01694000
        13343000  13345000 *14164500**14166000**14188000**14198000**14205000**14251000**14255000**14269000**14269120*
       *14269940**14318500**14319500**14402000**14404000*
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  13745000
STORE  -- PROCEDURE  -- DECLARED IN SEGMENT 69 AT 08063000
        08070000  08130000  08135000  08138000  08216000
STORE  -- REAL  -- VALUE PARAMETER  -- DECLARED IN SEGMENT 70 AT 08084000
        08082000  08083000  08089000
STOREFIX  -- INTEGER  -- DECLARED IN SEGMENT 70 AT 08091000
       *08093000* 08157000  08171000
STREAMERR  -- LABEL  -- DECLARED IN SEGMENT 98 AT 14017000  -- OCCURS AT 14137000
        14021000
STREAMSTMT  -- PROCEDURE  -- DECLARED IN SEGMENT 3 AT 16001000  -- FORWARD AT 03040000
        07011000  13771000  14415400  16130000  16229000  16235000  16495000
```

```
STREAMTOG  --  BOOLEAN  --  DECLARED IN SEGMENT 3 AT 01417000
        02692000  02692500  04279500  04281000  04283000  04284500  05344400  07011000 *07661000**07676000**12110100*
       *12127100* 13310580  13343000 *13771000**13773000* 14077000  14082000 *14280000* 14298000  14318500  14324000
        14361000  14377000  14385000 *14461000*
STREAMV  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01297000
        07745000  07768100  14278000
STREAMWORDS  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 05230000
        05236000  13771000  13773000  14324000  14460000
STRMPROCSTMT  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 07426000  --  FORWARD AT 03030000
        06160000  07446000  07737000
STRNGCON  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01211000
        02705100  07075000  07604110  07665500  07668000  08032000  16210000  16384000
STRNGXT  --  LABEL  --  DECLARED IN SEGMENT 24 AT 02638000  --  OCCURS AT 02701000
STRPROCID  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01189000
        07431000  14281000
STUFF  --  FILE  --  DECLARED IN SEGMENT 3 AT 01561600
        05350300  09281000  13677500
STUFF  --  STREAM PROCEDURE  --  DECLARED IN SEGMENT 11 AT 02196530
        02196580
STUFFBIT  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01001140
        02516000  05350100  13677100
STUFFF  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 05199000
        05203000
STUFFTOG  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01001440
        05350100  13677100
ST1  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 9 AT 02090000
        02090500 *02120000* 02124500 *02126000*
ST2  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 9 AT 02090000
        02090500 *02121000* 02121500
SUB  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01674000
        06016000  08030000  08044000  08214000  12015000  16329000
SUBC  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 13213000
SUBDEC  --  LABEL  --  DECLARED IN SEGMENT 98 AT 14014000  --  OCCURS AT 14163000
        14019000
SUBHAND  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 06194000  --  FORWARD AT 03011000
        06156000  06209000  07733000
SUBID  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01185000
        06198000  14164000
SUBLEVEL  --  INTEGER  --  DECLARED IN SEGMENT 3 AT 01402000
        05117000 *14477000**14492000*
SUBOP  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01556500
        05344570  05344710
SUBV  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01288000
SUPERFRMTID  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01183000
        05221000  05224000
SVARMONFILE  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01529000
SW  --  SWITCH LABEL  --  DECLARED IN SEGMENT 10 AT 02191250
        02194500
SW  --  SWITCH LABEL  --  DECLARED IN SEGMENT 20 AT 02326000
        02328000
SWITCHDEC  --  LABEL  --  DECLARED IN SEGMENT 98 AT 14015000  --  OCCURS AT 14200000
        14020000  14252000
SWITCHID  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01186000
        07507000  14207000
SWITCHIT  --  PROCEDURE  --  DECLARED IN SEGMENT 19 AT 02321000
        02359000  02377000  02388000  02393000  02395000  02397000  02402000  02415000  02419100  02419200  02421100
        02425000  02435000  02470000  02472000  02474000  02480000  02490000  02496000  02506000  02508000  02511000
```

```
        02516000  02520000  02522000  02531000  02567000  02600000
SWITCHV  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01292000
SWITMONFILE  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01539000
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**02692000* 02695000 *02701000**02705100**02721000* 02722000 *02724000* 02726000
        *02755000**02798000**02813000**02823000* 02824000 *02826000* 02827000  02842000 *02848000**02850000**02851000*
        *02865000* 02875000  02876000 *02877000* 02877010 *02877020**02878000**02882000* 02882200  02886000 *02896000*
         02900000  02901000  02910000  02910200  02910400  05096000 *05233000* 05234000  05236000 *08057000* 08058000
         08059000  08061000 *08069000* 08070000 *09282600**09388000* 09389000
T  --  REAL  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 01717900
T  --  STREAM VARIABLE  --  DECLARED IN SEGMENT 3 AT 02057000
         02058000  02059000
T  --  REAL  --  DECLARED IN SEGMENT 14 AT 02249000
        *02251000**02257000**02259000* 02260000
T  --  BOOLEAN  --  DECLARED IN SEGMENT 20 AT 02323000
T  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 21 AT 02441000
         02443000  02444000
T  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 21 AT 02446000
         02448000  02449000
T  --  INTEGER  --  DECLARED IN SEGMENT 25 AT 02957500
        *02960500**02961000* 02961500 *02964000* 02966500
T  --  BOOLEAN  --  DECLARED IN SEGMENT 28 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  --  STREAM VARIABLE  --  DECLARED IN SEGMENT 38 AT 05068000
        *05074000* 05081000 *05086000*
T  --  REAL  --  DECLARED IN SEGMENT 45 AT 05401000
        *05402000* 05402100  05403000  05404000  05405000  05409000 *05409300**05409400* 05409500
T  --  REAL  --  DECLARED IN SEGMENT 46 AT 05414000
        *05425400* 05426000  05428000 *05430500* 05431000
T  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 71 AT 08999075
         08999100
T  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 80 AT 09364000
         09369000
T  --  REAL  --  DECLARED IN SEGMENT 81 AT 09393020
T  --  REAL  --  DECLARED IN SEGMENT 88 AT 12004000
        *12044000* 12047000
T  --  REAL  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 12101000
         12108000  12117000
T  --  REAL  --  DECLARED IN SEGMENT 90 AT 13199000
        *13200000* 13201000  13202000  13203000  13204000  13205000  13206000  13207000
T  --  REAL  --  DECLARED IN SEGMENT 100 AT 14163500
        *14166500* 14168000
T  --  REAL  --  DECLARED IN SEGMENT 110 AT 16067000
        *16070000*
T  --  INTEGER  --  DECLARED IN SEGMENT 117 AT 16364000
         16393000  16394000
TABLE  --  INTEGER PROCEDURE  --  DECLARED IN SEGMENT 3 AT 02635000
        *02920000**02924000* 02927000  05002000  05003000  05221000  05224000  07066000  07397210  07401000  07600950
```

```
        07604020  07748000  07751000  07768130  10263400  10282000  12020000  12112000  13753100  13761100  14063000
        14516000  15091400  15091600  15091800  15184750  15184800  15184850  15276010  15301900  15302000  15302100
        16482300
TAKE  -- REAL PROCEDURE  -- DECLARED IN SEGMENT 3 AT 03053000  -- FORWARD AT 01717700
TAKE  -- REAL PROCEDURE  -- DECLARED IN SEGMENT 3 AT 05005000
        02001810  02001820  02724000  02725000 *05006000* 05124000  05131000  05189000  06082000  06104300  06191000
        07042000  07090000  07098000  07412000  07441000  07527000  07529000  07539000  07601000  09133900  09134100
        10263110  13226000  13229000  13249000  13255000  13257000  13279000  13284000  13286000  13348000  13362000
        13364000  13365000  13408000  13677200  13767000  14084000  14088000  14118000  14168000  14259150  14269600
        14310000  14320000  14333300  14352000  14388100  14411200  14415200  14425000  14430000  14438000  14446000
        14448000  14451200  14470000  14483200  14483400  15376300  16070000  16089000  16138000  16139000  16163000
        16166000  16172000  16254000  16260000  16449000
TAKEFRST  -- REAL PROCEDURE  -- DECLARED IN SEGMENT 3 AT 05188000
       *05189000* 05222000  06197000  07435000
TALL  -- REAL  -- DECLARED IN SEGMENT 107 AT 15072000
       *15077000* 15080000 *15083000* 15085000  15085200  15086400  15086800  15090600  15092200  15092600  15095600
        15098000  15100000  15100600  15101600  15103400  15128100  15184400  15185200  15185300  15185500  15186800
        15187300  15255000  15267000  15276020  15276060  15276080  15276090  15276210  15290000  15295100  15301000
        15301200  15301500  15304000  15304200  15305300  15305500  15306200  15306400  15376300
TALLYV  -- DEFINE  -- DECLARED IN SEGMENT 3 AT 01239000
        16319000  16341000
TAN  -- DEFINE  -- DECLARED IN SEGMENT 108 AT 16014000
        16213000
TAPE  -- FILE  -- DECLARED IN SEGMENT 3 AT 01561000
        01835800  02201750  02452000  02454000  02457000  09282000  17002500
TAPE  -- STREAM VARIABLE  -- VALUE PARAMETER  -- DECLARED IN SEGMENT 3 AT 02188000
        02188500
TAPELAST  -- LABEL  -- DECLARED IN SEGMENT 12 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
       *07506000**07507000* 07516200  07518000 *10243000**10260000* 10275000
TB1  -- BOOLEAN  -- DECLARED IN SEGMENT 101 AT 14204000
       *14206000*
TCLASS  -- INTEGER  -- DECLARED IN SEGMENT 116 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*
TEDOC  -- STREAM VARIABLE  -- NAME PARAMETER  -- DECLARED IN SEGMENT 3 AT 13683000
        13686000  13703000  13706000
TEMPL  -- INTEGER  -- DECLARED IN SEGMENT 31 AT 04166000
       *04172000* 04175000  04184500  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 *09041000*
TEST  -- PROCEDURE  -- DECLARED IN SEGMENT 69 AT 08041000
        08051000  08131000  08220000
TESTCODE  -- DEFINE  -- DECLARED IN SEGMENT 114 AT 16192000
        16211000  16212500
TESTLEV  -- DEFINE  -- DECLARED IN SEGMENT 105 AT 14384100
        14387000  14389000  14411300  14415300  14451000  14461100
TESTVOID  -- LABEL  -- DECLARED IN SEGMENT 12 AT 02210500  -- OCCURS AT 02234000
        02232500
THENBRANCH  -- INTEGER  -- DECLARED IN SEGMENT 53 AT 06295000
       *06297000* 06301000
```

```
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 54 AT 06314000
      *06315400**06316400**06320200* 06320400 *06326000* 06327000  06328000  06332000
TIME1  --  REAL  --  DECLARED IN SEGMENT 3 AT 01300000
       01828000  09385000 *16495200**17091165*
TIMINGS  --  REAL ARRAY  --  DECLARED IN SEGMENT 119 AT 17002012
      *17002040**17002075**17022300* 17054800  17055010 *17080100**17091520**17094410**17095510**17096350*
TL  --  INTEGER  --  DECLARED IN SEGMENT 30 AT 04118000
      *04119000* 04124000
TL  --  INTEGER  --  DECLARED IN SEGMENT 60 AT 07483000
      *07485000* 07487000
TL  --  REAL  --  DECLARED IN SEGMENT 91 AT 13221000
      *13258000* 13277000
TLCR  --  INTEGER  --  DECLARED IN SEGMENT 3 AT 01330000
      *02202050* 02202060  02202070  02224500 *02456000*
TLCR  --  REAL  --  NAME PARAMETER  --  DECLARED IN SEGMENT 12 AT 02202750
       02202500  02203500  02204000
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 73 AT 09005000
       09006000
TOP  --  LABEL  --  DECLARED IN SEGMENT 47 AT 05439000  --  OCCURS AT 05452000
       05445000
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  04119500  04120100  04120200  04120500
TRB  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01682000
       04318000
TRNSFER  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01251000
       16379000
TRP  --  DEFINE  --  DECLARED IN SEGMENT 108 AT 16019000
       16383000
TRUTHV  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01208000
TRW  --  DEFINE  --  DECLARED IN SEGMENT 106 AT 14419000
TSSINTYPE  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01561430
       09430060
TSSTOG  --  BOOLEAN  --  DECLARED IN SEGMENT 81 AT 09393020
      *09414026* 09430050
TSUBLEVEL  --  REAL  --  DECLARED IN SEGMENT 98 AT 14029000
      *14477000* 14492000
TURNONSTOPLIGHT  --  STREAM PROCEDURE  --  DECLARED IN SEGMENT 3 AT 02011000
       02202070  02214500  02218250  02235750  02369000  02717000  02905000
TWXA  --  REAL ARRAY  --  DECLARED IN SEGMENT 3 AT 01561700
       05350180  05350300  13677400  13677500
TYPE  --  REAL  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 03055000
TYPE  --  INTEGER  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 05247000
```

```
        05245000  05246000  05248000  05249000  05250000  05253000
TYPE  --  INTEGER  --  DECLARED IN SEGMENT 53 AT 06295000
       *06299000*
TYPE  --  REAL  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 13316000
        13314000  13315000  13333500  13338000  13342000
TYPE  --  REAL  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 13716000
        13714000  13715000  13729000
TYPE  --  SWITCH LABEL  --  DECLARED IN SEGMENT 108 AT 16476000
        16477000
TYPE  --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 124 AT 17047300
        17049300
TYPEREF  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01007420
        02001695  07598100  07660600  08178100  13310525  14312100  15091200  15184700  16159100  16318500  17092000
        17092300  17092900
TYPEV  --  REAL  --  DECLARED IN SEGMENT 3 AT 01611000
       *13380000**13389000* 13390000 *14091000**14161010**14161120* 14162000 *14281000**14284000* 14285000 *14292000*
       *14294000* 14319000
TYPEV  --  REAL  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 03071000
        03069000  03070000
TYPEV  --  REAL  --  DECLARED IN SEGMENT 100 AT 14163500
       *14164000* 14165000  14167000
T1  --  REAL  --  DECLARED IN SEGMENT 3 AT 01693000
       *13381000* 13383000  13389000 *14161000**14161020* 14161040  14161120
T1  --  STREAM VARIABLE  --  DECLARED IN SEGMENT 3 AT 02001860
       *02001874* 02001876
T1  --  REAL  --  DECLARED IN SEGMENT 34 AT 04278000
       *04280500* 04281000  04281500 *04284000* 04284500  04285000
T1  --  INTEGER  --  DECLARED IN SEGMENT 36 AT 04502000
       *04550000* 04551000
T1  --  REAL  --  DECLARED IN SEGMENT 38 AT 05049000
       *05088000**05091000* 05096000 *05097000**05099000*
T1  --  REAL  --  DECLARED IN SEGMENT 49 AT 06062000
       *06072000* 06077000 *06079000* 06080000 *06081000* 06082000  06087000  06089000 *06104200* 06104600 *06107200*
       *06108000* 06110000  06110100
T1  --  REAL  --  DECLARED IN SEGMENT 51 AT 06184000
       *06185000* 06187000 *06189000* 06191000
T1  --  REAL  --  DECLARED IN SEGMENT 52 AT 06196000
       *06197000* 06206000  06207000
T1  --  REAL  --  DECLARED IN SEGMENT 59 AT 07427000
       *07435000* 07436000 *07440000* 07442000  07443000
T1  --  REAL  --  DECLARED IN SEGMENT 64 AT 07562000
       *07566000* 07572000  07573000  07575000 *07577000* 07579000  07583000  07585100
T1  --  REAL  --  DECLARED IN SEGMENT 69 AT 08173000
       *08179000**08193000* 08202000  08203000  08204000  08205000  08206000  08207000  08225000
T1  --  STREAM VARIABLE  --  DECLARED IN SEGMENT 3 AT 13684000
       *13685000* 13707000  13711000
T1  --  REAL  --  DECLARED IN SEGMENT 107 AT 15073000
        15087800  15089200  15092800  15095400  15095700  15096000  15096400  15097600  15103400  15183400  15184000
        15185100  15185600  15185900  15186100  15186700  15187300  15276060  15276070  15276150  15276210  15300100
        15300700  15302300  15302800  15303100  15303800  15305000  15307000  15307200 *15376300**15376500**15376600*
        15376700
T2  --  REAL  --  DECLARED IN SEGMENT 3 AT 01693000
T2  --  STREAM VARIABLE  --  DECLARED IN SEGMENT 3 AT 02001860
       *02001874* 02001876
T2  --  INTEGER  --  DECLARED IN SEGMENT 36 AT 04502000
       *04550000*
T2  --  REAL  --  DECLARED IN SEGMENT 38 AT 05049000
```

```
      *05092000* 05093000  05095000  05098000
T2  --  REAL  --  DECLARED IN SEGMENT 49 AT 06062000
      *06072000* 06074000  06075000  06079000 *06080000* 06081000 *06084000* 06087000  06089000 *06090000* 06092000
      *06093000* 06099100  06099200 *06104300* 06104400 *06106000**06106200* 06110200
T2  --  REAL  --  DECLARED IN SEGMENT 51 AT 06184000
      *06185000* 06189000 *06190000**06191000*
T2  --  REAL  --  DECLARED IN SEGMENT 64 AT 07562000
      *07583000**07585000* 07586000
T2  --  REAL  --  DECLARED IN SEGMENT 67 AT 07657500
      *07663000* 07668500  07669000  07671500
T2  --  REAL  --  DECLARED IN SEGMENT 69 AT 08173000
      *08198000* 08208000
T2  --  STREAM VARIABLE  --  DECLARED IN SEGMENT 3 AT 13684000
      *13686000* 13694000  13698000
T2  --  REAL  --  DECLARED IN SEGMENT 107 AT 15074000
       15087800  15096000  15097600  15103400  15183400  15185900  15186700  15187300  15276070  15276150  15276210
       15300100  15303100  15303800  15307200 *15376300* 15376400
T3  --  REAL  --  DECLARED IN SEGMENT 38 AT 05049000
      *05091000* 05092000 *05093000* 05094000  05096000 *05098000*
T3  --  REAL  --  DECLARED IN SEGMENT 69 AT 08173000
      *08199000* 08209000
T3  --  STREAM VARIABLE  --  DECLARED IN SEGMENT 3 AT 13684000
       13689000  13697000  13700000  13710000
T4  --  REAL  --  DECLARED IN SEGMENT 38 AT 05049000
      *05095000* 05096000
T4  --  REAL  --  DECLARED IN SEGMENT 69 AT 08173000
      *08200000* 08210000
U  --  REAL  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 05411100
       05437000  05440000  05441000  05443000  05446000  05454000  05457000  05458000
UNHOOK  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 13353000  --  FORWARD AT 01626000
       13305300
UNKNOWNID  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01177000
       06151000
UNTILV  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01226000
       07018000  07485000  07646603  08113000  08128000  08184000
UPDATETIMES  --  PROCEDURE  --  DECLARED IN SEGMENT 119 AT 17002050
       17022200  17091175  17119100
UPPER  --  REAL  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 05299000
       05297000  05298000  05304000
USEROPINX  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01001172
       02335000  02339000  02353000  02357000  02600000  09414022
USESWITCH  --  SWITCH LABEL  --  DECLARED IN SEGMENT 12 AT 02210750
       02212000
USETHESWITCH  --  LABEL  --  DECLARED IN SEGMENT 12 AT 02210250  --  OCCURS AT 02211750
       02228100  02228150  02229500  02231500  02232250  02234000  02234600  02235250
V  --  OWN REAL  --  DECLARED IN SEGMENT 69 AT 08010000
       08078000  08099000  08181000  08188000 *08225000**08227000*
VAL  --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 3 AT 01741200
       01741300
VALTOG  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 124 AT 17047350
       17047450  17049430
VALUEV  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01258000
       14341000
VARIABLE  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 15070000  --  FORWARD AT 03038000
       06140000  06164000  07050000  07058000  07073000  07083000  07402000  07739000  08228000  12037000  15187600
       15304700  15377000
```

```
VBIT  --  BOOLEAN  --  DECLARED IN SEGMENT 57 AT 07039000
       *07043000* 07045000
VO  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01150000
        07043000  13203000
VOIDBIT  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01001150
        02231500  02415000
VOIDCR  --  REAL  --  DECLARED IN SEGMENT 3 AT 01785000
        02231750 *02232000**02410500* 02411000
VOIDING  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01001450
        02231500
VOIDPLACE  --  REAL  --  DECLARED IN SEGMENT 3 AT 01785000
       *02232000* 02410500  02412000
VOIDRANGE  --  ALPHA  --  DECLARED IN SEGMENT 19 AT 02365200
        02410000  02411000  02412000  02485000  02486000  02487000
VOIDTAPE  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01001460
        02228100  02232500
VOIDTBIT  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01001160
        02228100  02232500  02490000
VOIDTCR  --  REAL  --  DECLARED IN SEGMENT 3 AT 01785000
        02228125  02228150  02233000 *02233500**02485500* 02486000
VOIDTPLACE  --  REAL  --  DECLARED IN SEGMENT 3 AT 01785000
       *02233500* 02485500  02487000
VONE  --  BOOLEAN  --  DECLARED IN SEGMENT 3 AT 01590000
       *13203000**13341000* 13373000 *16000400*
VP  --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 3 AT 01756000
        01761000  01780000
VRET  --  OWN REAL  --  DECLARED IN SEGMENT 69 AT 08010000
       *08054000* 08077000 *08176000* 08198000 *08208000* 08211000  08224000 *08228000*
W  --  INTEGER  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 01717700
W  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 02054000
        02055000
W  --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 42 AT 05309000
        05311000
W  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 96 AT 13673100
        13673250
WAITINGFORFWDREF  --  OWN BOOLEAN  --  DECLARED IN SEGMENT 129 AT 17091110
       *17092100* 17093000 *17093200*
WD  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 73 AT 09005000
        09006000
WHAT  --  REAL  --  DECLARED IN SEGMENT 82 AT 09393080
       *09393170* 09393180  09393190
WHATISIT  --  LABEL  --  DECLARED IN SEGMENT 19 AT 02362000  --  OCCURS AT 02578000
        02364000  02365000  02379000  02398000  02475000  02517000  02568000
WHILESTMT  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 07490000
        07496000  07757000
WHILEV  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01224000
        08113000  08133000  08139000
WHO  --  REAL  --  DECLARED IN SEGMENT 82 AT 09393080
       *09393110* 09393130  09393170 *09393190* 09393200
WHOLE  --  REAL  --  DECLARED IN SEGMENT 59 AT 07427000
       *07430000* 07432000  07438000  07439000 *07441000* 07442000  07444000
WITHV  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01259000
WOP  --  REAL ARRAY  --  DECLARED IN SEGMENT 3 AT 01371000
        02534000 *02536000* 04280500  04281000  04281500  04284000  04284500  04285000
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 92 AT 13301000
       *13302000* 13307000  13310000  13312000
WRITEAX  --  STREAM PROCEDURE  --  DECLARED IN SEGMENT 57 AT 07038100
        07086300
WRITELINE  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 02181000
        02195000  02195750  02196590  02214200  02228750  02421000  04150000  05039600  05046000  05097500  05099000
        05325490  06123000  07025020  07086500  09384500  09387000  09389500  13653500  13676000
WRITEOUT  --  STREAM PROCEDURE  --  DECLARED IN SEGMENT 49 AT 06064000
        06120000
WRITEPRT  --  PROCEDURE  --  DECLARED IN SEGMENT 3 AT 05325010
        05325500  05376100
WRITERROR  --  STREAM PROCEDURE  --  DECLARED IN SEGMENT 37 AT 05016000
        05031000  05044000
WRITNEW  --  BOOLEAN STREAM PROCEDURE  --  DECLARED IN SEGMENT 3 AT 02016000
       *02020000* 02194000
X  --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 38 AT 05067000
        05072000  05074000
X  --  REAL  --  DECLARED IN SEGMENT 83 AT 09393270
       *09393400* 09393410  09393420  09393430
X  --  REAL  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 3 AT 13591000
        13589000  13590000  13593000
XBIT  --  INTEGER  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 19 AT 02321000
        02327000  02335000  02339000  02353000  02357000
XCH  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01675000
        04507000  06203000  08065000  08123000  08228000  12054000  15100400  15303900
XINFO  --  REAL ARRAY  --  DECLARED IN SEGMENT 3 AT 01007245
        02001660  02001695  02001795  02001815 *02001822**13310200* 13310350 *13310950*
XIT  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01676000
        13743000
XIT  --  LABEL  --  DECLARED IN SEGMENT 12 AT 02210500  --  OCCURS AT 02237750
        02213250  02214750
XIT  --  STREAM LABEL  --  DECLARED IN SEGMENT 43 AT 05325080  --  OCCURS AT 05325430
        05325340  05325350  05325370  05325380  05325390  05325400  05325410
XIT  --  LABEL  --  DECLARED IN SEGMENT 66 AT 07646436  --  OCCURS AT 07646674
        07646458
XITR  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01430000
XLUN  --  INTEGER  --  DECLARED IN SEGMENT 3 AT 01007380
       *13310450* 16495300  17004500 *17004600* 17022300 *17025100**17025300* 17025400 *17043000*
XMARK  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01007455
        07598100  07660600  08178100  13310525  14312100  15091200  15184700  16159100  16318500
XMAS  --  REAL  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 81 AT 09393050
        09393110  09393130  09393140  09393170 *09393180* 09393190  09393200
XMODE  --  INTEGER  --  DECLARED IN SEGMENT 3 AT 01001530
        02328000 *02332000**02367000**02383000**02385000* 02408000  02419100  02425500 *02478000* 02483000  02525000
        02577000
XMODE0  --  LABEL  --  DECLARED IN SEGMENT 20 AT 02325000  --  OCCURS AT 02329000
        02326000
XMODE1  --  LABEL  --  DECLARED IN SEGMENT 20 AT 02325000  --  OCCURS AT 02333000
        02326000
XMODE2  --  LABEL  --  DECLARED IN SEGMENT 20 AT 02325000  --  OCCURS AT 02337000
        02326000
XMODE3  --  LABEL  --  DECLARED IN SEGMENT 20 AT 02325000  --  OCCURS AT 02341000
        02326000
XMODE4  --  LABEL  --  DECLARED IN SEGMENT 20 AT 02325000  --  OCCURS AT 02355000
```

```
        02326000
XREF  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01001461
        02882200  07598100  07660600  08178100  13310050  13310525  13310580  13341200  14312100  14469100  15090600
        15091200  15184400  15184700  15301000  16159100  16318500  16495300
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**07660600**08178100**13310525**14312100**15091200**15184700**16159100**16318500*
       *17003000* 17004000  17076000  17079000
XREFBIT  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01001170
        02419100  02882200  07598100  07660600  08178100  13310050  13310525  13310580  13341200  14312100  14469100
        15090600  15091200  15184400  15184700  15301000  16159100  16318500  16495300
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  13310950
XRFFINFO  --  DEFINE  --  DECLARED IN SEGMENT 119 AT 17002005
        17025100  17025300  17080000
XREFIT  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01007440
        02882200  13310575  14469100  15090600  15184400  15301000
XREFPT  --  INTEGER  --  DECLARED IN SEGMENT 3 AT 01007355
        02001670 *02001685* 02001695 *02001705* 07598100  07660600  08178100  13310525 *13341300* 14312100  15091200
        15184700  16159100  16318500 *17003000**17004600**17074000**17077000* 17079000 *17084000* 17091400  17094200
       *17094600* 17095900 *17096100**17096400**17118000*
Y  --  REAL  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 81 AT 09393240
        09393290 *09393330* 09393350  09393400 *09393430* 09393440  09393450
Z  --  REAL  --  DECLARED IN SEGMENT 3 AT 01490000
        13746000 *14411200**14415200* 14457000 *14506000*
Z  --  INTEGER  --  DECLARED IN SEGMENT 49 AT 06063500
       *06099100*
ZERO  --  STREAM PROCEDURE  --  DECLARED IN SEGMENT 67 AT 07658500
        07660500  07662000
ZP1  --  DEFINE  --  DECLARED IN SEGMENT 3 AT 01677000


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

PHASE ONE - SORT  1773 IDENTIFIERS
        1:09 ELAPSED TIME (MIN:SEC)
        0:35 PROCESSOR TIME
        0:51 I/O TIME

PHASE TWO - SORT  11995 REFERENCES
        2:03 ELAPSED TIME (MIN:SEC)
        1:53 PROCESSOR TIME
        0:20 I/O TIME

PHASE THREE - PRINT CROSS REFERENCE (   3686 LINES)
        1:16 ELAPSED TIME (MIN:SEC)
        0:51 PROCESSOR TIME
        0:41 I/O TIME


 LABEL  000000000LINE   001772342 COMPILE ESPOL/DISK ALGOL LIBRARY                                    ALGOL  /ESPOL
```