ESPOL /DISK

SOURCE FILE: SYMBOL /ESPOL

```
 $SET OMIT LISTA = LIST                                                    %113-
BEGIN COMMENT OUTERMOST BLOCK;

 INTEGER ERRORCOUNT; COMMENT NUMBER OF ERROR MSGS, MCP WILL TYPE
                     SYNTX ERR AT EOJ IF THIS IS NON-ZERO, MUST BE @R+25;
 INTEGER SAVETIME; COMMENT SAVE-FACTOR FOR CODE FILE, GIVEN BY MCP.
                   IF COMPILE & GO =0, FOR SYNTAX, =-1. MUST BE AT R+26;
INTEGER CARDNUMBER; % SEQ # OF CARD BEING PROCESSED.
INTEGER CARDCOUNT;   % NUMBER OF CARDS PROCESSED.
 BOOLEAN BUILDLINE;
    COMMENT RR1-RR11 ARE USED IN SOME PROCEDURES IN
    PLACE OF LOCALS TO SAVE STACK SPACE;
REAL RR1,RR2,RR3,RR4,RR5,RR6,RR7,RR8,RR9,RR10,RR11;
    COMMENT SOME OF THE RRI ARE USED TO PASS FILE INFORMATION
         TO THE MAIN BLOCK;
    COMMENT EXAMIN RETURNS THE CHARACTER AT ABSOLUTE ADDRESS NCR;
REAL STREAM PROCEDURE EXAMIN(NCR); VALUE NCR;
         BEGIN SI←NCR;DI←LOC EXAMIN;DI←DI+7; DS←CHR END;
INTEGER STREAM PROCEDURE GETF(Q);VALUE Q;
    BEGIN SI←LOC GETF; SI←SI-7;DI←LOC Q;DI←DI+5;
         SKIP 3 DB; 9(IF SB THEN DS←SET ELSE DS←RESET; SKIP SB);
         DI←LOC Q;SI←Q;DS←WDS;SI←Q;GETF←SI
    END GETF;
COMMENT START SETTING UP FILE PARAMETERS;
         IF EXAMIN(RR11←GETF(3)+"Y08") ≠12 THEN RR1←5 ELSE
         BEGIN RR1←2;RR2←150 END;
         IF EXAMIN(RR11+5) ≠12 THEN RR3←4 ELSE
         BEGIN RR3←2; RR4←150 END;
         IF EXAMIN(RR11+10)=12 THEN
              BEGIN RR5←2;RR6←10;RR7←150 END ELSE
         BEGIN RR5←1;RR6←56;RR7←10 END;
         IF EXAMIN(RR11+15)=12 THEN
              BEGIN RR8←10;RR9←150 END ELSE
BEGIN RR8←50×10+50+1; RR9←10 END;                                          %901-
BEGIN COMMENT MAIN BLOCK;
INTEGER OPINX; % USED FOR INDEXING INTO OPTIONS ARRAY.

BOOLEAN SETTING; % USED BY DOLLARCARD FOR AN OPTION"S SETTING.
 INTEGER NEWINX, ADDVALUE, BASENUM, TOTALNO;
DEFINE OPARSIZE = 200 #;
ARRAY OPTIONS[0:OPARSIZE];
BOOLEAN OPTIONWORD;
DEFINE   CHECKBIT       =  1#,
         DEBUGBIT       =  2#,
```

```
                                                    00000999 C  0001:0000:0
                                                    00500000 T  0001:0000:0
                                          START OF SEGMENT **********    2
                                                    00501000 T  0002:0000:0
                                                    00502000 T  0002:0000:0
                                                    00503000 T  0002:0000:0
                                                    00504000 T  0002:0000:0
                                                    00504100 T  0002:0000:0
                                                    00504150 T  0002:0000:0
                                                    00504700 T  0002:0000:0
                                                    00505000 T  0002:0000:0
                                                    00506000 T  0002:0000:0
                                                    00507000 T  0002:0000:0
                                                    00508000 T  0002:0000:0
                                                    00509000 T  0002:0000:0
                                                    00510000 T  0002:0000:0
                                                    00511000 T  0002:0000:0
                                                    00512000 T  0002:0000:0
                                                    00523000 T  0002:0002:0
                                                    00524000 T  0002:0002:0
                                                    00525000 T  0002:0003:2
                                                    00526000 T  0002:0005:2
                                                    00527000 T  0002:0006:1
                                                    00528000 T  0002:0007:3
                                                    00529000 T  0002:0007:3
                                                    00530000 T  0002:0013:2
                                                    00531000 T  0002:0016:1
                                                    00532000 T  0002:0020:0
                                                    00533000 T  0002:0022:0
                                                    00534000 T  0002:0024:0
                                                    00535000 T  0002:0027:2
                                                    00536000 T  0002:0029:3
                                                    00537000 T  0002:0032:0
                                                    00538000 P  0002:0034:0
                                                    01000000 T  0002:0037:3
                                                    01000800 T  0002:0037:3
                                          START OF SEGMENT **********    3
                                                    01000802 T  0003:0000:0
                                                    01000860 T  0003:0000:0
                                                    01000902 T  0003:0000:0
                                                    01000904 T  0003:0000:0
                                                    01000910 T  0003:0002:1
                                                    01000920 T  0003:0002:1
                                                    01000930 T  0003:0002:1
```

```
                DECKBIT         =  3#,
                FORMATBIT       =  4#,
                INTBIT          =  5#,
                LISTABIT        =  6#,
                LISTBIT         =  7#,
                LISTPBIT        =  8#,
                MCPBIT          =  9#,
                MERGEBIT        = 10#,
                NESTBIT         = 11#,
                NEWBIT          = 12#,
                NEWINCLBIT      = 13#,
                OMITBIT         = 14#,
                PRINTDOLLARBIT  = 15#,
                PRTBIT          = 16#,
                PUNCHBIT        = 17#,
                PURGEBIT        = 18#,
                SEGSBIT         = 19#,
                SEQBIT          = 20#,
                SEQERRBIT       = 21#,
                SINGLBIT        = 22#,
                STUFFBIT        = 23#,
                VOIDBIT         = 24#,
                VOIDTBIT        = 25#,
                XREFBIT         = 26#,                                    %108-
                BENDBIT         = 27#,                                    %108-
                USEROPINX       = 28#;                                    %108-
        COMMENT IF A NEW COMPILER-DEFINED OPTION IS ADDED, CHANGE USEROPINX
                AND ADD OPTION IN DEFINES BELOW, IN DOLLARCARD, AND IN
                FILL STATEMENT IN INITIALIZATION OF COMPILER;
        DEFINE  CHECKTOG        = OPTIONWORD.[CHECKBIT:1] #,
                DEBUGTOG        = OPTIONWORD.[DEBUGBIT:1] #,
                DECKTOG         = OPTIONWORD.[DECKBIT:1] #,
                FORMATOG        = OPTIONWORD.[FORMATBIT:1] #,
                INTOG           = OPTIONWORD.[INTBIT:1] #,
                LISTATOG        = OPTIONWORD.[LISTABIT:1] #,
                LISTOG          = OPTIONWORD.[LISTBIT:1] #,
                LISTPTOG        = OPTIONWORD.[LISTPBIT:1] #,
                MCPTOG          = OPTIONWORD.[MCPBIT:1] #,
                MERGETOG        = OPTIONWORD.[MERGEBIT:1] #,
                NESTOG          = OPTIONWORD.[NESTBIT:1] #,
                NEWTOG          = OPTIONWORD.[NEWBIT:1] #,
                NEWINCL         = OPTIONWORD.[NEWINCLBIT:1] #,
                OMITTING        = OPTIONWORD.[OMITBIT:1] #,
                PRINTDOLLARTOG  = OPTIONWORD.[PRINTDOLLARBIT:1] #,
                PRTOG           = OPTIONWORD.[PRTBIT:1] #,
                PUNCHTOG        = OPTIONWORD.[PUNCHBIT:1] #,
                PURGETOG        = OPTIONWORD.[PURGEBIT:1] #,
                SEGSTOG         = OPTIONWORD.[SEGSBIT:1] #,
                SEQTOG          = OPTIONWORD.[SEQBIT:1] #,
        COMMENT SEQTOG INDICATES RESEQUENCING IS TO BE DONE;
                SEQERRTOG       = OPTIONWORD.[SEQERRBIT:1] #,
                SINGLTOG        = OPTIONWORD.[SINGLBIT:1] #,
                STUFFTOG        = OPTIONWORD.[STUFFBIT:1] #,
                VOIDING         = OPTIONWORD.[VOIDBIT:1] #,
                VOIDTAPE        = OPTIONWORD.[VOIDTBIT:1] #,
                XREF            = OPTIONWORD.[XREFBIT:1] #,               %108-
                BEND            = OPTIONWORD.[BENDBIT:1]#,                %108-
```

```
01000940 T  0003:0002:1
01000950 T  0003:0002:1
01000960 T  0003:0002:1
01000970 T  0003:0002:1
01000980 T  0003:0002:1
01000990 T  0003:0002:1
01001000 T  0003:0002:1
01001010 T  0003:0002:1
01001020 T  0003:0002:1
01001030 T  0003:0002:1
01001040 T  0003:0002:1
01001050 T  0003:0002:1
01001060 T  0003:0002:1
01001070 T  0003:0002:1
01001080 T  0003:0002:1
01001090 T  0003:0002:1
01001100 T  0003:0002:1
01001110 T  0003:0002:1
01001120 T  0003:0002:1
01001130 T  0003:0002:1
01001140 T  0003:0002:1
01001150 T  0003:0002:1
01001160 T  0003:0002:1
01001170 P  0003:0002:1
01001171 C  0003:0002:1
01001172 C  0003:0002:1
01001180 T  0003:0002:1
01001190 T  0003:0002:1
01001200 T  0003:0002:1
01001210 T  0003:0002:1
01001220 T  0003:0002:1
01001230 T  0003:0002:1
01001240 T  0003:0002:1
01001250 T  0003:0002:1
01001260 T  0003:0002:1
01001270 T  0003:0002:1
01001280 T  0003:0002:1
01001290 T  0003:0002:1
01001300 T  0003:0002:1
01001310 T  0003:0002:1
01001320 T  0003:0002:1
01001330 T  0003:0002:1
01001340 T  0003:0002:1
01001350 T  0003:0002:1
01001360 T  0003:0002:1
01001370 T  0003:0002:1
01001380 T  0003:0002:1
01001390 T  0003:0002:1
01001400 T  0003:0002:1
01001410 T  0003:0002:1
01001420 T  0003:0002:1
01001430 T  0003:0002:1
01001440 T  0003:0002:1
01001450 T  0003:0002:1
01001460 T  0003:0002:1
01001461 C  0003:0002:1
01001462 C  0003:0002:1
```

```
                DUMMY           = #;
BOOLEAN NOHEADING;      % TRUE IF DATTME HAS NOT BEEN CALLED.
BOOLEAN NEWBASE;        % NEW BASENUM FOUND ON A NEW $-CARD.
BOOLEAN LASTCRDPATCH;   % NORMALLY FALSE, SET TO TRUE WHEN THE
                        % LAST CARD FROM SYMBOLIC LIBRARY READ
                        % IS PATCHED FROM THE CARD READER.
INTEGER XMODE;          % TELLS DOLLARCARD HOW TO SET OPTIONS.
BOOLEAN DOLLARTOG;      % TRUE IF SCANNING A DOLLAR CARD.
INTEGER ERRMAX;         % COMPILATION STOPS IF EXCEEDED.
BOOLEAN SEQXEQTOG;      % GIVE SEQ. NO. WHEN DS-ING OBJ.
BOOLEAN LISTER;         % LISTOG OR LISTATOG OR DEBUGTOG.
ALPHA MEDIUM;           % INPUT IS: T,C,P,CA,CB,CC.
INTEGER MYCLASS;        % USED IN DOLLARCARD EVALUATION.
REAL BATMAN;            % USED IN DOLLARCARD EVALUATION.
 ARRAY SPECIAL[0:31];
    COMMENT THIS ARRAY HOLDS THE INTERNAL CODE FOR THE SPECIAL
            CHARACTORS; IT IS FILLED DURING INITIALIZATION;

ARRAY INFO [0:127,0:255];
%*****************************************************************
%               X R E F   S T U F F                             %110-
%*****************************************************************
%                                                               %110-
ARRAY                                                           %110-
  XREFAY2[0:29],        % ARRAY OF ONE WORD REFFRENCE RECORDS.  %110-
                        % THE LAYOUT OF EACH WORD IS            %110-
                        %                                      %110-
                        %       ,[1:5]  TYPE OF REFERENCE      %110-
                        %                 = 0 FOR FORWARD DECL  %110-
                        %                 = 1 FOR LABEL OCCURENCE %110-
                        %                 = 2 FOR NORMAL DECL   %110-
                        %                 = 4 FOR NORMAL REFERENCE %110-
                        %                 = 5 FOR ASSIGNMENT    %110-
                        %                                      %110-
                        %                NOTE: THE LOWER ORDER BIT
                        %                      OF THIS FIELD IS ON
                        %                      IF YOU WANT STARS%110-
                        %                      AROUND THIS REFERENCE
                        %                      IN THE XREF     %110-
                        %                                      %110-
                        %       ,[6:15] IDENTIFIER ID. NO.     %110-
                        %                THIS IS A UNIQUE NUMBER THAT
                        %                IS ASSIGNED WHEN THE  %110-
                        %                IDENTIFIER IS ENCOUNTERE %110-
                        %                FOR THE FIRST TIME.   %110-
                        %                                      %110-
                        %       ,[21:27] SEQUENCE NUMBER       %110-
                        %                                      %110-
  XREFAY1[0:9],         % RECORD BUFFER AREA FOR WRITING OUT THE %110-
                        % NAME INFORMATION RECORDS, ONE RECORD  %110-
                        % IS WRITTEN FOR EACH IDENTIFIER IN THE SYMBOL
                        % TABLE WHEN THE IDENTIFIER IS PURGED FROM THE
                        % SYMBOL TABLE, I.E., WHEN LEAVING THE BLOCK%110-
                        % IN WHICH THE IDENTIFIER IS DECLARED.  %110-
                        %                                      %110-
                        % THE LAYOUT OF EACH RECORD IS:         %110-
                        %                                      %110-
```

```
01001470 T  0003:0002:1
01001480 T  0003:0002:1
01001490 T  0003:0002:1
01001500 T  0003:0002:1
01001510 T  0003:0002:1
01001520 T  0003:0002:1
01001530 T  0003:0002:1
01001540 T  0003:0002:1
01001550 T  0003:0002:1
01001560 T  0003:0002:1
01001570 T  0003:0002:1
01001580 T  0003:0002:1
01001590 T  0003:0002:1
01001600 T  0003:0002:1
01003000 T  0003:0002:1
01004000 T  0003:0005:2
01005000 T  0003:0005:2
01006000 T  0003:0005:2
01007000 T  0003:0005:2
01007005 C  0003:0007:3
01007010 C  0003:0007:3
01007015 C  0003:0007:3
01007020 C  0003:0007:3
01007025 C  0003:0007:3
01007030 C  0003:0007:3
01007035 C  0003:0010:1
01007040 C  0003:0010:1
01007045 C  0003:0010:1
01007050 C  0003:0010:1
01007051 C  0003:0010:1
01007055 C  0003:0010:1
01007060 C  0003:0010:1
01007065 C  0003:0010:1
01007070 C  0003:0010:1
01007075 C  0003:0010:1
01007080 C  0003:0010:1
01007085 C  0003:0010:1
01007090 C  0003:0010:1
01007095 C  0003:0010:1
01007100 C  0003:0010:1
01007105 C  0003:0010:1
01007110 C  0003:0010:1
01007115 C  0003:0010:1
01007120 C  0003:0010:1
01007125 C  0003:0010:1
01007130 C  0003:0010:1
01007135 C  0003:0010:1
01007140 C  0003:0010:1
01007145 C  0003:0010:1
01007150 C  0003:0013:2
01007155 C  0003:0013:2
01007160 C  0003:0013:2
01007165 C  0003:0013:2
01007170 C  0003:0013:2
01007175 C  0003:0013:2
01007180 C  0003:0013:2
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
% 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
                        % 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
                        % 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
  BOOLEAN DEFINING;                                                %108-    01007650 C  0003:0018:0
  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                 01052000 T  0003:0018:0
```

```
          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+3-31] POINTS TO THE ENTRY FOR SYMBOL.
       ALL ENTRIES OF IDNETIFIERS HAVE THE INFORMATION DESCRIBED ABOVE
       THAT IS,THE ELBAT WORD FOLLOWED BY THE WORD CONTAING THE FIRST
       FIVE CHARACTERS OF ALPHA,AND ANY ADDITIONAL WORDS OF ALPHA IF
        NECESSARY.
       THIS IS SUFFICIENT FOR ENTRIES OF THE FOLLOWING TYPES.
             REAL
             BOOLEAN
             INTEGER
             ALPHA
             FILE
             FORMAT
             LIST
         OTHER ENTRIES REQUIRE ADDITIONAL INFORMATION.
      ARRAYS:
          THE FIRST WORD OF ADDITIONAL INFO CONTAINS THE NUMBER OF
          DIMENSIONS(IN THE LOW ORDER PART).[40:8]
         EACH SUCCEEDING WORD CONTAINS INFORMATION ABOUT EACH LOWER
          BOUND IN ORDER OF APPEARANCE,ONE WORD FOR EACH LOWER BOUND.
          THESE WORDS ARE MADE UP AS FOLLOWS:
               [23:12] =ADD OPERATOR SYLLABLE (0101) OR
                        SUB OPERATOR SYLLABLE (0301) CORRESPONDING
                        RESPECTIVELY TO WHETHER THE LOWER BOUND IS
```

```
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
01109000 T  0003:0018:0
```

```
                        TO BE ADDED TO THE SUBSCRIPT IN INDEXING OR                    01110000 T  0003:0018:0
                        SUBTRACTED.                                                    01111000 T  0003:0018:0
                  [35:11] =11 BIT ADDRESS OF LOWER BOUND,IF THE LOWER BOUND            01112000 T  0003:0018:0
                        REQUIRES A PRT OR STACK CELL,OTHERWISE THE BIT                 01113000 T  0003:0018:0
                        35 IS IGNORED AND THE NEXT TEN BITS([36:10])                   01114000 T  0003:0018:0
                        REPRESENT THE ACTUAL VALUE OF THE LOWER BOUND                  01115000 T  0003:0018:0
                   [46:2] =00 OR 10  DEPENDING ON WHETHER THE [35:11] VALUE            01116000 T  0003:0018:0
                        IS A LITERAL OR OPERAND,RESPECTIVELY.                          01117000 T  0003:0018:0
PROCEDURES:                                                                            01118000 T  0003:0018:0
    THE FIRST WORD OF ADDITIONAL INFO CONTAINS THE NUMBER OF                           01119000 T  0003:0018:0
    PARAMETERS [40:8]                                                                  01120000 T  0003:0018:0
    IF A STREAM PROCEDURE THEN THIS WORD CONTAINS ALSO IN                              01121000 T  0003:0018:0
    [13:11] ENDING  PRT ADDRESS FOR LABELS,                                            01122000 T  0003:0018:0
    [ 7:6]  NO OF LABELS REQUIRING PRT ADDRESSES, AND [1:6] NUMBER                     01123000 T  0003:0018:0
    OF LOCALS.                                                                         01124000 T  0003:0018:0
    SUCCEEDING WORDS (ONE FOR EACH FORMAL PARAMETER,IN ORDER                           01125000 T  0003:0018:0
                        OF APPEARANCE IN FORMAL PARAPART) ARE                          01126000 T  0003:0018:0
    ELBAT WORDS SPECIFYING TYPE OF EACH PARAMETER AND WHETHER                          01127000 T  0003:0018:0
    VALUE OR NOT([10:1]).                                                              01128000 T  0003:0018:0
    THE ADDRESS([16:11]) IS THE F- ADDRESS FOR EACH.                                   01129000 T  0003:0018:0
    IF THE PARAMETER IS AN ARRAY THEN THE INCR FIELD([27:8])                           01130000 T  0003:0018:0
    CONTAINS THE NUMBER OF DIMENSIONS,OTHERWISE INCR IS MEANINGLESS.                   01131000 T  0003:0018:0
    LINK([35:13]) IS MEANINGLESS.                                                      01132000 T  0003:0018:0
    IF A STREAM PROCEDURE THEN THE CLASS OF EACH PARAMATER IS                          01133000 T  0003:0018:0
    THAT OF LOCAL ID OR FILE ID, DEPENDING ON WHETHER OR NOT A RELEASE                 01134000 T  0003:0018:0
    IS DONE IN THE STREAM PROCEDURE.                                                   01135000 T  0003:0018:0
LABELS:                                                                                01136000 T  0003:0018:0
    AT DECLARATION TIME THE ADDITIONAL INFO CONTAINS 0.  THE SIGN                      01137000 T  0003:0018:0
    BIT TELLS WHETHER OR NOT THE DEFINITION POINT HAS BEEN REACHED.                    01138000 T  0003:0018:0
    IF SIGN = 0, THEN [36:12] CONTAINS AN ADDRESS IN CODEARRAY OF A                    01139000 T  0003:0018:0
    LIST OF FORWARD REFERENCES TO THIS LABEL.  THE END OF LIST FLAG IS                 01140000 T  0003:0018:0
    0.  IF SIGN =0, THEN [36:12] CONTAINS L FOR THIS LABEL.                            01141000 T  0003:0018:0
SWITCHES:                                                                              01142000 T  0003:0018:0
    THE FIELD [36:12] CONTAINS L FOR THE BEGINNING OF SWITCH DECLAR-                   01143000 T  0003:0018:0
    ATION.  [24:12] CONTAINS L FOR FIRST SIMPLE REFERENCE TO SWITCH.                   01144000 T  0003:0018:0
    IF SWITCH IS NOT SIMPLE, IT IS MARKED FORMAL.  HERE SIMPLE MEANS                   01145000 T  0003:0018:0
    NO POSSIBILITY OF JUMPING OUT OF A BLOCK.                                ;         01146000 T  0003:0018:0
            DEFINE MON    =[ 1: 1]#,                                                   01147000 T  0003:0018:0
                   CLASS  =[ 2: 7]#,                                                   01148000 T  0003:0018:0
                   FORMAL =[ 9: 1]#,                                                   01149000 T  0003:0018:0
                   VO     =[10: 1]#,                                                   01150000 T  0003:0018:0
                   LVL    =[11: 5]#,                                                   01151000 T  0003:0018:0
                   ADDRESS=[16:11]#,                                                   01152000 T  0003:0018:0
                   INCR   =[27: 8]#,                                                   01153000 T  0003:0018:0
                   LINK   =[35:13]#,                                                   01154000 T  0003:0018:0
                   SBITF  =[21:6]#, % STARTING BIT FOR FIELD ID.  %112-                01154200 C  0003:0018:0
                   NBITF  =[27:6]#, % NUMBER OF BITS FOR FIELD ID.%112-                01154300 C  0003:0018:0
                   LINKR  =[35: 5]#,                                                   01155000 T  0003:0018:0
                   LINKC  =[40: 8]#;                                                   01156000 T  0003:0018:0
COMMENT THESE DEFINES ARE USED TO PICK APART THE ELBAT WORD.                           01157000 T  0003:0018:0
            MON IS THE BIT WHICH IS ON IF THE QUANTITY IS MONITORED.                   01158000 T  0003:0018:0
            CLASS IS THE PRINCIPAL IDENTIFICATION OF A GIVEN                           01159000 T  0003:0018:0
                QUANTITY.                                                              01160000 T  0003:0018:0
            FORMAL IS THE BIT WHICH IS ON IF THE QUANTITY IS A FORMAL                  01161000 T  0003:0018:0
                PARAMETER.                                                             01162000 T  0003:0018:0
            VO IS THE VALUE-OWN BIT. IF FORMAL = 1 THEN THE BIT                        01163000 T  0003:0018:0
                DISTINGUISHES VALUE PARAMETERS FROM OTHERS, IF                         01164000 T  0003:0018:0
```

```
                    FORMAL = 0 THEN THE BIT DISTINGUISHES OWN VARIABLES                01165000 T  0003:0018:0
                    FROM OTHERS.                                                       01166000 T  0003:0018:0
                LVL GIVES THE LEVEL AT WHICH A QUANTITY WAS DECLARED.                  01167000 T  0003:0018:0
                ADDRESS GIVES THE STACK OR PRT ADDRESS.                                01168000 T  0003:0018:0
                INCR GIVES A RELATIVE LINK TO ANY ADDITIONAL INFORMATION               01169000 T  0003:0018:0
                    NEEDED, RELATIVE TO THE LOCATION IN INFO.                          01170000 T  0003:0018:0
                LINK CONTAINS A LINK TO THE LOCATION IN INFO IF THE                    01171000 T  0003:0018:0
                    QUANTITY LIES IN ELBAT, OTHERWISE IT LINKS TO THE                  01172000 T  0003:0018:0
                    NEXT ITEM IN THE STACK. ZERO IS AN END FLAG.                       01173000 T  0003:0018:0
                LINKR AND LINKC ARE SUBDIVISIONS OF LINK.;                             01174000 T  0003:0018:0
COMMENT  CLASSES FOR ALL QUANTITIES  -  OCTAL CLASS IS IN COMMENT;                     01175000 T  0003:0018:0
   COMMENT   CLASSES FOR IDENTIFIERS;                                                  01176000 T  0003:0018:0
DEFINE  UNKNOWNID       =00#,    COMMENT 000;                                       01177000 T  0003:0018:0
        STLABID         =01#,    COMMENT 001;                                       01178000 T  0003:0018:0
        LOCLID          =02#,    COMMENT 002;                                       01179000 T  0003:0018:0
        DEFINEDID       =03#,    COMMENT 003;                                       01180000 T  0003:0018:0
        LISTID          =04#,    COMMENT 004;                                       01181000 T  0003:0018:0
        FRMTID          =05#,    COMMENT 005;                                       01182000 T  0003:0018:0
        SUPERFRMTID     =06#,    COMMENT 006;                                       01183000 T  0003:0018:0
        REALSUBID       =07#,    COMMENT 007;                                       01184000 T  0003:0018:0
        SUBID           =08#,    COMMENT 010;                                       01185000 T  0003:0018:0
        SWITCHID        =09#,    COMMENT 011;                                       01186000 T  0003:0018:0
        PROCID          =10#,    COMMENT 012;                                       01187000 T  0003:0018:0
        INTRNSICPROCID  =11#,    COMMENT 013;                                       01188000 T  0003:0018:0
        STRPROCID       =12#,    COMMENT 014;                                       01189000 T  0003:0018:0
        BOOSTRPROCID    =13#,    COMMENT 015;                                       01190000 T  0003:0018:0
        REALSTRPROCID   =14#,    COMMENT 016;                                       01191000 T  0003:0018:0
        ALFASTRPROCID   =15#,    COMMENT 017;                                       01192000 T  0003:0018:0
        INTSTRPROCID    =15#,    COMMENT 017;                                       01193000 T  0003:0018:0
        BOOPROCID       =17#,    COMMENT 021;                                       01194000 T  0003:0018:0
        REALPROCID      =18#,    COMMENT 022;                                       01195000 T  0003:0018:0
        ALFAPROCID      =19#,    COMMENT 023;                                       01196000 T  0003:0018:0
        INTPROCID       =19#,    COMMENT 023;                                       01197000 T  0003:0018:0
        BOOID           =21#,    COMMENT 025;                                       01198000 T  0003:0018:0
        REALID          =22#,    COMMENT 026;                                       01199000 T  0003:0018:0
        ALFAID          =23#,    COMMENT 027;                                       01200000 T  0003:0018:0
        INTID           =23#,    COMMENT 027;                                       01201000 T  0003:0018:0
        BOOARRAYID      =25#,    COMMENT 031;                                       01202000 T  0003:0018:0
        REALARRAYID     =26#,    COMMENT 032;                                       01203000 T  0003:0018:0
        ALFAARRAYID     =27#,    COMMENT 033;                                       01204000 T  0003:0018:0
        INTARRAYID      =27#,    COMMENT 033;                                       01205000 T  0003:0018:0
        NAMEID          =30#,    COMMENT 036;                                       01205200 T  0003:0018:0
        INTNAMEID       =31#,    COMMENT 037;                                       01205400 T  0003:0018:0
        LABELID         =32#,    COMMENT 040;                                       01206000 T  0003:0018:0
COMMENT   CLASSES FOR PRIMARY BEGINNERS;                                           01207000 T  0003:0018:0
        TRUTHV          =33#,    COMMENT 041;                                       01208000 T  0003:0018:0
        NONLITNO        =34#,    COMMENT 042;                                       01209000 T  0003:0018:0
        LITNO           =35#,    COMMENT 043;                                       01210000 T  0003:0018:0
        STRNGCON        =36#,    COMMENT 044;                                       01211000 T  0003:0018:0
        LEFTPAREN       =37#,    COMMENT 045;                                       01212000 T  0003:0018:0
        POLISHV         =38#,    COMMENT 046;                                       01212100 T  0003:0018:0
        ASTRISK         =39#,    COMMENT 047;                                       01212200 T  0003:0018:0
COMMENT   CLASS FOR ALL DECLARATORS;                                               01213000 T  0003:0018:0
        DECLARATORS     =40#,    COMMENT 050;                                       01214000 T  0003:0018:0
COMMENT   CLASSES FOR STATEMENT BEGINNERS                                          01215000 T  0003:0018:0
        DOUBLEV         =42#,    COMMENT 052;                                       01222000 T  0003:0018:0
        FORV            =43#,    COMMENT 053;                                       01223000 T  0003:0018:0
```

```
          WHILEV          =44#,    COMMENT 054;                                    01224000 T  0003:0018:0
          DOV             =45#,    COMMENT 055;                                    01225000 T  0003:0018:0
          UNTILV          =46#,    COMMENT 056;                                    01226000 T  0003:0018:0
          ELSEV           =47#,    COMMENT 057;                                    01227000 T  0003:0018:0
          ENDV            =48#,    COMMENT 060;                                    01228000 T  0003:0018:0
          CASEV           =49#,    COMMENT 061;                          %115-     01229000 C  0003:0018:0
          SEMICOLON       =50#,    COMMENT 062;                                    01230000 T  0003:0018:0
          IFV             =51#,    COMMENT 063;                                    01231000 T  0003:0018:0
          GOV             =52#,    COMMENT 064;                                    01232000 T  0003:0018:0
          IOCLASS         =53#,    COMMENT 065;                                    01233000 T  0003:0018:0
          BEGINV          =54#,    COMMENT 066;                                    01234000 T  0003:0018:0
COMMENT   CLASSES FOR STREAM RESERVED WORDS;                                       01235000 T  0003:0018:0
          SIV             =55#,    COMMENT 067;                                    01236000 T  0003:0018:0
          DIQ             =56#,    COMMENT 070;                                    01237000 T  0003:0018:0
          CIV             =57#,    COMMENT 071;                                    01238000 T  0003:0018:0
          TALLYV          =58#,    COMMENT 072;                                    01239000 T  0003:0018:0
          DSV             =59#,    COMMENT 073;                                    01240000 T  0003:0018:0
          SKIPV           =60#,    COMMENT 074;                                    01241000 T  0003:0018:0
          JUMPV           =61#,    COMMENT 075;                                    01242000 T  0003:0018:0
          DBV             =62#,    COMMENT 076;                                    01243000 T  0003:0018:0
          SBV             =63#,    COMMENT 077;                                    01244000 T  0003:0018:0
          TOGGLEV         =64#,    COMMENT 100;                                    01245000 T  0003:0018:0
          SCV             =65#,    COMMENT 101;                                    01246000 T  0003:0018:0
          LOCV            =66#,    COMMENT 102;                                    01247000 T  0003:0018:0
          DCV             =67#,    COMMENT 103;                                    01248000 T  0003:0018:0
          LOCALV          =68#,    COMMENT 104;                                    01249000 T  0003:0018:0
          LITV            =69#,    COMMENT 105;                                    01250000 T  0003:0018:0
          TRNSFER         =70#,    COMMENT 106;                                    01251000 T  0003:0018:0
COMMENT   CLASSES FOR VARIOUS MISCELLANEOUS QUANTITIES;                            01252000 T  0003:0018:0
          COMMENTV        =71#,    COMMENT 107;                                    01253000 T  0003:0018:0
          FORWARDV        =72#,    COMMENT 110;                                    01254000 T  0003:0018:0
          STEPV           =73#,    COMMENT 111;                                    01255000 T  0003:0018:0
          THENV           =74#,    COMMENT 112;                                    01256000 T  0003:0018:0
          TOV             =75#,    COMMENT 113;                                    01257000 T  0003:0018:0
          VALUEV          =76#,    COMMENT 114;                                    01258000 T  0003:0018:0
          WITHV           =77#,    COMMENT 115;                                    01259000 T  0003:0018:0
          COLON           =78#,    COMMENT 116;                                    01260000 T  0003:0018:0
          COMMA           =79#,    COMMENT 117;                                    01261000 T  0003:0018:0
          CROSSHATCH      =80#,    COMMENT 120;                                    01262000 T  0003:0018:0
          LFTBRKET        =81#,    COMMENT 121;                                    01263000 T  0003:0018:0
          PERIOD          =82#,    COMMENT 122;                                    01264000 T  0003:0018:0
          RTBRKET         =83#,    COMMENT 123;                                    01265000 T  0003:0018:0
          RTPAREN         =84#,    COMMENT 124;                                    01266000 T  0003:0018:0
          AMPERSAND       =85#,    COMMENT 125;                                    01266500 T  0003:0018:0
COMMENT   CLASSES FOR OPERATORS;                                                   01267000 T  0003:0018:0
          HEXOP           =86#,    COMMENT 126;                                    01268000 T  0003:0018:0
          BITOP           =87#,    COMMENT 127;                                    01269000 T  0003:0018:0
          ISOLATE         =88#,    COMMENT 130;                                    01270000 T  0003:0018:0
          OPERATOR        =89#,    COMMENT 131;                                    01271000 T  0003:0018:0
          NOTOP           =90#,    COMMENT 132;                                    01272000 T  0003:0018:0
          ASSIGNOP        =91#,    COMMENT 133;                                    01273000 T  0003:0018:0
          EQVOP           =92#,    COMMENT 134;                                    01274000 T  0003:0018:0
          OROP            =93#,    COMMENT 135;                                    01275000 T  0003:0018:0
          ANDOP           =94#,    COMMENT 136;                                    01276000 T  0003:0018:0
          RELOP           =95#,    COMMENT 137;                                    01277000 T  0003:0018:0
          ADOP            =96#,    COMMENT 140;                                    01278000 T  0003:0018:0
          MULOP           =97#,    COMMENT 141;                                    01278500 T  0003:0018:0
```

```
%       STRING          =99#,    COMMENT 143;                                  %112-   01278600 T  0003:0018:0
        FIELDID         =125#,   COMMENT 175;                                  %112-   01278700 C  0003:0018:0
    COMMENT   SUBCLASSES FOR DECLARATORS (KEPT IN ADDRESS);                             01279000 T  0003:0018:0
        OWNV            =01#,    COMMENT  01;                                       01280000 T  0003:0018:0
        SAVEV           =02#,    COMMENT  02;                                       01281000 T  0003:0018:0
        BOOV            =03#,    COMMENT  03;                                       01282000 T  0003:0018:0
        REALV           =04#,    COMMENT  04;                                       01283000 T  0003:0018:0
        ALFAV           =05#,    COMMENT  05;                                       01284000 T  0003:0018:0
        INTV            =05#,    COMMENT  05;                                       01285000 T  0003:0018:0
        LABELV          =07#,    COMMENT  07;                                       01286000 T  0003:0018:0
        DUMPV           =08#,    COMMENT  10;                                       01287000 T  0003:0018:0
        SUBV            =09#,    COMMENT  11;                                       01288000 T  0003:0018:0
        OUTV            =10#,    COMMENT  12;                                       01289000 T  0003:0018:0
        INV             =11#,    COMMENT  13;                                       01290000 T  0003:0018:0
        MONITORV        =12#,    COMMENT  14;                                       01291000 T  0003:0018:0
        SWITCHV         =13#,    COMMENT  15;                                       01292000 T  0003:0018:0
        PROCV           =14#,    COMMENT  16;                                       01293000 T  0003:0018:0
        ARRAYV          =15#,    COMMENT  17;                                       01294000 T  0003:0018:0
        NAMEV           =16#,    COMMENT  20;                                       01295000 T  0003:0018:0
        FILEV           =17#,    COMMENT  21;                                       01296000 T  0003:0018:0
        STREAMV         =18#,    COMMENT  22;                                       01297000 T  0003:0018:0
        DEFINEV         =19#,    COMMENT  23;                                  %112-   01298000 P  0003:0018:0
        AUXMEMV         =20#,    COMMENT 24;                                   %112-   01298500 C  0003:0018:0
        FIELDV          =21#;    COMMENT 25;                                   %112-   01298600 C  0003:0018:0
DEFINE DDES         =  8#,                                                              01299000 T  0003:0018:0
       ADES         = 28#,                                                              01299010 T  0003:0018:0
       PDES         = 29#,                                                              01299020 T  0003:0018:0
       LDES         = 30#,                                                              01299030 T  0003:0018:0
       CHAR         = 31#,                                                              01299040 T  0003:0018:0
       FACTOP       = ASTRISK#,                                                         01299100 T  0003:0018:0
       OPERATORS    = HEXOP#,                                                           01299200 T  0003:0018:0
       FILEID       = 0#,                                                               01299300 T  0003:0018:0
       MAXINTRINSIC = 150#, % USED IN BUILDING INTABLE @ 09414120                       01299400 T  0003:0018:0
       INTRINSICADR = (MAXINTRINSIC DIV 30)#; % RESERVES SEG FOR INTABLE                01299500 T  0003:0018:0
REAL TIME1;                                                                             01300000 T  0003:0018:0
BOOLEAN ASTOG;                                                                          01300100 T  0003:0018:0
BOOLEAN SAF;                                                                            01300200 T  0003:0018:0
INTEGER SCRAM;                                                                          01301000 T  0003:0018:0
   COMMENT SCRAM CONTAINS THE SCRAMBLE INDEX FOR THE LAST IDENTIFIER                    01302000 T  0003:0018:0
           OR RESERVED WORD SCANNED;                                                    01303000 T  0003:0018:0
ALPHA ARRAY ACCUM[0:10];                                                                01304000 T  0003:0018:0
   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
   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
   COMMENT COUNT CONTAINS THE NUMBER OF CHARACTORS OF THE LAST ITEM                     01314000 T  0003:0023:2
           SCANNED;                                                                     01315000 T  0003:0023:2
ALPHA Q;                                                                                01316000 T  0003:0023:2
   COMMENT Q CONTAINS ACCUM[1] FOR THE LAST IDENTIFIER OR RESERVED                      01317000 T  0003:0023:2
           WORD SCANNED;                                                                01318000 T  0003:0023:2
ARRAY ELBAT[0:75]; INTEGER I, NXTELBT;                                                  01319000 T  0003:0023:2
   COMMENT ELBAT IS AN ARRAY HOLDING ELBAT WORDS FOR RECENTLY SCANNED                   01320000 T  0003:0025:3
```

```
                    QUANTITIES.  THE TABLE ROUTINE MAINTAINS THIS ARRAY.
                    (ELBAT IS TABLE SPELLED BACKWARDS.)  THE TABLE ROUTINE
                    GUARANTIES THAT ELBAT ALWAYS CONTAINS THE ELBAT WORDS
                    FOR THE LAST 10 QUANTITIES SCANNED.  NXTELBT IS AN INDEX
                    POINTING TO THE NEXT AVAILABLE WORD IN ELBAT.  I IS AN
                    INDEX USED BY THE REST OF THE COMPILER TO FETCH THINGS
                    FROM ELBAT.  I IS ALSO MAINTAINED BY THE TABLE ROUTINE;
  INTEGER ELCLASS;
     COMMENT ELCLASS USUALLY CONTAINS ELBAT[I].CLASS;
  INTEGER FCR, NCR, LCR,TLCR,CLCR;
  INTEGER MAXTLCR;
     COMMENT FCR CONTAINS ABSOLUTE ADDRESS OF THE FIRST CHARACTOR OF
                    THE CARD IMAGE CURRENTLY BEING SCANNED, NCR THE ADDRESS
                    OF THE NEXT CHARACTOR TO BE SCANNED, AND LCR THE LAST
                    CHARACTOR (COLUMN 73). TLCR AND CLCR CONTAIN ADDRESS OF
                    THE LAST CHARACTER IN THE TAPE AND CARD BUFFERS.  MAXTLCR
                    IS THE MAXIMUM OF TLCR WHEN THE INPUT IS BLOCKED;
  ARRAY TEN[-46:69];

  DEFINE PRTBASE=129#,PRTOP=896#;  COMMENT PASE AND TOP OF PRT;
 ARRAY PRT[PRTBASE:PRTOP];
 INTEGER DISKADR,CORADR; COMMENT GLOBALS FOR PROGDESCRLDR;
 INTEGER SGAVL;COMMENT NEXT AVAILABLE SEGMENT NUMBER;
 INTEGER SGNO;COMMENT THIS IS THE CURRENT SEGMENT NUMBER;
          ARRAY  COP,WOP[0:127];
     COMMENT  THE EMIT ROUTINES PLACE EACH SYLLABLE INTO THE EDOC ARRAY
                    AS SPECIFIED BY "L".
                    IF THE DEBUGTOG IS TRUE COP AND WOP ARE FILLED WITH
                    THE BCD FOR THE OPERATORS,OTHERWISE THEY ARE NOT USED;
 REAL  LASTENTRY ;
     COMMENT LASTENTRY IS USED BY EMITNUM AND CONSTANTCLEAN. IT POINTS
                    INTO INFO[0,*] AT THE NEXT AVAILABLE CELL FOR CONSTANTS;
 BOOLEAN MRCLEAN ;
     COMMENT  NO  CONSTANTCLEAN ACTION TAKES PLACE WHILE MRCLEAN IS
                    FALSE. THIS FEATURE IS USED BY BLOCK BECAUSE OF THE
                    POSSIBILITY THAT CONSTANTCLEAN WILL USE INFO[NEXTINFO]
                    DURING AN ARRAY DECLARATION ;
 REAL GT1,GT2,GT3,GT4,GT5;
 INTEGER GTI1;
     COMMENT THESE VARIABLES ARE USED FOR TEMPORARY STORAGE;
 INTEGER RESULT;
     COMMENT THIS VARIABLE IS USED FOR A DUAL PURPOSE BY THE TABLE
                    ROUTINE AND THE SCANNER.  THE TABLE ROUTINE USES THIS
                    VARIABLE TO SPECIFY SCANNER OPERATIONS AND THE SCANNER
                    USES IT TO INFORM THE TABLE ROUTINE OF THE ACTION TAKEN;
 INTEGER LASTUSED;
     COMMENT LASTUSED IS A VARIABLE THAT CONTROLS THE ACTION OF
                    READACARD, THE ROUTINE WHICH READS CARDS AND INITIALIZES
                    OR PREPARES THE CARD FOR THE SCANNER.
          LASTUSED  LAST CARD READ FROM
          --------  ------------------
              1     CARD READER ONLY, NO TAPE.
              2     CARD READER, TAPE AND CARD MERGE.
              3     TAPE, TAPE AND CARD MERGE.
              4     INITIALIZATION ONLY, CARD ONLY.
 ;
 BOOLEAN LINKTOG;
```

```
01321000 T  0003:0025:3
01322000 T  0003:0025:3
01323000 T  0003:0025:3
01324000 T  0003:0025:3
01325000 T  0003:0025:3
01326000 T  0003:0025:3
01327000 T  0003:0025:3
01328000 T  0003:0025:3
01329000 T  0003:0025:3
01330000 T  0003:0025:3
01331000 T  0003:0025:3
01332000 T  0003:0025:3
01333000 T  0003:0025:3
01334000 T  0003:0025:3
01335000 T  0003:0025:3
01336000 T  0003:0025:3
01337000 T  0003:0025:3
01340000 T  0003:0025:3
01341000 T  0003:0028:0
01342000 T  0003:0028:0
01343000 T  0003:0028:0
01344000 T  0003:0030:1
01369000 T  0003:0030:1
01370000 T  0003:0030:1
01371000 T  0003:0030:1
01372000 T  0003:0034:0
01373000 T  0003:0034:0
01374000 T  0003:0034:0
01375000 T  0003:0034:0
01376000 T  0003:0034:0
01377000 T  0003:0034:0
01378000 T  0003:0034:0
01379000 T  0003:0034:0
01380000 T  0003:0034:0
01381000 T  0003:0034:0
01382000 T  0003:0034:0
01383000 T  0003:0034:0
01384000 T  0003:0034:0
01384500 T  0003:0034:0
01385000 T  0003:0034:0
01386000 T  0003:0034:0
01387000 T  0003:0034:0
01388000 T  0003:0034:0
01389000 T  0003:0034:0
01390000 T  0003:0034:0
01391000 T  0003:0034:0
01392000 T  0003:0034:0
01393000 T  0003:0034:0
01394000 T  0003:0034:0
01394500 T  0003:0034:0
01394600 T  0003:0034:0
01395000 T  0003:0034:0
01396000 T  0003:0034:0
01397000 T  0003:0034:0
01398000 T  0003:0034:0
01398300 T  0003:0034:0
01399000 T  0003:0034:0
```

```
   COMMENT LINKTOG IS FALSE IF THE LAST THING EMITTED IS A LINK,
           OTHERWISE IT IS TRUE;
 INTEGER LEVEL,FRSTLEVEL,SUBLEVEL,MODE;
   COMMENT THESE VARIABLES ARE MAINTAINED BY THE BLOCK ROUTINE TO KEEP
           TRACK OF LEVELS OF DEFINITION.  LEVEL GIVES THE DEPTH OF
           NESTING IN DEFINITION, WHERE EACH BLOCK AND EACH PROCEDURE
           GIVES RISE TO A NEW LEVEL.  SUBLEVEL GIVES THE LEVEL OF
           THE PARAMETERS OF THE PROCEDURE CURRENTLY BEING COMPILED.
           FRSTLEVEL IS THE LEVEL OF THE PARAMETERS OF THE MOST
           GLOBAL OF THE PROCEDURES CURRENTLY BEING COMPILED.  MODE
           IS THE CURRENT DEPTH OF THE PROCEDURE IN WHICH WE ARE
           NESTED (AT COMPILE TIME);
BOOLEAN ERRORTOG;
   COMMENT ERRORTOG IS TRUE IF MESSAGES ARE CURRENTLY ACCEPTABLE TO THE
           ERROR ROUTINES.  ERRORCOUNT IS THE COUNT OF ERROR MSSGS;
BOOLEAN ENDTOG;    COMMENT ENDTOG TELLS THE TABLE TO ALLOW
                   COMMENT TO BE PASSED BACK TO COMPOUNDTAIL;
 BOOLEAN STREAMTOG;
   COMMENT STREAMTOG IS TRUE IF WE ARE COMPILING STREAM STATEMENT.  IT
           IS USED TO CONTROL COUMPOUNDTAIL;
 DEFINE FS = 1#, FP = 2#, FL = 3#, FR =4#;
   COMMENT THESE DEFINES ARE USED WHEN CALLING THE VARIABLE ROUTINE.
           THEIR PURPOSES IS TO TELL VARIABLE WHO IS CALLING.
           THEIR MEANING IS:
           FS MEANS FROM STATEMENT,
           FP MEANS FROM PRIMARY,
           FL MEANS FROM LIST,
           FR MEANS FROM FOR;
INTEGER L;
   COMMENT   L IS THE LOCATION OF THE NEXT SYLLABLE TO BE EMITTED;
DEFINE BLOCKCTR = 16#, JUNK = 17 #, XITR = 18 #, LSTRTN = 19#;
DEFINE ATYPE =3#, BTYPE=ATYPE#,DTYPE=ATYPE#;
BOOLEAN TB1;
   COMMENT TB1 IS A TEMPORARY BOOLEAN VARIABLE;
INTEGER JUMPCTR;
   COMMENT   JUMPCTR IS A VARIABLE USED FOR COMMUNICATION BETWEEN BLOCK
           AND GENGO.  IT GIVES HIGHEST LEVEL TO WHICH A JUMP HAS
           BEEN MADE FROM WITHIN A THE PRESENTLY BEING COMPILED
           SEGMENT.  THE BLOCK COMPILES CODE TO INCREMENT AND DECRE-
           MENT THE BLOCKCTR ON THE BASIS OF JUMPCTR AT COMPLETION
           OF COMPILATION OF A SEGMENT - I.E. THE BLOCKCTR IS TALLIED
           IF LEVEL = JUMPCTR;



DEFINE BUMPL = L←L+2#;
   COMMENT BUMPL IS USED MOSTLY TO PREPARE A FORWARD JUMP;
DEFINE IDMAX =   LABELID#;
   COMMENT IDMAX IS THE MAXIMUM CLASS NUMBER FOR IDENTIFIERS;
INTEGER DEFINECTR,DEFINEINDEX;
   REAL  JOINFO,       COMMENT POINTS TO PSEUDO LABEL FOR JUMP OUTS;
        LPRT, COMMENT SHOWS LOCATION OF THE LAST LABEL IN THE PRT ;
     NESTLEVEL,        COMMENT  COUNTS NESTING FOR GO TO AND JUMP OUTS;
         JUMPLEVEL;    COMMENT NUMBER OF LEVELS TO BE JUMPED OUT;
   COMMENT THE REALS ABOVE ARE FOR STREAM STATEMENT;
   ARRAY  MACRO[0:35];
```

```
01400000 T  0003:0034:0
01401000 T  0003:0034:0
01402000 T  0003:0034:0
01403000 T  0003:0034:0
01404000 T  0003:0034:0
01405000 T  0003:0034:0
01406000 T  0003:0034:0
01407000 T  0003:0034:0
01408000 T  0003:0034:0
01409000 T  0003:0034:0
01410000 T  0003:0034:0
01411000 T  0003:0034:0
01412000 T  0003:0034:0
01413000 T  0003:0034:0
01414000 T  0003:0034:0
01415000 T  0003:0034:0
01416000 T  0003:0034:0
01417000 T  0003:0034:0
01418000 T  0003:0034:0
01419000 T  0003:0034:0
01420000 T  0003:0034:0
01421000 T  0003:0034:0
01422000 T  0003:0034:0
01423000 T  0003:0034:0
01424000 T  0003:0034:0
01425000 T  0003:0034:0
01426000 T  0003:0034:0
01427000 T  0003:0034:0
01428000 T  0003:0034:0
01429000 T  0003:0034:0
01430000 T  0003:0034:0
01452000 T  0003:0034:0
01457000 T  0003:0034:0
01458000 T  0003:0034:0
01459000 T  0003:0034:0
01460000 T  0003:0034:0
01461000 T  0003:0034:0
01462000 T  0003:0034:0
01463000 T  0003:0034:0
01464000 T  0003:0034:0
01465000 T  0003:0034:0
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
01477000 T  0003:0034:0
01478000 T  0003:0034:0
01479000 T  0003:0034:0
01480000 T  0003:0034:0
01481000 T  0003:0034:0
01482000 T  0003:0034:0
01483000 T  0003:0034:0
01484000 T  0003:0034:0
01485000 T  0003:0034:0
01486000 T  0003:0034:0
01487000 T  0003:0034:0
```

```
    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
        Z;     COMMENT CONTAINS 1ST WORD OF INFO FOR STREAM FUNCTIONS;             01490000 T  0003:0036:1
      ARRAY NEWTAPBUF[0:9];                                                        01490510 T  0003:0036:1
 SAVE ARRAY DEFINEARRAY[0:34];                                          %116-     01491000 P  0003:0039:2
    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
    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
    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
              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
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  0003:0041:3
                           THE PRINTI ROUTINE;                                     01556000 T  0003:0041:3
DEFINE SUBOP=48#;                                                                  01556500 T  0003:0041:3
 FILE OUT CODE DISK SERIAL[1:1](1,1023);                                           01556900 T  0003:0041:3
```

```
FILE IN CARD(RR1,10,RR2);
FILE OUT LINE DISK SERIAL[20:2400](RR3,15,RR4,SAVE 10);
 ARRAY LINE[0:20]; COMMENT PRINT OUTPUT BUILT IN LIN;
INTEGER DA;
SAVE FILE OUT NEWTAPE DISK SERIAL[20:2400](RR5,RR6,RR7,SAVE 1);
FILE IN TAPE "OCRDIMG"(2,RR8,RR9);
SAVE ARRAY CBUFF,TBUFF[0:9]; % INPUT BUFFERS,
        FILE DSK1 DISK SERIAL [20:816](2,10,30);                    %108-
        FILE DSK2 DISK SERIAL [20:450](2,30,30);                    %108-
FILE OUT CODISK DISK SERIAL [20:600] (2,30,300);
FILE OUT DISK DISK [1:2100] "MCP""DISK"(3,30,300,SAVE 99);
DEFINE    MCPTYPE              = 63#,
          DCINTYPE             = 62#,
          TSSINTYPE            = 61#;
COMMENT  FSPOL CODE FILES ARE UNIQUELY TYPED IN THEIR FILE
          HEADERS.   HEADER[4].[36:6] IS THE FIELD USED TO CONTAIN
          THE TYPE;
FILE OUT DECK 0 (2,10);
FILE STUFF DISK SERIAL[20:150](2,10,30,SAVE 15);
ARRAY TWXA[0:16];
 REAL C;
    COMMENT  C CONTAINS ACTUAL VALUE OF LAST CONSTANT SCANNED;
 REAL T;
    COMMENT  T IS A TEMPORARY CELL;
 INTEGER TCOUNT;
    REAL STACKCT;
    COMMENT  TCOUNT IS A VARIABLE WHICH HOLDS A PREVIOUS VALUE OF COUNT
              FOR THE USE OF CONVERT;
        DEFINE LASTSEQUENCE = 149#,                                 %115-
        LASTSEQROW = 2#;
```

```
REAL FOULED;

BOOLEAN
      FUNCTOG,     COMMENT TELLS WHETHER PROCEDURE BEING DECLARED IS A
                           FUNCTION;
      P2,          COMMENT GENERALY TELLS WHETHER OWN WAS SEEN;
      P3,          COMMENT TELLS WHETHER SAVE WAS SEEN;
      VONF,        COMMENT VALUE OR OWN FIELD OF ELBAT WORD;
      FORMALF,     COMMENT FORMAL FIELD OF ELBAT WORD;
      PTOG,        COMMENT TELLS THAT FORMAL PARAPART IS BEING PROCESSD;
SPECTOG,
      STOPENTRY,   COMMENT THIS MAKES THE ENTRY PROCEDURE ENTER ONLY
                           ONE ID AND THEN EIXT;
      AJUMP;       COMMENT TELLS WHETHER A JUMP IS HANGING;
```

```
01557000 T  0003:0045:3
01558000 T  0003:0050:0
01559010 T  0003:0057:2
01559020 T  0003:0059:3
01560000 T  0003:0059:3
01561000 T  0003:0066:1
01561056 T  0003:0070:1
01561085 C  0003:0074:0
01561087 C  0003:0078:1
01561300 T  0003:0082:1
01561400 T  0003:0087:2
01561410 T  0003:0094:0
01561420 T  0003:0094:0
01561430 T  0003:0094:0
01561440 T  0003:0094:0
01561450 T  0003:0094:0
01561460 T  0003:0094:0
01561500 T  0003:0094:0
01561600 T  0003:0098:0
01561700 T  0003:0105:2
01562000 T  0003:0107:3
01563000 T  0003:0107:3
01564000 T  0003:0107:3
01565000 T  0003:0107:3
01566000 T  0003:0107:3
01566010 T  0003:0107:3
01567000 T  0003:0107:3
01568000 T  0003:0107:3
01569000 P  0003:0107:3
01570000 T  0003:0107:3
01571000 T  0003:0107:3
01572000 T  0003:0107:3
01573000 T  0003:0107:3
01574000 T  0003:0107:3
01575000 T  0003:0107:3
01576000 T  0003:0107:3
01577000 T  0003:0107:3
01578000 T  0003:0107:3
01579000 T  0003:0107:3
01580000 T  0003:0107:3
01581000 T  0003:0107:3
01582000 T  0003:0107:3
01583000 T  0003:0107:3
01583100 T  0003:0107:3
01584000 T  0003:0107:3
01585000 T  0003:0107:3
01586000 T  0003:0107:3
01587000 T  0003:0107:3
01588000 T  0003:0107:3
01589000 T  0003:0107:3
01590000 T  0003:0107:3
01591000 T  0003:0107:3
01592000 T  0003:0107:3
01593000 T  0003:0107:3
01594000 T  0003:0107:3
01595000 T  0003:0107:3
01596000 T  0003:0107:3
```

```
BOOLEAN STOPDEFINE;
INTEGER MAXSAVE;
          COMMENT THIS CONTAINS THE SIZE OF THE MAXIMUM SAVE ARRAY
          DECLARED. IT IS USED TO HELP DETERMINE STORAGE REQUIREMENTS
          FOR THE PROGRAM PARAMETER BLOCK FOR THE OBJECT PROGRAM;
 REAL
      KLASSF,       COMMENT CLASS IN LOW ORDER 7 BITS;
      ADDRSF,       COMMENT ADDRESS IN LOW ORDER 11 BITS;
      LEVELF,       COMMENT LVL IN LOW ORDER 5 BITS;
      LINKF,        COMMENT LINK IN LOW ORDER 13 BITS;
      INCRF,        COMMENT INCR CN LOW ORDER 8 BITS;
      PROINFO,      COMMENT CONTAINS ELBAT WORD FOR PROCEDURE BEING
                    DECLARED;
      G,            COMMENT GLOBAL TEMPORARY FOR BLOCK;
      TYPEV,        COMMENT USED TO CARRY CLASS OF IDENTIFIER
                    BEING DECLARED;
    PROADD,           COMMENT CONTAINS ADDRESS OF PROCEDURE BEING
                    DECLARED;
      MARK ,        COMMENT CONTAINS INDEX INTO INFO WHERE FIRST WORD
                    OF ADDITIONAL INFO FOR A PROCEDURE ENTRY;
      PJ,           COMMENT FORMAL PARAMETER COUNTER;
       J,           COMMENT ARRAY COUNTER;
      LASTINFO,     COMMENT INDEX TO LAST ENTRY IN INFO;
      NEXTINFO,     COMMENT INDEX FOR NEXT ENTRYIN INFO;
      FIRSTX,       COMMENT RELATIVE ADD OF FIRST EXECUTABLE CODE
                    IN BLOCK,INITIALIZED TO 4095 EACH TIME;
       SAVEL;       COMMENT SAVE LOCATION FOR FIXUPS IN BLOCK;
INTEGER NCII;       COMMENT THIS CONTAINS THE COUNT OF CONSTANTS
                    ENTERED IN INFO AT ANY GIVEN TIME;
PROCEDURE UNHOOK;FORWARD;
PROCEDURE MAKEUPACCUM;FORWARD;
DEFINE PURPT=[4:8]#,SECRET=2#;
 COMMENT THESE DEFINES GIVE THE NAMES OF THE WORD MODE OPERATORS.  THE
    NUMBERS REFER TO THE APPROPRIATE SECTION OF THE PRODUCT SPECS.  THE
    FULL NAME IS ALSO GIVEN;
 DEFINE
   ADD =  16#,    COMMENT (0101) 7.4.2.1    ADD;
   BBC =  22#,    COMMENT (0131) 7.4.5.4    BRANCH BACKWARD CONDITIONAL;
   BBW = 534#,    COMMENT (4131) 7.4.5.2    BRANCH BACKWARD;
   BFC =  38#,    COMMENT (0231) 7.4.5.3    BRANCH FORWARD CONDITIONAL;
   BFW = 550#,    COMMENT (4231) 7.4.5.1    BRANCH FORWARD;
   CDC = 168#,    COMMENT (1241) 7.4.10.4   CONSTRUCT DESCRIPTOR CALL;
   CHS = 134#,    COMMENT (1031) 7.4.7.11   CHANGE SIGN;
   COC =  40#,    COMMENT (0241) 7.4.10.3   CONSTRUCT OPERAND CALL;
   COM = 130#,    COMMENT (1011) 7.4.10.5   COMMUNICATION OPERATOR;
   DEL =  10#,    COMMENT (0045) 7.4.9.3    DELETE;
   DUP = 261#,    COMMENT (2025) 7.4.9.2    DUPLICATE;
   EQL = 581#,    COMMENT (4425) 7.4.4.3    EQUAL;
   LBC = 278#,    COMMENT(2131) 7.4.5.9     GO BACKWARD CONDITIONAL;
   LBU = 790#,    COMMENT(6131) 7.4.5.7     GO BACKWARD (WORD);
   GEQ =  21#,    COMMENT (0125) 7.4.4.2    GREATER THAN OR EQUAL TO;
   LFC = 294#,    COMMENT(2231) 7.4.5.8     GO FORWARD CONDITIONAL;
    LFU = 806#,   COMMENT(6231) 7.4.5.6     GO FORWARD (WORD);
   GTR =  37#,    COMMENT (0225) 7.4.4.1    GREATER THAN;
   IDV = 384#,    COMMENT (3001) 7.4.2.5    INTEGER DIVIDE;
   INX =  24#,    COMMENT (0141) 7.4.10.2   INDEX;
   ISD = 532#,    COMMENT (4121) 7.4.6.3    INTEGER STORE DESTRUCTIVE;
```

```
01597000 T  0003:0107:3
01598000 T  0003:0107:3
01599000 T  0003:0107:3
01600000 T  0003:0107:3
01601000 T  0003:0107:3
01602000 T  0003:0107:3
01603000 T  0003:0107:3
01604000 T  0003:0107:3
01605000 T  0003:0107:3
01606000 T  0003:0107:3
01607000 T  0003:0107:3
01608000 T  0003:0107:3
01609000 T  0003:0107:3
01610000 T  0003:0107:3
01611000 T  0003:0107:3
01612000 T  0003:0107:3
01613000 T  0003:0107:3
01614000 T  0003:0107:3
01615000 T  0003:0107:3
01616000 T  0003:0107:3
01617000 T  0003:0107:3
01618000 T  0003:0107:3
01619000 T  0003:0107:3
01620000 T  0003:0107:3
01621000 T  0003:0107:3
01622000 T  0003:0107:3
01623000 T  0003:0107:3
01624000 T  0003:0107:3
01625000 T  0003:0107:3
01626000 T  0003:0107:3
01627000 T  0003:0112:0
01628000 T  0003:0112:0
01629000 T  0003:0112:0
01630000 T  0003:0112:0
01631000 T  0003:0112:0
01632000 T  0003:0112:0
01633000 T  0003:0112:0
01634000 T  0003:0112:0
01635000 T  0003:0112:0
01636000 T  0003:0112:0
01637000 T  0003:0112:0
01638000 T  0003:0112:0
01639000 T  0003:0112:0
01640000 T  0003:0112:0
01641000 T  0003:0112:0
01642000 T  0003:0112:0
01643000 T  0003:0112:0
01644000 T  0003:0112:0
01645000 T  0003:0112:0
01646000 T  0003:0112:0
01647000 T  0003:0112:0
01648000 T  0003:0112:0
01649000 T  0003:0112:0
01650000 T  0003:0112:0
01651000 T  0003:0112:0
01652000 T  0003:0112:0
01653000 T  0003:0112:0
```

```
    TSN = 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
INTEGER MAXROW;                                                                    01684000 T  0003:0112:0
            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
INTEGER F;                                                                         01688000 T  0003:0112:0
              REAL NLO,NHI,TLO,THI;                                                01689000 T  0003:0112:0
              BOOLEAN DPTOG;                                                       01690000 T  0003:0112:0
              COMMENT  THE ABOVE THINGS ARE TEMP STORAGE FOR DOUBLE NOS;           01691000 T  0003:0112:0
BOOLEAN DOLLAR2TOG;                                                                01691500 T  0003:0112:0
DEFINE FZERO=896#;                                                                 01692000 T  0003:0112:0
REAL T1,T2,N,K,AKKUM;                                                              01693000 T  0003:0112:0
BOOLEAN STOPGSP;                                                                   01694000 T  0003:0112:0
INTEGER BUP;                                                                       01695000 T  0003:0112:0
BOOLEAN INLINETOG;                                                                 01695500 T  0003:0112:0
            COMMENT UNIQUE GLOBAL TEMP FOR BLOCK;                                  01696000 T  0003:0112:0
ARRAY GTA1[0:10];                                                                  01697000 T  0003:0112:0
 BOOLEAN ARRAY SPRT[0:31];                                                         01698000 T  0003:0114:1
   COMMENT  SPRT IS TO BE CONSIDERED TO BE AN ARRAY OF 32 32 BIT                   01699000 T  0003:0117:2
              FIELDS.  THE 32 BITS ARE IN THE LOW ORDER PART OF EACH               01700000 T  0003:0117:2
              WORD.  THE BIT IS ON IF AND ONLY IF THE CORRESPONDING                01701000 T  0003:0117:2
              PRT CELL HAS A PERMANENT ASSIGNMENT;                                 01702000 T  0003:0117:2
 INTEGER PRTI,PRTIMAX;                                                             01703000 T  0003:0117:2
   COMMENT  PRTIMAX GIVES NEXT PRT CELL AVAILABLE FOR PERMANENT ASSIGN-            01704000 T  0003:0117:2
              MENT.  PRTI GIVES NEXT PRT CELL POSSIBLY AVAILABLE FOR               01705000 T  0003:0117:2
```

```
                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;
 ALPHA PROCEDURE B2D(B); VALUE B; REAL B; FORWARD;
 REAL PROCEDURE TAKE(W); VALUE W; INTEGER W; FORWARD;
 BOOLEAN MACROID;
 REAL PROCEDURE FIXDEFINEINFO(T); VALUE T; REAL T; FORWARD;
 PROCEDURE ERR (ERRNUM); VALUE ERRNUM; INTEGER ERRNUM; FORWARD;
 INTEGER PROCEDURE GIT(L); VALUE L; REAL L; FORWARD;
 ARRAY CALLA[0:31,0:255];
 DEFINE CALL[CALL1]=CALLA[(GT3-CALL1),LINKR,GT3,LINKC]#;
 REAL CALLX,CALLINFO,NESTCTR,NESTCUR;
 ARRAY NESTPRT[PRTBASE:PRTOP];
 ARRAY  SORTPRT[0:PRTOP-PRTBASE];

COMMENT  "BLANKET" BLANKS OUT N+1 WORDS IN "THERE";
STREAM PROCEDURE BLANKET(N,THERE); VALUE N;
    BEGIN
    DI:=THERE; DS:=8 LIT" "; SI:=THERE; DS:=N WDS;
    END BLANKET;
STREAM PROCEDURE CHANGESEQ(VAL,OLDSEQ); VALUE OLDSEQ;
        BEGIN  DI:=OLDSEQ; SI:=VAL; DS:=8 DEC  END CHANGESEQ;
STREAM PROCEDURE SEQUENCEERROR(L);
    BEGIN  DI:=L; DS:=16 LIT"SEQUENCE ERROR  ";  END SEQUENCEERROR;
STREAM PROCEDURE GETVOID(VP,NCR,LCR,SEQ);  VALUE NCR,LCR;
    BEGIN
    LABEL L,EXIT;
    LOCAL N;
    SI:=NCR; DI:=VP; DS:=8 LIT "0";
    2(34(IF SC=" " THEN SI:=SI+1 ELSE JUMP OUT 2 TO L));
    GO TO EXIT;                        % NO VOID RANGE GIVEN, RETURN ZERO.
L:
    IF SC="%" THEN GO TO EXIT;     % STILL NO RANGE.
    IF SC="'" THEN
         BEGIN
         SI:=SI+1; DI:=LCR; DS:=1 LIT"'"; % STOPPER FOR SCAN
         NCR:=SI; % TEMP. STORAGE, SINCE NCR IS "LOCAL" TO GETVOID.
         8(IF SC="'" THEN JUMP OUT ELSE
              BEGIN  TALLY:=TALLY+1; SI:=SI+1 END);
         END
    ELSE BEGIN
         NCR:=SI; % TEMP. STORAGE, SINCE NCR IS "LOCAL" TO GETVOID.
         DI:=LCR; DS:=1 LIT" "; % STOPPER FOR SCAN
         8(IF SC=" " THEN JUMP OUT ELSE
              BEGIN  TALLY:=TALLY+1; SI:=SI+1 END);
         END;
    SI:=NCR; DI:=VP; DI:=DI+8; % RESTORE POINTERS.
    N:=TALLY; DI:=DI-N; DS:=N CHR;
EXIT:
```

```
01706000 T  0003:0117:2
01707000 T  0003:0117:2
01708000 T  0003:0117:2
01709000 T  0003:0117:2
01710000 T  0003:0117:2
01711000 T  0003:0117:2
01712000 T  0003:0117:2
01713000 T  0003:0117:2
01714000 T  0003:0117:2
01715000 T  0003:0117:2
01716000 T  0003:0117:2
01717000 T  0003:0117:2
01717700 T  0003:0117:2
01717800 T  0003:0117:2
01717900 T  0003:0117:2
01718000 T  0003:0117:2
01719000 T  0003:0117:2
01720000 T  0003:0117:2
01721000 T  0003:0119:3
01722000 T  0003:0119:3
01724000 T  0003:0119:3
01725000 T  0003:0122:0
01726000 T  0003:0126:0
01737300 T  0003:0126:0
01737350 T  0003:0126:0
01737400 T  0003:0126:0
01737450 T  0003:0127:2
01737500 T  0003:0129:2
01741200 T  0003:0129:3
01741300 T  0003:0129:3
01742100 T  0003:0131:2
01742110 T  0003:0131:2
01756000 T  0003:0133:3
01757000 T  0003:0133:3
01758000 T  0003:0134:0
01759000 T  0003:0134:0
01761000 T  0003:0134:0
01762000 T  0003:0135:3
01763000 T  0003:0138:1
01764000 T  0003:0138:1
01764500 T  0003:0138:1
01765000 T  0003:0139:3
01766000 T  0003:0140:0
01767000 T  0003:0140:0
01768000 T  0003:0141:3
01769000 T  0003:0142:0
01770000 T  0003:0143:3
01771000 T  0003:0145:2
01772000 T  0003:0145:2
01773000 T  0003:0145:2
01774000 T  0003:0145:3
01775000 T  0003:0146:0
01776000 T  0003:0147:3
01777000 T  0003:0149:2
01780000 T  0003:0149:2
01781000 T  0003:0149:3
01782000 T  0003:0151:2
```

```
    END OF GETVOID;
REAL VOIDCR,VOIDPLACE,VOIDTCR,VOIDTPLACE;
FORMAT
        BUG(X24,4(A4,X2));


PROCEDURE DATIME;
    BEGIN
    INTEGER H,MIN,Q; ALPHA N1,N2;

    ALPHA STREAM PROCEDURE DATER(DATE); VALUE DATE;
        BEGIN
        DI:=LOC DATER; SI:=LOC DATE; SI:=SI+2;
        2(DS:=2 CHR; DS:=LIT"/"); DS:=2 CHR;
        END OF DATER;
    H:=TIME1 DIV 216000; MIN:=(TIME1 DIV 3600) MOD 60;
    N1:=DISK.MFID; N2:=DISK.FID;
    WRITE(LINE,
        <X22,"BURROUGHS B-5700 ESPOL COMPILER MARK ",
"XVI.0.116"
        ,"  ",A6,"DAY, ",0,",  ",I2,":",A2,X1,A3,
        ////X45,A1,A6,"/",A1,A6,/X45,15("=")//>,

        TIME(6),DATER(TIME(5)),12*REAL(Q:=H MOD 12=0)+Q,
        Q:=MIN MOD 10+(MIN DIV 10)*64,
        IF H≥12 THEN "PM," ELSE "AM,",
        N1,[6:6],N1,N2,[6:6],N2);
    IF MERGETOG THEN % INDICATE NAME OF SOURCE FILE.                    %111-
      WRITE(LINE,<X40,"SOURCE FILE: ",A1,A6,"/",A1,A6,//>,              %111-

        (N1:=TAPE.MFID),[6:6],N1,(N2:=TAPE.FID),[6:6],N2);              %111-
    NOHEADING:=FALSE;
    END OF DATIME;

 COMMENT  THIS SECTION CONTAINS ALL CODE PERTAINENT TO READING CARDS
                AND SCANNING THEM;
%*********************************************************************
%                                                                     %110-
%       MISCELLANEOUS CROSS REFERENCE PROCEDURES                      %110-
%                                                                     %110-
%*********************************************************************
%                                                                     %110-
PROCEDURE CROSSREFIT(INDEX,SEQNO,REFTYPE);                            %110-
  VALUE INDEX,SEQNO,REFTYPE;                                          %110-
  REAL  INDEX,SEQNO,REFTYPE;                                          %110-
BEGIN                                                                 %110-
 IF XREFINFO[INDEX].IDNOF ≠ 0 THEN % SAVE                             %110-
 BEGIN                                                                %110-
  IF XREFPT > 29 THEN % NO SLOTS LEFT IN ARRAY. WRITE IT OUT.         %110-
    BEGIN                                                             %110-
      WRITE(DSK2,30,XREFAY2[*]);                                      %110-
      XREFPT := 0;                                                    %110-
    END;                                                              %110-
  XREFAY2[XREFPT] := SEQNO & REFTYPE TYPEREF & XREFINFO[INDEX]        %110-
                                        REFIDNOF;                     %110-
  XREFPT := XREFPT + 1; % EVEN THOUGH THE ARRAY MAY BE FULL NOW WE    %110-
                        % CANT WRITE IT OUT BECAUSE SOME ROUTINES     %110-
```

```
                01784000 T  0003:0151:2
                01785000 T  0003:0151:2
                01800000 T  0003:0151:2
                01802000 T  0003:0151:2
          START OF SEGMENT **********    4
            4 IS    8 LONG, NEXT SEG    3
                01820000 T  0003:0151:2
                01821000 T  0003:0151:2
                01822000 T  0003:0151:2
          START OF SEGMENT **********    5
                01823000 T  0005:0000:0
                01824000 T  0005:0000:0
                01825000 T  0005:0000:0
                01826000 T  0005:0000:1
                01827000 T  0005:0002:0
                01828000 T  0005:0003:2
                01828500 T  0005:0007:2
                01829000 T  0005:0013:3
                01830000 T  0005:0015:2
%117-           01831000 P  0006:0015:2
                01832000 T  0006:0015:2
                01832500 T  0006:0015:2
            6 IS   40 LONG, NEXT SEG    5
                01833000 T  0005:0016:0
                01834000 T  0005:0032:0
                01835000 T  0005:0036:1
                01835500 T  0005:0043:2
                01835600 C  0005:0053:2
                01835700 C  0005:0054:0
            7 IS   14 LONG, NEXT SEG    5
                01835800 C  0005:0057:2
                01836000 T  0005:0077:2
                01837000 T  0005:0078:0
            5 IS   82 LONG, NEXT SEG    3
                02000000 T  0003:0151:2
                02001000 T  0003:0151:2
                02001605 C  0003:0151:2
                02001610 C  0003:0151:2
                02001615 C  0003:0151:2
                02001620 C  0003:0151:2
                02001630 C  0003:0151:2
                02001635 C  0003:0151:2
                02001640 C  0003:0151:2
                02001645 C  0003:0151:2
                02001650 C  0003:0151:2
                02001655 C  0003:0151:2
                02001660 C  0003:0151:2
                02001665 C  0003:0155:3
                02001670 C  0003:0156:0
                02001675 C  0003:0157:2
                02001680 C  0003:0157:3
                02001685 C  0003:0161:3
                02001690 C  0003:0162:1
                02001695 C  0003:0162:1
                02001700 C  0003:0166:1
                02001705 C  0003:0168:0
                02001710 C  0003:0169:3
```

```
                                % WILL LOOK BACK AT THE ENTRY WE JUST PUT
                                % IN AND FIX IT UP.
 END;
END OF CROSSREFIT;
%
PROCEDURE CROSSREFDUMP(INDEX);
  VALUE INDEX;
  REAL INDEX;
BEGIN
  STREAM PROCEDURE MOVEXREFINFO(S,D,N);

    VALUE N;
  BEGIN
    SI := D; DI := D; DS := 8 LIT " "; DS := 7 WDS; % BLANK RECORD
    SI := S; SI := SI + 3; DI := D; DS := N CHR; % MOVE IDENTIFIER
  END OF MOVEXREFINFO;
  %
 IF XREFINFO[INDEX].IDNOF ≠ 0 THEN % DUMP IT
 BEGIN
  MOVEXREFINFO(INFO[INDEX,LINKR,INDEX,LINKC+1],XREFAY1[*],
                TAKE(INDEX+1),[12:6]);
  XREFAY1[8] := XREFINFO[INDEX];
  XREFAY1[9] := TAKE(INDEX); % FLBAT WORD
  WRITE(DSK1,10,XREFAY1[*]);
  XREFINFO[INDEX] := 0;
 END;
END OF CROSSREFDUMP;

COMMENT  OCTIZE REFORMATS ACCUM FOR OCTAL CONSTANTS;
BOOLEAN STREAM PROCEDURE OCTIZE(S,D,SKP,CNT); VALUE SKP,CNT;
    BEGIN
    SI:=S; SI:=SI+4; DI:=D; SKP(DS:=3 RESET); % RIGHT JUSTIFY.
    CNT(IF SC≥"8" THEN TALLY:=1 ELSE IF SC<"0" THEN TALLY:=1; SKIP 3 SB;
        3(IF SB THEN DS:=SET FLSE DS:=RESET; SKIP SB));
    SI:=D; IF SB THEN
         BEGIN  TALLY:=1; DI:=D; DS:=RESET   END; % PREVENT FLAG BIT.
    OCTIZE:=TALLY; % "1" = NON OCTAL CHARACTER OR FLAG BIT.
    END OCTIZE;
COMMENT  HEXIZE REFORMATS ACCUM FOR HEXADECIMAL CONSTANTS;
BOOLEAN STREAM PROCEDURE HEXIZE(S,D,SKP,CNT); VALUE SKP,CNT;
    BEGIN  LOCAL T1,T2,TEMP2,TEMP1; LABEL AGIN;
COMMENT  LOCAL VARIABLES ARE LOCATED IN REVERSE ORDER FROM THE
         WAY THEY ARE DECLARED IN STREAM PROCEDURES;
    DI:=LOC TEMP1; CNT(DS:=LIT"1"); % IN CASE A CHAR=A,B,C,D,OR F.
    SI:=S; SI:=SI+3; DI:=LOC TEMP1; % WE MAY OVERFLOW INTO TEMP2.
    CNT(IF SC<"0" THEN IF SC≥"A" THEN IF SC≤"F" THEN % WORK HARD.
         BEGIN
         T1:=SI; T2:=DI; DI:=T1; SI:=T2; % FLIP, MAN.
         DS:=3 RESET; SI:=T1; DI:=T2; % FLIP BACK.
         DS:=1 ADD; DI:=DI-1; SKIP 2 DB; DS:=1 SET; SKIP 3 DB;
         GO AGIN;
         END;
         IF SC<"0" THEN TALLY:=1; DS:=CHR; % < 0 = NON-HEX CHARACTER.
AGIN:
     );
    SI:=LOC TEMP1; DI:=D; SKP(DS:=4 RESET); % RIGHT ADJUST CONSTANT.
    CNT(SKIP 2 SB;
```

```
%110-          02001715 C  0003:0169:3
%110-          02001720 C  0003:0169:3
%110-          02001725 C  0003:0169:3
%110-          02001730 C  0003:0169:3
%110-          02001735 C  0003:0169:3
%110-          02001740 C  0003:0169:3
%110-          02001745 C  0003:0169:3
%110-          02001750 C  0003:0169:3
%110-          02001755 C  0003:0169:3
%110-          02001760 C  0003:0169:3
                    START OF SEGMENT **********    8
%110-          02001765 C  0008:0000:0
%110-          02001770 C  0008:0000:0
%110-          02001775 C  0008:0000:0
%110-          02001780 C  0008:0002:0
%110-          02001785 C  0008:0003:2
%110-          02001790 C  0008:0003:3
%110-          02001795 C  0008:0003:3
%110-          02001800 C  0008:0007:3
%110-          02001805 C  0008:0008:0
%110-          02001810 C  0008:0011:3
%110-          02001815 C  0008:0013:3
%110-          02001820 C  0008:0017:3
%110-          02001821 C  0008:0019:2
%110-          02001822 C  0008:0023:3
%110-          02001825 C  0008:0027:2
%110-          02001830 C  0008:0027:2
                      8 IS   28 LONG, NEXT SEG    3
               02001836 T  0003:0169:3
               02001838 T  0003:0169:3
               02001840 T  0003:0169:3
               02001842 T  0003:0170:0
               02001844 T  0003:0172:0
               02001846 T  0003:0175:2
               02001848 T  0003:0177:3
               02001850 T  0003:0178:0
               02001852 T  0003:0179:3
               02001854 T  0003:0179:3
               02001856 T  0003:0180:1
               02001858 T  0003:0180:1
               02001860 T  0003:0180:1
               02001862 T  0003:0181:2
               02001864 T  0003:0181:2
               02001866 T  0003:0181:2
               02001868 T  0003:0183:2
               02001870 T  0003:0183:3
               02001872 T  0003:0187:2
               02001874 T  0003:0187:2
               02001876 T  0003:0188:1
               02001878 T  0003:0189:3
               02001880 T  0003:0190:1
               02001882 T  0003:0191:2
               02001884 T  0003:0191:2
               02001886 T  0003:0192:0
               02001888 T  0003:0192:0
               02001890 T  0003:0192:0
               02001892 T  0003:0194:0
```

```
      4(IF SB THEN DS:=SET ELSE DS:=RESET; SKIP SB));% FINAL CONVERT.
    SI:=D; IF SB THEN
        BEGIN  TALLY:=1; DI:=D; DS:=RESET   END; % PREVENT FLAG BIT.
    HEXIZE:=TALLY; % "1" IF PROGRAMMER GOOFED.
    END HEXIZE;
COMMENT  PUTSEQNO PUTS THE SEQUENCE NUMBER OF THE CARD-IMAGE
         CURRENTLY BEING SCANNED INTO THE INFO TABLE IN CASE
         IT IS NEEDED FOR FUTURE REFERENCE;
STREAM PROCEDURE PUTSEQNO(INFO,LCR); VALUE LCR;
    BEGIN  DI:=INFO; SI:=LCR; DS:=WDS;  END PUTSEQNO;
COMMENT  TURNONSTOPLIGHT TURNS THE LIGHT "RED" ON THE "CORNER".
         I.E., THE PURPOSE OF THIS ROUTINE IS TO INSERT A PER-
         CENT SIGN IN COLUMN 73 AS AN END OF CARD SENTINEL FOR
         THE SCANNER;
STREAM PROCEDURE TURNONSTOPLIGHT(RED,CORNER); VALUE RED,CORNER;
    BEGIN  DI:=CORNER; SI:=LOC CORNER; SI:=SI-1; DS:=CHR  END;
    COMMENT WRITNEW TRANSFERS THE CARD IMAGE TO THE NEWTAPE BUFFER
              AND REPORTS IF THE CARD MIGHT BE CONTROL CARD;
 BOOLEAN STREAM PROCEDURE WRITNEW(NEW,FCR); VALUE FCR;
         BEGIN SI ← FCR; IF SC ≠ "$" THEN TALLY ← 1;
         DI←NEW;DS←10 WDS;
              WRITNEW ← TALLY END WRITNEW;
    COMMENT MKABS CONVERTS A DESCRIPTOR  TO AN ABSOLUTE ADDRESS;
 REAL STREAM PROCEDURE MKABS(A);
              BEGIN DI ← A; MKABS ← DI END MKABS;
REAL STREAM PROCEDURE CONV(ACCUM,SKP,N);VALUE SKP,N;
              BEGIN
                   SI← ACCUM; SI←SI+SKP;SI←SI+3;DI←LOC CONV;DS←N OCT
              END;
 STREAM PROCEDURE  MOVECHARACTERS(N,SORCE,SSKIP,DEST,DSKIP);
                   VALUE  N,SSKIP,DSKIP;
                   BEGIN
                   SI←SORCE ; DI←DEST;
                   SI←SI+SSKIP; DI← DI+DSKIP ;
                   DS ← N CHR ;
                   END ;
 COMMENT  MOVECHARACTERS MOVES  N CHARACTERS FROM THE SSKIP-TH CHAR IN
             "SORCE" TO THE DSKIP-TH CHAR IN "DEST".          ;
 STREAM PROCEDURE MOVE(W)"WORDS FROM"(A)"TO"(B); VALUE W;
              BEGIN SI ← A; DI ← B; DS ← W WDS END;
 STREAM PROCEDURE RESIZE(FIEL);
 BEGIN LOCAL T;
  SI←FIEL; DI←LOC T; DS←WDS;
 SI←T;DI←FIEL;DI←DI+1; SKIP 2 DB; DS←10 SET
 END;
COMMENT  EQUAL COMPARES COUNT CHARACTERS LOCATED AT A AND B FOR
         EQUALITY.  THIS ROUTINE IS USED IN THE LOOK-UP OF ALPHA
         QUANTITIES IN THE DIRECTORY;
BOOLEAN STREAM PROCEDURE EQUAL(COUNT,A,B); VALUE COUNT;
    BEGIN
    TALLY:=1; SI:=A; DI:=B;
    IF COUNT SC=DC THEN EQUAL:=TALLY
    END EQUAL;
PROCEDURE READACARD;  FORWARD;
PROCEDURE DOLLARCARD; FORWARD;
BOOLEAN PROCEDURE BOOLEXP; FORWARD;
PROCEDURE SCANNER;
```

```
02001894 T  0003:0195:3
02001895 T  0003:0197:3
02001896 T  0003:0198:1
02001897 T  0003:0199:3
02001898 T  0003:0200:0
02002000 T  0003:0201:2
02003000 T  0003:0201:2
02004000 T  0003:0201:2
02005000 T  0003:0201:2
02006000 T  0003:0201:2
02007000 T  0003:0202:0
02008000 T  0003:0202:0
02009000 T  0003:0202:0
02010000 T  0003:0202:0
02011000 T  0003:0202:0
02012000 T  0003:0202:0
02014000 T  0003:0203:2
02015000 T  0003:0203:2
02016000 T  0003:0203:2
02017000 T  0003:0203:2
02018000 T  0003:0205:2
02020000 T  0003:0205:3
02021000 T  0003:0206:1
02022000 T  0003:0206:1
02023000 T  0003:0206:1
02041000 T  0003:0208:1
02042000 T  0003:0208:1
02043000 T  0003:0209:2
02044000 T  0003:0210:1
02045000 T  0003:0211:3
02046000 T  0003:0211:3
02047000 T  0003:0211:3
02048000 T  0003:0212:0
02049000 T  0003:0212:1
02050000 T  0003:0213:3
02051000 T  0003:0214:0
02052000 T  0003:0214:0
02053000 T  0003:0214:0
02054000 T  0003:0214:0
02055000 T  0003:0214:0
02056000 T  0003:0216:0
02057000 T  0003:0216:0
02058000 T  0003:0217:2
02059000 T  0003:0217:3
02060000 T  0003:0219:2
02061000 T  0003:0219:2
02061500 T  0003:0219:2
02062000 T  0003:0219:2
02062500 T  0003:0219:2
02063000 T  0003:0219:2
02063500 T  0003:0220:0
02064000 T  0003:0220:1
02064500 T  0003:0221:3
02065000 T  0003:0222:1
02065500 T  0003:0222:1
02065600 T  0003:0222:1
02066000 T  0003:0222:1
```

```
    BEGIN                                                                  02066500 T  0003:0222:1
COMMENT  "SCAN" IS THE STREAM PROCEDURE WHICH DOES THE ACTUAL SCANNING,    02067000 T  0003:0222:1
         IT IS DRIVEN BY A SMALL WORD MODE PROCEDURE CALLED "SCANNER",     02067500 T  0003:0222:1
         WHICH CHECKS FOR A QUANTITY BEING BROKEN ACROSS A CARD. "SCAN"    02068000 T  0003:0222:1
         IS CONTROLLED BY A VARIABLE CALLED "RESULT". "SCAN" ALSO          02068500 T  0003:0222:1
         INFORMS THE WORLD OF ITS ACTION BY MEANS OF THE SAME VARIABLE.    02069000 T  0003:0222:1
         HENCE THE VARIABLE "RESULT" IS PASSED BY BOTH NAME AND VALUE.     02069500 T  0003:0222:1
         THE MEANING OF "RESULT" AS INPUT IS:                              02070000 T  0003:0222:1
            VALUE    MEANING                                               02070500 T  0003:0222:1
            =====    =======================================               02071000 T  0003:0222:1
              0      INITIAL CODE - DEBLANK AND START TO FETCH THE         02071500 T  0003:0222:1
                     NEXT QUANTITY.                                        02072000 T  0003:0222:1
              1      CONTINUE BUILDING AN IDENTIFIER (INTERRUPTED BY       02072500 T  0003:0222:1
                     END-OF-CARD BREAK).                                   02073000 T  0003:0222:1
              2      LAST QUANTITY BUILT WAS SPECIAL CHARACTER. HENCE,     02073500 T  0003:0222:1
                     EXIT (INTERRUPTION BY END-OF-CARD BREAK IS NOT        02074000 T  0003:0222:1
                     IMPORTANT).                                           02074500 T  0003:0222:1
              3      CONTINUE BUILDING A NUMBER (INTERRUPTED BY END-OF-     02075000 T  0003:0222:1
                     CARD BREAK).                                          02075500 T  0003:0222:1
              4      LAST THING WAS AN ERROR (COUNT EXCEEDED 63). HENCE,   02076000 T  0003:0222:1
                     EXIT (INTERRUPTION BY END-OF-CARD BREAK NOT           02076500 T  0003:0222:1
                     IMPORTANT).                                           02077000 T  0003:0222:1
              5      GET NEXT CHARACTER AND EXIT.                          02077500 T  0003:0222:1
              6      SCAN A COMMENT.                                       02078000 T  0003:0222:1
              7      DEBLANK ONLY.                                         02078500 T  0003:0222:1
         THE MEANING OF "RESULT" AS OUTPUT IS:                             02079000 T  0003:0222:1
            VALUE    MEANING                                               02079500 T  0003:0222:1
            =====    =======================================               02080000 T  0003:0222:1
              1      AN IDENTIFIER WAS BUILT.                              02080500 T  0003:0222:1
              2      A SPECIAL CHARACTER WAS OBTAINED.                     02081000 T  0003:0222:1
              3      A NUMBER (INTEGER) WAS BUILT.                         02081500 T  0003:0222:1
         "SCAN" PUTS ALL STUFF SCANNED (EXCEPT FOR COMMENTS AND            02082000 T  0003:0222:1
         DISCARDED BLANKS) INTO "ACCUM" (CALLED "ACCUMULATOR"              02082500 T  0003:0222:1
         FOR THE REST OF THIS DISCUSSION).                                 02083000 T  0003:0222:1
         "COUNT" IS THE VARIABLE THAT GIVES THE NUMBER OF CHARACTERS       02083500 T  0003:0222:1
         "SCAN" HAS PUT INTO THE "ACCUMULATOR". SINCE "SCAN" NEEDS         02084000 T  0003:0222:1
         THE VALUE SO THAT IT CAN PUT MORE CHARACTERS INTO THE "ACCUM-     02084500 T  0003:0222:1
         ULATOR" AND NEEDS TO UPDATE "COUNT" FOR THE OUTSIDE WORLD,        02085000 T  0003:0222:1
         "COUNT" IS PASSED BY BOTH NAME AND VALUE.  IT IS ALSO             02085500 T  0003:0222:1
         CONVENIENT TO HAVE (63-COUNT), THIS IS CALLED "COMCOUNT".         02086000 T  0003:0222:1
         "NCR" (NEXT CHARACTER TO BE SCANNED) IS ALSO PASSED BY            02086500 T  0003:0222:1
         NAME AND VALUE SO THAT IT MAY BE UPDATED.                         02087000 T  0003:0222:1
         "ST1" AND "ST2" ARE TEMPORARY STORAGES WHICH ARE EXPLICITLY       02087500 T  0003:0222:1
         PASSED TO "SCAN" IN ORDER TO OBTAIN THE MOST USEFULL STACK        02088000 T  0003:0222:1
         ARRANGEMENT.                                                      02088500 T  0003:0222:1
         ;                                                                 02089000 T  0003:0222:1
      STREAM PROCEDURE SCAN(NCR,COUNTV,ACCUM,COMCOUNT,RESULT,RESULTV,      02089500 T  0003:0222:1
                                                                  START OF SEGMENT **********    9
                           COUNT,ST2,NCRV,ST1);                           02090000 T  0009:0000:0
                       VALUE COUNTV, COMCOUNT,RESULTV,ST2,NCRV,ST1;       02090500 T  0009:0000:0
         BEGIN                                                             02091000 T  0009:0000:0
         LABEL DEBLANK,NUMBERS,IDBLDR,GNC,K,EXIT,FINIS,L,ERROR,            02091500 T  0009:0000:0
               COMMENTS,COMMANTS;                                          02092000 T  0009:0000:0
         DI:=RESULT; DI:=DI+7; SI:=NCRV;                                   02092500 T  0009:0000:0
COMMENT  SETUP "DI" FOR A CHANGE IN "RESULT" AND "SI" FOR A LOOK AT        02093000 T  0009:0000:1
         THE BUFFER;                                                       02093500 T  0009:0000:1
         CI:=CI+RESULTV;    % SWITCH ON VALUE OF RESULT;                   02094000 T  0009:0000:1
```

```
                    GO DEBLANK;   % 0 IS INITIAL CODE.                                  02094500 T  0009:0001:2
                    GO IDBLDR;    % 1 IS ID CODE.                                       02095000 T  0009:0001:3
                    GO FINIS;     % 2 IS SPECIAL CHARACTER CODE.                        02095500 T  0009:0001:3
                    GO NUMBERS;   % 3 IS NUMBER CODE.                                   02096000 T  0009:0002:0
                    GO FINIS;     % 4 IS ERROR CODE.                                    02096500 T  0009:0002:0
                    GO GNC;       % 5 IS GET NEXT CHARACTER CODE.                       02097000 T  0009:0002:1
                    GO COMMANTS;  % 6 IS COMMENT CODE.                                  02097500 T  0009:0002:1
                                  % 7 IS DEBLANK ONLY CODE.                            02098000 T  0009:0003:2
                    IF SC=" " THEN                                                     02098500 T  0009:0003:2
K:                      BEGIN  SI:=SI+1; IF SC=" " THEN GO K  END;                     02099000 T  0009:0003:3
                    GO FINIS;                                                          02099500 T  0009:0005:2
DEBLANK;                                                                               02100000 T  0009:0005:2
                    IF SC=" " THEN                                                     02100500 T  0009:0005:2
L:                      BEGIN  SI:=SI+1; IF SC=" " THEN GO L  END;                     02101000 T  0009:0006:1
COMMENT             IF WE ARRIVE HERE WE HAVE A NON-BLANK CHARACTER;                   02101500 T  0009:0008:0
                    NCRV:=SI;                                                          02102000 T  0009:0008:0
                    IF SC ≥ "0" THEN GO NUMBERS;                                       02102500 T  0009:0008:0
                    IF SC=ALPHA THEN GO IDBLDR;                                        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
                    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
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  0009:0032:0
         RESULT,COUNT,0,NCR,0);                                                        02129000 T  0009:0034:1
    IF NCR=LCR THEN                                                                    02129500 T  0009:0036:0
         BEGIN                                                                         02130000 T  0009:0037:2
         READACARD;                                                                    02130500 T  0009:0037:3
         GO TO L; % GO DIRECTLY TO L, DO NOT PASS GO,                                  02135500 T  0009:0038:0
                  % DO NOT COLLECT $200.                                               02136000 T  0009:0038:1
         END;                                                                          02136500 T  0009:0038:1
    END SCANNER;                                                                       02137000 T  0009:0038:1
                                                                                 9 IS   39 LONG, NEXT SEG    3
DEFINE WRITELINE = IF SINGLTOG THEN WRITE(LINE,15,LIN[*])                              02181000 T  0003:0222:1
                    ELSE WRITE(LINE[DBL],15,LIN[*])#,                                  02181250 T  0003:0222:1
       PRINTCARD = BEGIN                                                               02182500 T  0003:0222:1
                   EDITLINE(LIN,FCR,L DIV 4,L.[46:2],MEDIUM,OMITTING);                 02182750 T  0003:0222:1
                   IF NOHEADING THEN DATIME; WRITELINE;                                02183000 T  0003:0222:1
                   END #;                                                              02183250 T  0003:0222:1
STREAM PROCEDURE EDITLINE(LINE,NCR,R,L,SYMBOL,OMIT);                                   02183500 T  0003:0222:1
                 VALUE  NCR,R,L,SYMBOL,OMIT;                                           02183750 T  0003:0222:1
    BEGIN                                                                              02184000 T  0003:0222:1
    DI := LINE; DS := 16 LIT " ";                                                      02184250 T  0003:0223:2
    SI := NCR; DS := 9 WDS;                                                            02184500 T  0003:0225:3
    DS := 8 LIT " ";                                                                   02184750 T  0003:0226:0
    DS := WDS; % SEQUENCE NUMBER.                                                      02185000 T  0003:0227:2
    DS:=LIT" "; SI:=LOC SYMBOL; SI:=SI+6;                                              02185250 T  0003:0227:3
    DS:=2 CHR; DS:=LIT" ";                                                             02185500 T  0003:0228:1
    SI←LOC R;  DS←4 DEC;  DS←LIT ":";                                                  02185750 T  0003:0229:2
    SI←LOC L;  DS←1 DEC;                                                               02186000 T  0003:0230:0
    DS←6 LIT " ";                                                                      02186250 T  0003:0230:1
    OMIT(DI:=DI-12; DS:=8 LIT"    OMIT");                                              02186750 T  0003:0231:3
    END EDITLINE;                                                                      02187000 T  0003:0234:1
COMMENT  COMPARE COMPARES SEQUENCE NUMBERS OF TAPE AND CARD. IF                        02187250 T  0003:0234:1
         TAPE IS SMALLER THEN RESULT = 0 ELSE IF CARD IS SMALLER                       02187500 T  0003:0234:1
         RESULT = 1 ELSE RESULT = 2;                                                   02187750 T  0003:0234:1
REAL STREAM PROCEDURE COMPARE(TAPE,CARD); VALUE TAPE,CARD;                             02188000 T  0003:0234:1
    BEGIN                                                                              02188250 T  0003:0234:1
    SI := TAPE; DI := CARD;                                                            02188500 T  0003:0235:2
    IF 8 SC ≥ DC THEN                                                                  02188750 T  0003:0235:3
         BEGIN                                                                         02189000 T  0003:0236:0
         SI := SI-8; DI := DI-8; TALLY := 1;                                           02189250 T  0003:0236:0
         IF 8 SC = DC THEN TALLY := 2                                                  02189500 T  0003:0237:2
         END;                                                                          02189750 T  0003:0237:3
    COMPARE := TALLY                                                                   02190000 T  0003:0238:0
    END COMPARE;                                                                       02190250 T  0003:0238:0
PROCEDURE OUTPUTSOURCE;                                                                02190500 T  0003:0239:2
    BEGIN                                                                              02190750 T  0003:0239:2
    LABEL LCARD,LTAPE,AWAY;                                                            02191000 T  0003:0239:2
```

```
    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;

    PROCEDURE BEGINPRINT;
   BEGIN
         STREAM PROCEDURE STUFF(N,L); VALUE N;
         BEGIN
              DI:=L; DS:=8 LIT " "; SI:=L; DS:=13 WDS;
              SI:=LOC N; DS:=8 DEC;
         END;
          STUFF(BEGINSTACK[BSPOINT],LIN);
          IF NOHEADING THEN DATIME; WRITELINE;
   END BEGINPRINT;

PROCEDURE READACARD;
COMMENT  READACARD READS CARDS FROM EITHER THE CARD READER OR THE
         TAPE MERGING AS REQUESTED AND CREATING A NEW TAPE AND
         LISTING IF REQUESTED.  READACARD ALSO INSERTS A PERCENT
         SIGN AS AN END OF CARD SENTINEL IN COLUMN 73 AND SETS
         FCR,NCR,LCR,TLCR, AND CLCR;
    BEGIN
    PROCEDURE READTAPE;

              BEGIN
LABEL ENDREADTAPE, EOFT;

READ (TAPE, 10, TBUFF[*])[EOFT];
                  LCR:=MKABS(TBUFF[9]);
GO TO ENDREADTAPE;
EOFT:
DEFINEARRAY[25]:="ND;END."& "E"[1:43:5];
DEFINEARRAY[34]:="9999" & "9999"[1:25:23];
TLCR:= MKABS(DEFINEARRAY[34]);
PUTSEQNO (DEFINEARRAY[33],TLCR-8);
TURNONSTOPLIGHT("%", TLCR-8);
```

```
                START OF SEGMENT **********   10
                     02191250 T  0010:0000:0
                     02191500 T  0010:0006:0
                     02191750 T  0010:0007:3
                     02192000 T  0010:0008:0
                     02192250 T  0010:0009:3
                     02192500 T  0010:0010:0
                     02192750 T  0010:0011:2
                     02193000 T  0010:0012:0
                     02193250 T  0010:0014:1
                     02193500 T  0010:0015:3
                     02193750 T  0010:0015:3
                     02194000 T  0010:0016:0
                     02194250 T  0010:0023:2
                     02194500 T  0010:0025:3
                     02194750 T  0010:0027:3
                     02195000 T  0010:0028:0
                     02195250 T  0010:0045:2
                     02195500 T  0010:0045:3
                     02195750 T  0010:0046:0
                     02196000 T  0010:0062:0
                     02196250 T  0010:0062:0
                     02196500 T  0010:0062:0
          10 IS   63 LONG, NEXT SEG    3
%108-                02196510 C  0003:0239:2
%108-                02196520 C  0003:0239:2
%108-                02196530 C  0003:0239:2
     START OF SEGMENT **********   11
%108-                02196540 C  0011:0000:0
%108-                02196550 C  0011:0000:0
%108-                02196560 C  0011:0002:0
%108-                02196570 C  0011:0002:1
%108-                02196580 C  0011:0002:1
%108-                02196590 C  0011:0004:1
%108-                02196610 C  0011:0016:0
        11 IS   17 LONG, NEXT SEG    3
                     02196750 T  0003:0239:2
                     02197000 T  0003:0239:2
                     02197250 T  0003:0239:2
                     02197500 T  0003:0239:2
                     02197750 T  0003:0239:2
                     02198000 T  0003:0239:2
                     02198250 T  0003:0239:2
                     02198500 T  0003:0239:2
     START OF SEGMENT **********   12
                     02201500 T  0012:0000:0
%105-                02201510 C  0012:0000:0
     START OF SEGMENT **********   13
%105-                02201750 P  0013:0000:0
%105-                02202000 P  0013:0005:2
%105-                02202010 C  0013:0007:2
%105-                02202020 C  0013:0007:3
%105-                02202030 C  0013:0008:0
%105-                02202040 C  0013:0010:0
%105-                02202050 C  0013:0012:1
%105-                02202060 C  0013:0014:1
%105-                02202070 C  0013:0016:0
```

```
ENDREADTAPE:                                                                %105-
            END READTAPE;

    PROCEDURE SEQCOMPARE(TLCR,CLCR, LIB); VALUE LIB; BOOLEAN LIB;
                                          REAL TLCR, CLCR ;
        BEGIN
        MEDIUM:="C "; % CARD READER.
        IF GT1:=COMPARE(TLCR,CLCR)=0 THEN  % TAPE HAS LOW SEQUENCE NUMB
            BEGIN
            LCR:=TLCR; LASTUSED:=3;
            MEDIUM:="T "; % TAPE INPUT.
            END
        ELSE BEGIN
            IF GT1 ≠ 1 THEN % TAPE AND CARD HAVE SAME SEQ
                BEGIN
                MEDIUM:="P "; % CARD PATCHES TAPE.
                READTAPE;
                END;
            LCR:=CLCR;
            LASTUSED:=2;
            END;
        END OF SEQCOMPARE;
    LABEL CARDONLY, CARDLAST, TAPELAST,  EXIT, FIRSTTIME,
          EOF, USETHESWITCH,
          COMPAR, TESTVOID, XIT;
    SWITCH USESWITCH:=CARDONLY,CARDLAST,TAPELAST,FIRSTTIME;
    IF ERRORCOUNT≥ERRMAX THEN ERR(611); % ERR LIMIT EXCEEDED - STOP.
USETHESWITCH:
    DOLLAR?TOG:=FALSE;
    GO TO USESWITCH[LASTUSED];
    MOVE(1,INFO[LASTUSED,LINKR,LASTUSED,LINKC],
         DEFINEARRAY[DEFINEINDEX-2]);
    LASTUSED := LASTUSED + 1;
    NCR := LCR-1;
    GO TO XIT;
FIRSTTIME:
    READ(CARD,10,CBUFF[*]);
    FCR:=NCR:=(LCR:=MKABS(CBUFF[9]))-9;
    MEDIUM:="C ";
    IF EXAMIN(FCR)≠"$" AND LISTER THEN PRINTCARD;
    PUTSEQNO(INFO[LASTSEQROW,LASTSEQUENCE],LCR);
    CARDNUMBER:=CONV(INFO[LASTSEQROW,LASTSEQUENCE-1],5,8);          %108-
    TURNONSTOPLIGHT("%",LCR);
    GO XIT;
COMMENT  WE HAVE JUST INITIALIZED CARD INPUT;
CARDONLY:
    READ(CARD,10,CBUFF[*]);
    LCR := MKABS(CBUFF[9]); GO EXIT;
CARDLAST:
    READ(CARD,10,CBUFF[*])[EOF];
    CLCR := MKABS(CBUFF[9]);
    GO COMPAR;
EOF:
    DEFINEARRAY[25]:="NO;END,"&"E"[1:43:5];
    DEFINEARRAY[34]:="9999"&"9999"[1:25:23];
    CLCR:=MKABS(DEFINEARRAY[34]);
    PUTSEQNO(DEFINEARRAY[33],CLCR-8);
```

```
                                                                    02202080 C  0013:0017:3
                                                                    02202250 T  0013:0018:0
                                                          13 IS   24 LONG, NEXT SEG   12
                                                                    02202500 T  0012:0000:0
                                                                    02202750 T  0012:0000:0
                                                                    02203000 T  0012:0000:0
                                                                    02203250 T  0012:0000:0
                                                                    02203500 T  0012:0000:1
                                                                    02203750 T  0012:0003:2
                                                                    02204000 T  0012:0003:3
                                                                    02204250 T  0012:0005:2
                                                                    02204500 T  0012:0006:0
                                                                    02204750 T  0012:0006:0
                                                                    02205000 T  0012:0009:2
                                                                    02205250 T  0012:0009:3
                                                                    02205500 T  0012:0010:0
                                                                    02208500 T  0012:0011:2
                                                                    02208750 T  0012:0011:3
                                                                    02209000 T  0012:0011:3
                                                                    02209250 T  0012:0012:0
                                                                    02209500 T  0012:0013:2
                                                                    02209750 T  0012:0013:2
                                                                    02210000 T  0012:0015:2
                                                                    02210250 T  0012:0015:2
                                                                    02210500 T  0012:0015:2
                                                                    02210750 T  0012:0015:2
                                                                    02211500 T  0012:0020:0
                                                                    02211750 T  0012:0023:2
                                                                    02211800 T  0012:0023:2
                                                                    02212000 T  0012:0023:3
                                                                    02212250 T  0012:0025:3
                                                                    02212500 T  0012:0028:1
                                                                    02212750 T  0012:0029:3
                                                                    02213000 T  0012:0031:2
                                                                    02213250 T  0012:0032:0
                                                                    02213500 T  0012:0032:1
                                                                    02213750 T  0012:0033:2
                                                                    02214000 T  0012:0037:2
                                                                    02214100 T  0012:0040:1
                                                                    02214200 T  0012:0041:3
                                                                    02214250 T  0012:0061:2
                                                                    02214260 C  0012:0063:2
                                                                    02214500 T  0012:0067:2
                                                                    02214750 T  0012:0068:0
                                                                    02215000 T  0012:0068:1
                                                                    02215250 T  0012:0068:1
                                                                    02215500 T  0012:0069:2
                                                                    02215750 T  0012:0073:2
                                                                    02216000 T  0012:0075:3
                                                                    02216250 T  0012:0076:0
                                                                    02216500 T  0012:0081:2
                                                                    02216750 T  0012:0083:2
                                                                    02217000 T  0012:0083:3
                                                                    02217250 T  0012:0084:0
                                                                    02217500 T  0012:0086:0
                                                                    02217750 T  0012:0088:1
                                                                    02218000 T  0012:0090:1
```

```
    TURNONSTOPLIGHT("%",CLCR-8);                                                    02218250 T  0012:0092:0
%                                                                                   02218400 T  0012:0093:3
    GO COMPAR;                                                                      02218500 T  0012:0093:3
COMMENT  THIS RELEASES THE PREVIOUS CARD FROM CARD READER AND                       02218750 T  0012:0097:2
         SETS UP CLCR;                                                              02219000 T  0012:0097:2
TAPELAST:                                                                           02219250 T  0012:0097:2
    READTAPE;                                                                       02219500 T  0012:0097:2
COMMENT  THIS RELEASES THE PREVIOUS CARD FROM TAPE AND SETS UP TLCR;                02219750 T  0012:0097:3
COMPAR:                                                                             02224250 T  0012:0097:3
    SEQCOMPARE(TLCR,CLCR,FALSE);                                                    02224500 T  0012:0098:0
EXIT:                                                                               02225000 T  0012:0099:2
    NCR := FCR:= LCR - 9;                                                           02225250 T  0012:0100:0
COMMENT  SETS UP NCR AND FCR;                                                       02225500 T  0012:0101:3
    IF EXAMIN(FCR)≠"$" THEN % $-CARDS DON"T COUNT,                                  02225750 T  0012:0101:3
      IF COMPARE(MKABS(INFO[LASTSEQROW,LASTSEQUENCE]),LCR)=1 THEN                   02226000 T  0012:0103:3
        BEGIN                                                                       02226250 T  0012:0108:0
        FLAG(610); % SEQUENCE ERROR.                                                02226500 T  0012:0108:1
        SEQUENCEERROR(LIN);                                                         02226750 T  0012:0109:2
        END;                                                                        02227000 T  0012:0110:0
    IF LASTUSED=3 THEN                                                              02228050 T  0012:0110:0
        BEGIN                                                                       02228075 T  0012:0111:2
        IF VOIDTAPE THEN GO USETHESWITCH;                                           02228100 T  0012:0111:3
        IF VOIDTCR≠0 THEN                                                           02228125 T  0012:0113:2
          IF COMPARE(LCR,VOIDTCR)=0 THEN GO USETHESWITCH;                           02228150 T  0012:0113:3
        END;                                                                        02228175 T  0012:0116:1
    IF EXAMIN(FCR)="$" THEN                                                         02228250 T  0012:0116:1
        BEGIN                                                                       02228500 T  0012:0118:1
        IF LISTPTOG OR PRINTDOLLARTOG THEN PRINTCARD;                               02228750 T  0012:0119:2
        NCR:=NCR+32768; DOLLARCARD;                                                 02229000 T  0012:0136:1
COMMENT  DONT FORGET THAT NCR IS NOT WORD MODE, BUT CHAR, MODE POINTER;             02229250 T  0012:0138:0
        GO USETHESWITCH;                                                            02229500 T  0012:0138:0
        END;                                                                        02229750 T  0012:0140:0
    IF EXAMIN(FCR)=" " THEN                                                         02230000 T  0012:0140:0
      IF DOLLAR2TOG:=EXAMIN(FCR+32768)="$" THEN                                     02230100 T  0012:0141:3
        BEGIN                                                                       02230250 T  0012:0145:2
        OUTPUTSOURCE;                                                               02230500 T  0012:0145:3
        NCR:=NCR+65536; % SCAN PAST " $" (CHARACTER MODE),                          02230750 T  0012:0146:0
        DOLLARCARD;                                                                 02231000 T  0012:0147:2
        END;                                                                        02231250 T  0012:0147:3
    IF VOIDING THEN GO USETHESWITCH;                                                02231500 T  0012:0147:3
    IF VOIDCR≠0 THEN                                                                02231750 T  0012:0149:2
        IF COMPARE(LCR,VOIDCR)>0 THEN VOIDCR:=VOIDPLACE:=0                          02232000 T  0012:0150:0
          ELSE GO USETHESWITCH;                                                     02232250 T  0012:0153:2
    IF VOIDTAPE THEN GO TESTVOID;                                                   02232500 T  0012:0154:0
    IF VOIDTCR≠0 THEN                                                               02233000 T  0012:0155:3
        IF COMPARE(LCR,VOIDTCR)>0 THEN VOIDTCR:=VOIDTPLACE:=0 ELSE                  02233500 T  0012:0156:1
TESTVOID:  IF LASTUSED=3 THEN GO USETHESWITCH;                                      02234000 T  0012:0160:1
    CARDCOUNT:=CARDCOUNT+1;                                                         02234500 T  0012:0165:2
    IF DOLLAR2TOG THEN GO USETHESWITCH;                                             02234600 T  0012:0166:1
    PUTSEQNO(INFO[LASTSEQROW,LASTSEQUENCE],LCR);                                    02234750 T  0012:0167:3
        CARDNUMBER:=IF SEQTOG THEN TOTALNO+ADDVALUE ELSE          %108-             02234800 C  0012:0169:3
                    CONV(INFO[LASTSEQROW,LASTSEQUENCE-1],5,8);    %108-             02234900 C  0012:0172:0
    OUTPUTSOURCE;                                                                   02235000 T  0012:0175:3
    IF OMITTING THEN GO USETHESWITCH;                                               02235250 T  0012:0176:0
%                                                                                   02235500 T  0012:0177:3
    TURNONSTOPLIGHT("%",LCR);                                                       02235750 T  0012:0177:3
XIT:                                                                                02237750 T  0012:0178:1
```

```
    END READACARD;

REAL PROCEDURE CONVERT;
          BEGIN   REAL  T;  INTEGER  N;

               TLO←0;   THI←
                    T← CONV(ACCUM[1],TCOUNT,N←(COUNT-TCOUNT)MOD 8);
                    FOR N← TCOUNT+N STEP 8 UNTIL COUNT= 1 DO
               IF DPTOG THEN
               BEGIN
               DOUBLE(THI,TLO,100000000,0,0,×,CONV(ACCUM[1],N,8),0,+,←,
                              THI,TLO);
               T←THI;
               END  ELSE
                    T←  T×100000000+ CONV(ACCUM[1],N,8);
                    CONVERT←T;
               END;

REAL STREAM PROCEDURE FETCH(F); VALUE F;
    BEGIN  SI:=F; SI:=SI-8; DI:=LOC FETCH; DS:=WDS  END FETCH;
PROCEDURE DUMPINFO;
    BEGIN
    ARRAY A[0:14]; INTEGER JEDEN,DWA;

    STREAM PROCEDURE OCTALWORDS(S,D,N); VALUE N;
         BEGIN
         SI:=S; DI:=D;
         N(2(8(DS:=3 RESET; 3(IF SB THEN DS:=1 SET ELSE
              DS:=1 RESET; SKIP 1 SR)); DS:=1 LIT " ");DS:=2 LIT" ");
         END OF OCTALWORDS;
    STREAM PROCEDURE ALPHAWORDS(S,D,N); VALUE N;
         BEGIN
         SI:=S; DI:=D;
         N(2(4(DS:=1 LIT" "; DS:=1 CHR); DS:=1 LIT" "); DS:=2 LIT" ");
         END OF ALPHAWORDS;
    IF NOHEADING THEN DATIME;WRITE(LINE[DBL],<//"ELBAT">);

    FOR JEDEN:=0 STEP 6 UNTIL 71 DO
         BEGIN
         BLANKET(14,A); OCTALWORDS(ELBAT[JEDEN],A,6);
         WRITE(LINE[DBL],15,A[*]);
         END;
    BLANKET(14,A); OCTALWORDS(ELBAT[72],A,4);
    WRITE(LINE[DBL],15,A[*]);
    FOR JEDEN:=0 STEP 1 UNTIL NEXTINFO DIV 256 DO
         BEGIN
         WRITE(LINE[DBL],<//"INFO[",I2,",*]">,JEDEN);

         FOR DWA:=0 STEP 6 UNTIL 251 DO
              BEGIN
              BLANKET(14,A); ALPHAWORDS(INFO[JEDEN,DWA],A,6);
              WRITE(LINE,15,A[*]);
              BLANKET(14,A); OCTALWORDS(INFO[JEDEN,DWA],A,6);
              WRITE(LINE[DBL],15,A[*]);
              END;
         BLANKET(14,A); ALPHAWORDS(INFO[JEDEN,252],A,4);
         WRITE(LINE,15,A[*]);
```

```
                02238000 T  0012:0179:2
          12 IS  183 LONG, NEXT SEG    3
                02248000 T  0003:0239:2
                02249000 T  0003:0239:2
         START OF SEGMENT **********   14
                02250000 T  0014:0000:0
                02251000 T  0014:0000:1
                02252000 T  0014:0005:2
                02253000 T  0014:0010:0
                02254000 T  0014:0010:0
                02255000 T  0014:0010:1
                02256000 T  0014:0015:2
                02257000 T  0014:0016:0
                02258000 T  0014:0016:1
                02259000 T  0014:0016:1
                02260000 T  0014:0024:0
                02261000 T  0014:0024:1
          14 IS   28 LONG, NEXT SEG    3
                02262000 T  0003:0239:2
                02263000 T  0003:0239:2
                02264000 T  0003:0242:0
                02264050 T  0003:0242:0
                02264100 T  0003:0242:0
         START OF SEGMENT **********   15
                02264400 T  0015:0001:3
                02264450 T  0015:0001:3
                02264500 T  0015:0003:2
                02264550 T  0015:0003:3
                02264600 T  0015:0006:1
                02264650 T  0015:0009:2
                02264700 T  0015:0009:2
                02264750 T  0015:0009:2
                02264800 T  0015:0010:0
                02264850 T  0015:0010:1
                02264900 T  0015:0014:1
                02264950 T  0015:0014:1
          16 IS    6 LONG, NEXT SEG   15
                02265000 T  0015:0019:2
                02265050 T  0015:0020:0
                02265100 T  0015:0020:0
                02265150 T  0015:0023:2
                02265200 T  0015:0027:2
                02265250 T  0015:0029:3
                02265300 T  0015:0032:1
                02265350 T  0015:0036:1
                02265400 T  0015:0041:2
                02265450 T  0015:0041:2
          17 IS    8 LONG, NEXT SEG   15
                02265500 T  0015:0048:0
                02265550 T  0015:0049:2
                02265600 T  0015:0049:2
                02265650 T  0015:0052:1
                02265700 T  0015:0057:2
                02265750 T  0015:0060:1
                02265800 T  0015:0065:2
                02265850 T  0015:0067:2
                02265900 T  0015:0071:2
```

```
            BLANKET(14,A); OCTALWORDS(INFO[JEDEN,252],A,4);
            WRITE(LINE[DBL],15,A[*]);
            END;
        END OF DUMPINFO;

DEFINE SKAN = BEGIN
                COUNT:=RESULT:=ACCUM[1]:=0;
                SCANNER;
                Q:=ACCUM[1];
                END #;
COMMENT   DOLLARCARD HANDLES THE COMPILER CONTROL CARDS.
          ALL COMPILER- AND USER-DEFINED OPTIONS ARE KEPT
          IN THE ARRAY "OPTIONS".
          EACH OPTION HAS A TWO-WORD ENTRY:

            WORD            CONTAINS
            ----            --------
              1             ENTRY FROM ACCUM[1]: 00XZZZZ, WHERE
                            X IS THE SIZE OF THE ID AND
                            ZZZZZ IS THE FIRST FIVE CHARS OF THE ID.
              2             PUSH-DOWN, 47-BIT STACK CONTAINING THE
                            HISTORY OF THE SETTINGS OF THIS OPTION.

          IN "FINDOPTION", ALL COMPILER-DEFINED OPTIONS ARE USUALLY
          LOCATED BASED UPON A UNIQUE NUMBER ASSIGNED TO EACH.
          FOR ALL USER-DEFINED OPTIONS, A SEQUENTIAL TABLE SEARCH IS
          INITIATED USING "USEROPINX" AS THE INITIAL INDEX INTO THE
          "OPTIONS" ARRAY. IF THE NUMBER OF COMPILER-DEFINED OPTIONS
          IS CHANGED, THEN "USEROPINX" MUST BE ACCORDINGLY CHANGED.
          THE NUMBER OF USER DEFINED OPTIONS ALLOWED CAN BE
          CHANGED BY CHANGING THE DEFINE "OPARSIZE".
          THE VARIABLE "OPTIONWORD" CONTAINS THE CURRENT TRUE OR FALSE
          SETTING OF ALL OF THE COMPILER-DEFINED OPTIONS, ONE BIT PER
          OPTION.
 ;
BOOLEAN PROCEDURE FINDOPTION(BIT); VALUE BIT; INTEGER BIT;
    BEGIN
    LABEL FOUND;

    REAL ID;
    OPINX:=2*BIT-4;
    WHILE ID:=OPTIONS[OPINX:=OPINX+2]#0 DO
        IF Q=ID THEN GO FOUND;
    OPTIONS[OPINX]:=Q; % NEW USER-DEFINED OPTION.
FOUND:                                                          %103-
    IF OPINX +1>OPARSIZE THEN FLAG(602) ELSE % TOO MANY USER OPTIONS
    FINDOPTION:=BOOLEAN(OPTIONS[OPINX+1]);
    END FINDOPTION;

PROCEDURE DOLLARCARD;
    BEGIN
    STREAM PROCEDURE RESTORESEQNUM(LCR,INFO);  VALUE LCR;

        BEGIN
        DI:=LCR; SI:=INFO; DS:=WDS;
        END;
    PROCEDURE SWITCHIT(XBIT); VALUE XBIT; INTEGER XBIT;
```

```
                02265950 T  0015:0075:2
                02266000 T  0015:0079:2
                02266050 T  0015:0083:2
                02266100 T  0015:0083:3
          15 IS   88 LONG, NEXT SEG    3
                02277000 T  0003:0242:0
                02278000 T  0003:0242:0
                02279000 T  0003:0242:0
                02280000 T  0003:0242:0
                02281000 T  0003:0242:0
                02282000 T  0003:0242:0
                02283000 T  0003:0242:0
                02284000 T  0003:0242:0
                02285000 T  0003:0242:0
                02286000 T  0003:0242:0
                02287000 T  0003:0242:0
                02288000 T  0003:0242:0
                02289000 T  0003:0242:0
                02290000 T  0003:0242:0
                02291000 T  0003:0242:0
                02292000 T  0003:0242:0
                02293000 T  0003:0242:0
                02294000 T  0003:0242:0
                02295000 T  0003:0242:0
                02296000 T  0003:0242:0
                02297000 T  0003:0242:0
                02298000 T  0003:0242:0
                02299000 T  0003:0242:0
                02300000 T  0003:0242:0
                02301000 T  0003:0242:0
                02302000 T  0003:0242:0
                02303000 T  0003:0242:0
                02304000 T  0003:0242:0
                02305000 T  0003:0242:0
                02306000 T  0003:0242:0
                02307000 T  0003:0242:0
                02308000 T  0003:0242:0
                02309000 T  0003:0242:0
          START OF SEGMENT **********   18
                02310000 T  0018:0000:0
                02311000 T  0018:0000:0
                02312000 T  0018:0001:3
                02313000 T  0018:0005:2
                02314000 T  0018:0006:1
                02315000 P  0018:0008:0
                02316000 P  0018:0008:0
                02317000 T  0018:0010:1
                02318000 T  0018:0012:1
          18 IS   15 LONG, NEXT SEG    3
                02319000 T  0003:0242:0
                02320000 T  0003:0242:0
                02320200 T  0003:0242:0
          START OF SEGMENT **********   19
                02320400 T  0019:0000:0
                02320600 T  0019:0000:0
                02320800 T  0019:0000:1
                02321000 T  0019:0001:2
```

```
                BEGIN
                BOOLEAN B,T;

                INTEGER SAVEINX;
                LABEL XMODE0,XMODE1,XMODE2,XMODE3,XMODE4,ALONG;
                SWITCH SW:=XMODE0,XMODE1,XMODE2,XMODE3,XMODE4;
                SETTING:=FINDOPTION(XBIT); SKAN;
                GO SW[XMODE+1];
XMODE0:         % FIRST OPTION ON CARD, BUT NOT SET, RESET, OR POP.
                OPTIONWORD:=BOOLEAN(0);
                FOR SAVEINX:=1 STEP 2 UNTIL OPARSIZE DO OPTIONS[SAVEINX]:=0;
                XMODE:=LASTUSED:=1; % CARD INPUT ONLY.
XMODE1:         % NOT FIRST OPTION AND NOT BEING SET, RESET, OR POPPED.
                OPTIONS[OPINX+1]:=REAL(TRUE);
                IF XBIT<USEROPINX THEN OPTIONWORD:=OPTIONWORD & TRUE[XBIT:1];
                GO ALONG;
XMODE2:         % RESET.
                OPTIONS[OPINX+1]:=REAL(FALSE & SETTING[1:2:46]);
                IF XBIT<USEROPINX THEN OPTIONWORD:=OPTIONWORD & FALSE[XBIT:1];
                GO ALONG;
XMODE3:         % SET.
                SAVEINX:=OPINX; % REMEMBER OPTION WE ARE SETTING.
                B:=IF Q="1=0000" THEN BOOLEXP ELSE TRUE;
                OPTIONS[SAVEINX+1]:=REAL(B & SETTING[1:46]);
                IF XBIT<USEROPINX THEN OPTIONWORD:=OPTIONWORD & B [XBIT:1];
                GO ALONG;
XMODE4:         % POP.
                OPTIONS[OPINX+1]:=REAL(B:=SETTING.[1:46]);
                IF XBIT<USEROPINX THEN OPTIONWORD:=OPTIONWORD & B [XBIT:1];
ALONG:
                END SWITCHIT;

        LABEL EXIT,AGAIN,SKANAGAIN,LENGTH1,LENGTH2,LENGTH3,LENGTH4,
              LENGTH5,LENGTH6,LENGTH7,LENGTH8,LENGTH9,
              WHATISIT,
              CARDOPTION,MERGEOPTION;
        SWITCH OPTIONLENGTH:=LENGTH1,WHATISIT,LENGTH3,LENGTH4,LENGTH5,
                             LENGTH6,LENGTH7,WHATISIT,LENGTH9,WHATISIT;
        INTEGER SRESULT,SCOUNT;
        ALPHA VOIDRANGE;
        DOLLARTOG:=TRUE;
        MOVE(10,ACCUM[0],DEFINEARRAY[0]); % SAVE INFORMATION FOR
        SCOUNT:=COUNT; SRESULT:=RESULT;   % "TABLE" TO RESUME SCAN.
        XMODE:=0;
        PUTSEQNO(INFO[LASTSEQROW,LASTSEQUENCE],LCR);
        TURNONSTOPLIGHT("%",LCR);
SKANAGAIN:
        SKAN;
AGAIN:
        GO OPTIONLENGTH[MIN(COUNT,10)];
LENGTH1:
        IF Q = "1%0000" THEN GO EXIT;
        IF Q = "1$0000" THEN
            BEGIN SWITCHIT(PRINTDOLLARBIT); GO AGAIN END;
        IF Q = "1,0000" THEN GO SKANAGAIN;
        GO WHATISIT;
LENGTH2: % NO OPTIONS OF THIS LENGTH ARE CURRENTLY IMPLEMENTED.
```

```
02322000 T  0019:0001:2
02323000 T  0019:0001:2
START OF SEGMENT *********   20
02324000 T  0020:0000:0
02325000 T  0020:0000:0
02326000 T  0020:0000:0
02327000 T  0020:0005:3
02328000 T  0020:0011:2
02329000 T  0020:0013:3
02330000 T  0020:0014:0
02331000 T  0020:0014:1
02332000 T  0020:0019:3
02333000 T  0020:0020:1
02334000 T  0020:0021:2
02335000 T  0020:0022:1
02336000 T  0020:0026:1
02337000 T  0020:0027:2
02338000 T  0020:0028:0
02339000 T  0020:0030:1
02340000 T  0020:0034:1
02341000 T  0020:0035:2
02342000 T  0020:0036:0
02343000 T  0020:0036:1
02352000 T  0020:0039:3
02353000 T  0020:0042:1
02354000 T  0020:0046:1
02355000 T  0020:0048:0
02356000 T  0020:0048:0
02357000 T  0020:0051:2
02358000 T  0020:0055:2
02359000 T  0020:0056:0
20 IS   59 LONG, NEXT SEG   19
02360000 T  0019:0001:2
02361000 T  0019:0001:2
02362000 T  0019:0001:2
02363000 T  0019:0001:2
02364000 T  0019:0001:2
02365000 T  0019:0003:3
02365100 T  0019:0009:2
02365200 T  0019:0009:2
02366000 T  0019:0009:2
02366100 T  0019:0010:1
02366200 T  0019:0012:1
02367000 T  0019:0014:0
02368000 T  0019:0014:1
02369000 T  0019:0016:1
02370000 T  0019:0017:3
02371000 T  0019:0018:0
02372000 T  0019:0021:3
02373000 T  0019:0022:0
02374000 T  0019:0026:0
02375000 T  0019:0027:2
02376000 T  0019:0028:0
02377000 T  0019:0029:2
02378000 T  0019:0033:2
02379000 T  0019:0034:0
02380000 T  0019:0036:0
```

```
LENGTH3:
    IF Q = "3SETOO" THEN
        BEGIN XMODE:=3; GO SKANAGAIN END;
    IF Q = "3POPOO" THEN
        BEGIN XMODE:=4; GO SKANAGAIN END;
    IF Q = "3NEWOO" THEN
        BEGIN
        SWITCHIT(NEWBIT);
        IF Q = "4TAPEO" THEN GO SKANAGAIN;
        GO AGAIN;
        END;
    IF Q = "3SEQOO" THEN
        BEGIN SWITCHIT(SEQBIT); GO AGAIN END;
    IF Q = "3PRTOO" THEN
        BEGIN SWITCHIT(PRTBIT); GO AGAIN END;
    IF Q = "3MCPOO" THEN
        BEGIN SWITCHIT(MCPBIT); GO AGAIN END;
    GO WHATISIT;
LENGTH4:
    IF Q = "4LISTO" THEN
        BEGIN
        SWITCHIT(LISTBIT);
        GO AGAIN;
        END;
    IF Q = "4VOIDO" THEN
        BEGIN
        IF XMODE=0 THEN
            BEGIN
            GETVOID(VOIDRANGE,NCR,LCR,INFO[LASTSEQROW,LASTSEQUENCE]);
            IF VOIDCR=0 THEN VOIDCR:=MKABS(VOIDPLACE) ELSE
              IF COMPARE(MKABS(VOIDRANGE),VOIDCR)#1 THEN GO TO EXIT;
            MOVE(1,VOIDRANGE,VOIDPLACE);
            GO EXIT;
            END;
        SWITCHIT(VOIDBIT);
        GO AGAIN;
        END;
    IF Q = "4XREFO" THEN BEGIN SWITCHIT(XREFBIT); IF BOOLEAN(XMODE) THEN
        DEFINING := BOOLEAN(REAL(DEFINING)&1[1:47:1]);              %108-
                    GO AGAIN  END;                                  %108-
    IF Q = "4BENDO" THEN BEGIN SWITCHIT(BENDBIT); GO AGAIN END;     %108-
    IF Q = "4OMITO" THEN
        BEGIN IF NOT DOLLAR2TOG THEN BEGIN PRINTCARD; FLAG(605); END;
            SWITCHIT(OMITBIT); GO AGAIN END;
    IF Q = "4CARDO" THEN
        BEGIN
        Q:="4TAPEO"; % FAKE OUT SWITCHIT.
        SWITCHIT(MERGEBIT);
        IF XMODE#2 THEN MERGETOG:=NOT MERGETOG;
        OPTIONS[2xMERGEBIT-1]:=                 % CARD IS
          REAL(SETTING & (MERGETOG)[47:1]);     % INVERSE OF MERGE.
        IF MERGETOG THEN GO MERGEOPTION;
CARDOPTION:
        LASTUSED:=1;
        GO AGAIN;
        END;
    IF Q = "4TAPEO" THEN
```

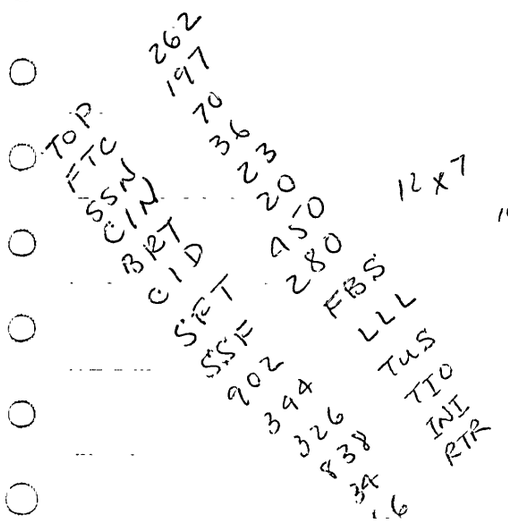
```
02381000 T  0019:0036:0
02382000 T  0019:0036:0
02383000 T  0019:0036:1
02384000 T  0019:0040:0
02385000 T  0019:0040:1
02386000 T  0019:0044:0
02387000 T  0019:0044:1
02388000 T  0019:0045:2
02389000 T  0019:0046:0
02390000 T  0019:0047:2
02391000 T  0019:0050:0
02392000 T  0019:0050:0
02393000 T  0019:0050:1
02394000 T  0019:0054:0
02395000 T  0019:0054:1
02396000 T  0019:0058:0
02397000 T  0019:0058:1
02398000 T  0019:0062:0
02399000 T  0019:0062:1
02400000 T  0019:0063:2
02401000 T  0019:0063:3
02402000 T  0019:0064:0
02404000 T  0019:0065:2
02405000 T  0019:0067:2
02406000 T  0019:0067:2
02407000 T  0019:0067:3
02408000 T  0019:0068:0
02409000 T  0019:0069:2
02410000 T  0019:0069:3
02410500 T  0019:0072:0
02411000 T  0019:0075:2
02412000 T  0019:0080:1
02413000 T  0019:0081:3
02414000 T  0019:0082:0
02415000 T  0019:0082:0
02418000 T  0019:0083:2
02419000 T  0019:0083:3
02419100 C  0019:0083:3
02419110 C  0019:0085:3
02419120 C  0019:0088:0
02419200 C  0019:0090:0
02420000 T  0019:0094:0
02421000 T  0019:0094:1
02421100 T  0019:0114:0
02422000 T  0019:0115:2
02423000 T  0019:0116:0
02424000 T  0019:0116:1
02425000 T  0019:0117:2
02425500 T  0019:0118:0
02426000 T  0019:0121:3
02427000 T  0019:0123:2
02428000 T  0019:0125:3
02429000 T  0019:0127:2
02430000 T  0019:0127:2
02431000 T  0019:0127:3
02432000 T  0019:0131:2
02433000 T  0019:0131:2
```

```
                BEGIN
                SWITCHIT(MERGEBIT);
                IF NOT MERGETOG THEN GO CARDOPTION;
MERGEOPTION:
                LASTUSED:=2; % NEXT CARD IS READ FROM READER,
                IF MAXTLCR=0 THEN
                     BEGIN
                     INTEGER STREAM PROCEDURE FEJ(F,T); VALUE T;

                          BEGIN
                          SI:=F; DI:=LOC T; DS:=WDS;
                          SI:=T; SI:=SI-16; DI:=LOC FEJ; DS:=WDS;
                          END FEJ;
                     STREAM PROCEDURE FIX(F,T); VALUE T;
                          BEGIN
                          SI:=F; SI:=SI-24; DI:=LOC T; DS:=WDS;
                          DI:=T; DI:=DI+47; SKIP 4 DB; DS:=2 RESET;
                          2(DI:=DI+48); DS:=8 LIT"00#01+0#";
                          END FIX;
                     IF GT1:=FEJ(TAPE,0)=10 THEN
                          BEGIN
                          REWIND(TAPE); FIX(TAPE,0);
                          END;
                     MAXTLCR:=GT1+TLCR:=9+MKABS(TBUFF[0]);
                     READ(TAPE,10,TBUFF[*]); % INITIALIZE TAPE INPUT,
                     LASTUSED:=2;
                     END;
                          GO AGAIN;
                          END;
                     IF Q = "4PAGE0" THEN
                          BEGIN
                          IF LISTER THEN WRITE(LINE[PAGE]);
                          GO SKANAGAIN;
                          END;
                     IF Q = "4INFO0" THEN
                          BEGIN DUMPINFO; GO SKANAGAIN END;
                     IF Q = "4SEGS0" THEN
                          BEGIN SWITCHIT(SEGSBIT); GO AGAIN END;
                     IF Q = "4NEST0" THEN
                          BEGIN SWITCHIT(NESTBIT); GO AGAIN END;
                     IF Q = "4DECK0" THEN
                          BEGIN SWITCHIT(DECKBIT); GO AGAIN END;
                     GO WHATISIT;
LENGTH5:
                     IF Q = "5RESET" THEN
                          BEGIN XMODE:=2; GO SKANAGAIN END;
                     IF Q = "5LISTP" THEN
                          BEGIN SWITCHIT(LISTPBIT); GO AGAIN; END;
                     IF Q = "5VOIDT" THEN
                          BEGIN
                          IF XMODE=0 THEN
                               BEGIN
                               GETVOID(VOIDRANGE,NCR,LCR,INFO[LASTSEQROW,LASTSEQUENCE]);
                               IF VOIDTCR=0 THEN VOIDTCR:=MKABS(VOIDTPLACE) ELSE
                                 IF COMPARE(MKABS(VOIDRANGE),VOIDTCR)#1 THEN GO TO EXIT;
                               MOVE(1,VOIDRANGE,VOIDTPLACE);
```

```
                02434000 T  0019:0131:3
                02435000 T  0019:0132:0
                02436000 T  0019:0133:2
                02437000 T  0019:0134:0
                02438000 T  0019:0135:2
                02439000 T  0019:0135:3
                02440000 T  0019:0136:1
                02441000 T  0019:0137:2
          START OF SEGMENT **********   21
                02442000 T  0021:0000:0
                02443000 T  0021:0000:0
                02444000 T  0021:0000:1
                02445000 T  0021:0001:3
                02446000 T  0021:0002:1
                02447000 T  0021:0002:1
                02448000 T  0021:0003:2
                02449000 T  0021:0004:0
                02450000 T  0021:0005:2
                02451000 T  0021:0007:2
                02452000 T  0021:0007:2
                02453000 T  0021:0010:1
                02454000 T  0021:0011:2
                02455000 T  0021:0014:0
                02456000 T  0021:0014:0
                02457000 T  0021:0017:3
                02458000 T  0021:0022:0
                02459000 T  0021:0022:1
           21 IS   24 LONG, NEXT SEG   19
                02460000 T  0019:0139:2
                02461000 T  0019:0139:3
                02462000 T  0019:0139:3
                02463000 T  0019:0140:0
                02464000 T  0019:0140:1
                02465000 T  0019:0145:3
                02466000 T  0019:0147:2
                02467000 T  0019:0147:2
                02468000 T  0019:0147:3
                02469000 T  0019:0151:2
                02470000 T  0019:0151:3
                02471000 T  0019:0155:2
                02472000 T  0019:0155:3
                02473000 T  0019:0159:2
                02474000 T  0019:0159:3
                02475000 T  0019:0163:2
                02476000 T  0019:0163:3
                02477000 T  0019:0164:0
                02478000 T  0019:0164:1
                02479000 T  0019:0168:0
                02480000 T  0019:0168:1
                02481000 T  0019:0172:0
                02482000 T  0019:0172:1
                02483000 T  0019:0173:2
                02484000 T  0019:0174:0
                02485000 T  0019:0174:1
                02485500 T  0019:0177:2
                02486000 T  0019:0180:0
                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  0019:0209:3
        IF Q = "5LISTA" THEN                                                            02509000 T  0019:0213:2
            BEGIN                                                                       02510000 T  0019:0213:3
            SWITCHIT(LISTABIT);                                                         02511000 T  0019:0214:0
            GO AGAIN;                                                                   02513000 T  0019:0215:2
            END;                                                                        02514000 T  0019:0217:2
        IF Q = "5STUFF" THEN                                                            02515000 T  0019:0217:2
            BEGIN SWITCHIT(STUFFBIT); GO AGAIN; END;                                    02516000 T  0019:0217:3
    GO WHATISIT;                                                                        02517000 T  0019:0221:2
LENGTH6:                                                                                02518000 T  0019:0221:3
    IF Q = "6SEQER" THEN                                                                02519000 T  0019:0222:0
        BEGIN SWITCHIT(SEQERRBIT); GO AGAIN END;                                        02520000 T  0019:0222:1
    IF Q = "6SINGL" THEN                                                                02521000 T  0019:0226:0
        BEGIN SWITCHIT(SINGLBIT); GO AGAIN  END;                                        02522000 T  0019:0226:1
    IF Q = "6SEQXE" THEN                                                                02523000 T  0019:0230:0
        BEGIN                                                                           02524000 T  0019:0230:1
        SEQXEQTOG:=XMODE≠2 AND XMODE≠ 4 OR SEQXEQTOG;%NEVER RESET.                      02525000 T  0019:0231:2
        IF BUILDLINE,[45:1] THEN BUILDLINE:=TRUE;                                       02526000 T  0019:0234:0
        GO SKANAGAIN;                                                                   02527000 T  0019:0236:0
        END;                                                                            02528000 T  0019:0238:0
    IF Q = "6DEBUG" THEN                                                                02529000 T  0019:0238:0
        BEGIN                                                                           02530000 T  0019:0238:1
        SWITCHIT(DEBUGBIT);                                                             02531000 T  0019:0239:2
        IF DEBUGTOG THEN                                                                02533000 T  0019:0240:0
            IF WOP[0]=0 THEN                                                            02534000 T  0019:0240:1
                BEGIN                                                                   02535000 T  0019:0242:0
                FILL WOP[*] WITH                                                        02536000 T  0019:0242:1
        "LITC","    ",                                                                  02537000 T  0019:0243:2
                                                                        START OF SEGMENT **********   22
        "OPDC","DFSC",                                                                  02538000 T  0019:0244:1
 11, "NOP ", 12, "PRT ", 13, "DFL ", 16, "ADD ", 18, "PRL ", 19, "LNG ",               02539000 T  0019:0244:1
 21, "GFQ ", 22, "BBC ", 24, "INX ", 35, "LOR ", 37, "GTR ", 38, "BFC ",               02540000 T  0019:0244:1
 39, "RTN ", 40, "COC ", 48, "SUB ", 64, "MUL ", 67, "LND ", 68, "STD ",               02541000 T  0019:0244:1
 69, "NFQ ", 71, "XIT ", 72, "MKS ",128, "DIV ",130, "COM ",131, "LQV ",               02542000 T  0019:0244:1
132, "SND ",133, "XCH ",134, "CHS ",167, "RTS ",168, "CDC ",260, "LOD ",               02543000 T  0019:0244:1
261, "DUP ",278, "LBC ",294,"LFC ",322,"ZP1 ",384, "IDV ",532, "ISD ",                 02544000 T  0019:0244:1
533, "LFQ ",534, "BBW ",548, "ISN ",549, "LSS ",550, "BFW ",581, "EQL ",               02545000 T  0019:0244:1
582, "SSP ",790, "LBU ",806, "LFU ",896, "RDV ",                                       02546000 T  0019:0244:1
1003,"SCI ",1004,"SAN ",1019,"SCS ",                                                   02547000 T  0019:0244:1
  76 DIA      76 DIB     77 DIA
```



26 unused slots

```
 1023,1023,1023,1023,1023,1023,1023,1023,1023,1023,1023,1023,1023,1023,
        1023,1023,1023,1023,1023,1023,1023,1023,1023,1023,1023,  1023;
                                                          126      127
                    FILL COP[*] WITH  % CHARACTER MODE MNEMONICS
  0,"   ",0,0,

  0,"FXC ", 2,"BSD ", 3,"BSS ", 4,"RDA ", 5,"TRW ",6,"SED ", 7,"TDA ",
 12,"SDA ",13,"SSA ",14,"SFD ",15,"SRD ",18,"SFS ",20,"TEQ ",21, "TNE ",
 22,"TEG ",23,"TGR ",24,"SRS ",25,"SFS ",28,"TEL ",29,"TLS ",30,"TAN ",
 31,"BIT ",32,"INC ",33,"STC ",34,"SEC ",35,"CRF ",36,"JNC ",37,"JFC ",
 38,"JNS ",39,"JFW ",40,"RCA ",41,"FNS ",42,"BNS ",43,"RSA ",44,"SCA ",
 45,"JRC ",46,"TSA ",47,"JRV ",48,"CEQ ",49,"CNE ",50,"CEG ",51,"CGR ",
52,"BIT ",53,"BIR ",54,"OCV ",55,"ICV ",56,"CEL ",57,"CLS ",58,"FSU ",
 59,"FAD ",60,"TRP ",61,"TRN ",62,"TRZ ",63,"TRS ",64,0,64,0,64,0,64,0,
 64,0,64,0,64,0,64,0;

                    END;                                          200
        GO AGAIN;
        END;
    IF Q = "6FORMA" THEN
        BEGIN SWITCHIT(FORMATBIT); GO AGAIN; END;
    GO WHATISIT;
LENGTH7:
%   IF Q = "7INCLU" THEN
%       BEGIN DOLLARCARD:=STARTINCLUDING;  GO EXIT;  END;
%   IF Q = "7INCLN" THEN
%       BEGIN SWITCHIT(NEWINCLBIT);  GO AGAIN; END;
LENGTH8; % NO OPTIONS OF THIS LENGTH ARE CURRENTLY IMPLEMENTED.
LENGTH9:
    IF Q = "9INTRI" THEN
        BEGIN
        INTOG:=XMODE≠2 AND XMODE≠ 4 OR INTOG; % NEVER RESET.
        GO SKANAGAIN;
        END;
WHATISIT:
    IF RESULT=3 THEN
        BEGIN
        BASENUM:=CONV(ACCUM[1],0,ACCUM[1].[12:6]);
        TOTALNO:=-10;
        NEWBASE:=TRUE;
        GO SKANAGAIN;
        END;
    IF RESULT=2 THEN
        BEGIN
        IF Q = "1+0000" THEN
            BEGIN
            SKAN;
            IF RESULT=3 THEN
             ADDVALUE:=CONV(ACCUM[1],0,ACCUM[1].[12:6])
         ELSE FLAG(600); % NUMBER EXPECTED.
            END;
        GO SKANAGAIN;
        END;
COMMENT  DID NOT RECOGNIZE OPTION;
    IF RESULT≠1 THEN % NOT AN IDENTIFIER.
        BEGIN FLAG(601); GO SKANAGAIN END;
    SWITCHIT(USEROPINX); % USEROPINX MEANS A USER-DEFINED OPTION.
```

```
02548000 T  0019:0244:1
02549000 T  0019:0244:1
      22 IS  128 LONG, NEXT SEG   19
02550000 T  0019:0244:1
02551000 T  0019:0245:2
    START OF SEGMENT *********   23
02552000 T  0019:0246:0
02553000 T  0019:0246:0
02554000 T  0019:0246:0
02555000 T  0019:0246:0
02556000 T  0019:0246:0
02557000 T  0019:0246:0
02558000 T  0019:0246:0
02559000 T  0019:0246:0
02560000 T  0019:0246:0
      23 IS  128 LONG, NEXT SEG   19
02563000 T  0019:0246:0
02564000 T  0019:0246:0
02565000 T  0019:0248:0
02566000 T  0019:0248:0
02567000 T  0019:0248:1
02568000 T  0019:0252:0
02569000 T  0019:0252:1
02570000 T  0019:0253:2
02571000 T  0019:0253:2
02572000 T  0019:0253:2
02573000 T  0019:0253:2
02574000 T  0019:0253:2
02575000 T  0019:0253:2
02576000 T  0019:0253:2
02576500 T  0019:0253:3
02577000 T  0019:0254:0
02577250 T  0019:0258:1
02577500 T  0019:0260:0
02578000 T  0019:0260:0
02579000 T  0019:0260:0
02580000 T  0019:0260:1
02581000 T  0019:0261:2
02582000 T  0019:0264:1
02583000 T  0019:0265:3
02584000 T  0019:0266:0
02585000 T  0019:0266:1
02586000 T  0019:0266:1
02587000 T  0019:0267:3
02588000 T  0019:0268:0
02589000 T  0019:0268:1
02590000 T  0019:0269:2
02591000 T  0019:0273:2
02592000 T  0019:0273:3
02593000 T  0019:0276:0
02594000 T  0019:0279:3
02595000 T  0019:0279:3
02596000 T  0019:0280:0
02597000 T  0019:0280:0
02598000 T  0019:0280:0
02599000 T  0019:0281:2
02600000 T  0019:0282:1
```

```
      GO AGAIN;
   EXIT:
      LISTER:=DEBUGTOG OR LISTOG OR LISTATOG;
      MOVF(10,DEFINEARRAY[0],ACCUM[0]); % RESTORE INFORMATION FOR
      COUNT:=SCOUNT; RESULT:=SRESULT;   % "TABLE" TO RESUME SCAN.
      RESTORESEQNUM(LCR,INFO[LASTSEQROW,LASTSEQUENCE]); % FOR VOID TESTS
      DOLLARTOG:=FALSE;
      END DOLLARCARD;

COMMENT  TABLE IS THE ROUTINE THAT MOST CODE IN THE COMPILER
         USES WHEN IT IS DESIRED TO SCAN ANOTHER LOGICAL QUANTITY.
         THE RESULT RETURNED IS THE CLASS OF THE ITEM DESIRED.
         TABLE MAINTAINS THE VARIABLES I AND NXTELBT AND THE ARRAY
         ELBAT.  ELBAT AND I ARE PRINCIPAL VARIABLES USED FOR
         COMUNICATION BETWEEN TABLE AND THE OUTSIDE WORLD.  NXTELBT
         IS ALMOST EXCLUSIVELY USED BY TABLE, ALTHOUGH AN OCCASION-
         AL OTHER USE IS MADE IN ORDER TO FORGET THAT SOMETHING WAS
         SCANNED.  (SEE, FOR EXAMPLE, COMPOUNDTAIL). FOR FURTHER
         GENERAL DISCUSSION SEE THE DECLARATION OF THESE VARIABLES.
           THE PARAMETER P IS THE ACTUAL INDEX OF THE QUANTITY
         DESIRED (USUALLY I-1, I, OR I+1).
           THE GENERAL PLAN OF TABLE IS THIS:
             I) IF P < NXTELBAT GO ON TO III).
            II) PROCESS ONE QUANTITY.
              A) SCAN.
              B) TEST FOR IDENTIFIER, NUMBER, OR SPECIAL CHARACTER.
               1) IDENTIFIER - LOOKUP IN DIRECTORY AND PROCESS
                    IN SPECIAL MANNER IF COMMENT OR DEFINED ID.
               2) NUMBER - PROCESS INTEGER PART, FRACTIONAL PART,
                    AND EXPONENT PART.
               3) TEST IF SPECIAL CHARACTER REQUIRES SPECIAL
                    PROCESSING - OTHERWISE GET ELBAT WORD FROM
                    SPECIAL.
              C) LOAD ELBAT AND INCREMENT NXTELBT.
              D) IF ELBAT IS FULL ADJUST ELBAT, NXTELBT, I, AND P.
              E) GO BACK TO I).
           III) RETURN WITH CLASS OF ELBAT[P].
         FURTHER DETAILS ARE GIVEN IN BODY OF TABLE.
;
INTEGER PROCEDURE TABLE(P); VALUE P; INTEGER P;
    BEGIN
    LABEL PERCENT,SPECIALCHAR,COMPLETE,COLON,DOT,ATSIGN,QUOTE,
          STRNGXT,MOVEIT,ARGH,FINISHNUMBER,
          SCANAGAIN,FPART,EPART,IPART,IDENT,ROSE,COMPOST,DOLLAR,RTPAREN,
          CROSSHATCH,NUMBEREND;
    SWITCH SPECIALSWITCH:=PERCENT,DOLLAR,DOT,ATSIGN,COLON,QUOTE,
                          RTPAREN,CROSSHATCH;
    SWITCH RESULTSWITCH:=IDENT,SPECIALCHAR,IPART;
    WHILE P ≥ NXTELBT
    DO   BEGIN
SCANAGAIN:
         COUNT:=RESULT:=ACCUM[1]:=0; SCANNER;
         GO RESULTSWITCH[RESULT];
ARGH:
         Q:=ACCUM[1]; FLAG(141); GO SCANAGAIN;
SPECIALCHAR:
```

```
02601000 T  0019:0283:3
02602000 T  0019:0284:0
02602500 T  0019:0284:0
02602600 T  0019:0287:2
02602700 T  0019:0289:2
02602800 T  0019:0290:1
02603000 T  0019:0292:1
02604000 T  0019:0293:2
19 IS  296 LONG, NEXT SEG    3
02605000 T  0003:0242:0
02606000 T  0003:0242:0
02607000 T  0003:0242:0
02608000 T  0003:0242:0
02609000 T  0003:0242:0
02610000 T  0003:0242:0
02611000 T  0003:0242:0
02612000 T  0003:0242:0
02613000 T  0003:0242:0
02614000 T  0003:0242:0
02615000 T  0003:0242:0
02616000 T  0003:0242:0
02617000 T  0003:0242:0
02618000 T  0003:0242:0
02619000 T  0003:0242:0
02620000 T  0003:0242:0
02621000 T  0003:0242:0
02622000 T  0003:0242:0
02623000 T  0003:0242:0
02624000 T  0003:0242:0
02625000 T  0003:0242:0
02626000 T  0003:0242:0
02627000 T  0003:0242:0
02628000 T  0003:0242:0
02629000 T  0003:0242:0
02630000 T  0003:0242:0
02631000 T  0003:0242:0
02632000 T  0003:0242:0
02633000 T  0003:0242:0
02634000 T  0003:0242:0
02635000 T  0003:0242:0
02636000 T  0003:0242:0
02637000 T  0003:0242:0
START OF SEGMENT **********   24
02638000 T  0024:0000:0
02639000 T  0024:0000:0
02640000 T  0024:0000:0
02641000 T  0024:0000:0
02642000 T  0024:0002:1
02643000 T  0024:0007:2
02644000 T  0024:0012:1
02645000 T  0024:0013:2
02646000 T  0024:0014:0
02647000 T  0024:0015:2
02648000 T  0024:0017:3
02649000 T  0024:0019:3
02650000 T  0024:0020:0
02651000 T  0024:0022:0
```

```
                GT1:=ACCUM[1].[18:6] - 2;                                          02652000 T  0024:0023:2
                ENDTOG:=GT1 = 57 AND ENDTOG;                                       02653000 T  0024:0025:2
COMMENT         OBTAIN ACTUAL CHARACTER FROM ACCUM;                                02654000 T  0024:0026:1
                T:=SPECIAL[GT1&GT1[42:41:3]];                                      02655000 T  0024:0026:1
COMMENT         NOTICE COMPRESSION TECHNIQUE USED TO SHORTEN TABLE OF              02656000 T  0024:0028:1
                ELBAT WORDS FOR SPECIAL CHARACTERS;                                02657000 T  0024:0028:1
                IF GT1:=T.INCR = 0 THEN GO COMPLETE;                               02658000 T  0024:0028:1
                GO SPECIALSWITCH[GT1];                                             02659000 T  0024:0031:2
COMMENT         INCR FIELD OF SPECIAL CHARACTER IS NON-ZERO FOR SPECIAL             02660000 T  0024:0033:2
                CHARACTERS REQUIRING SPECIAL HANDLING.  INCR IS SWITCHED           02661000 T  0024:0033:2
                ON TO OBTAIN DISCRIMINATION;                                       02662000 T  0024:0033:2
COLON:          RESULT:=7; SCANNER; COMMENT ELIMINATE BLANKS - CHECKING             02663000 T  0024:0033:2
                                            FOR := IN PLACE OF ← ;                 02664000 T  0024:0034:0
                IF EXAMIN (NCR) = "=" THEN                                         02665000 T  0024:0034:0
                     BEGIN RESULT:=0; SCANNER; T:=SPECIAL[13]  END;                02666000 T  0024:0036:0
                RESULT:=2; GO COMPLETE;                                            02667000 T  0024:0038:1
DOT:                                                                               02668000 T  0024:0040:0
                IF EXAMIN(NCR)>9 OR ENDTOG THEN GO COMPLETE;                       02680000 T  0024:0040:0
                NHI:=NLO:=0;                                                       02681000 T  0024:0043:2
                C:=0; GO FPART;                                                    02682000 T  0024:0044:0
ATSIGN:                                                                            02683000 T  0024:0045:3
                RESULT:=0; SCANNER;          % SCAN PAST "@".                    02684000 T  0024:0046:0
                IF COUNT>17 THEN GO ARGH;   % 16 CHARS. + "@".                    02685000 T  0024:0047:2
                IF OCTIZE(ACCUM[1],C,17-COUNT,COUNT-1) THEN                        02686000 T  0024:0048:1
                     BEGIN  Q:=ACCUM[1]; FLAG(521); GO SCANAGAIN  END;             02686500 T  0024:0051:3
                GO NUMBEREND;                                                      02687000 T  0024:0054:1
COMMENT         DOT AND ATSIGN ENTER NUMBER CONVERSION AT CORRECT SPOT;            02689000 T  0024:0055:2
QUOTE:                                                                             02690000 T  0024:0055:2
                COUNT:=0;                                                          02691000 T  0024:0055:2
                T:=IF STREAMTOG THEN 63                                            02692000 T  0024:0055:3
            ELSE IF REAL(STREAMTOG)>1 THEN 8 ELSE 7;                               02692500 T  0024:0056:0
                DO   BEGIN                                                         02693000 T  0024:0060:0
                     RESULT:=5; SCANNER;                                           02694000 T  0024:0060:0
                     IF COUNT>T THEN                                               02695000 T  0024:0061:2
                          BEGIN  Q:=ACCUM[1]; FLAG(520); GO SCANAGAIN  END;        02696000 T  0024:0062:0
                     END UNTIL EXAMIN(NCR) = """;                                  02697000 T  0024:0064:1
                Q:=ACCUM[1]; RESULT:=5; SCANNER; COUNT:=COUNT-1;                   02698000 T  0024:0067:2
                IF COUNT<0 THEN COUNT:=COUNT+64;                                   02699000 T  0024:0070:1
                ACCUM[1]:=Q; RESULT:=4;                                            02700000 T  0024:0073:2
STRNGXT:        T:=C:=0;                                                           02701000 T  0024:0075:2
                IF COUNT < 8 THEN                                                  02703000 T  0024:0076:0
MOVEIT:                                                                            02704000 T  0024:0077:2
                MOVECHARACTERS(COUNT,ACCUM[1],3,C,8-COUNT);                        02705000 T  0024:0078:0
                T.CLASS:=STRNGCON;                                                 02705100 T  0024:0080:1
                GO COMPLETE;                                                       02705200 T  0024:0082:0
COMMENT         CROSSHATCH HANDLES TWO SITUATIONS:                                 02707000 T  0024:0082:1
                THE CROSSHATCH AT END OF DEFINE DECLARATIONS AND                   02708000 T  0024:0082:1
                THE CROSSHATCH AT END OF ALPHA REPRESENTING DEFINED IDS.           02709000 T  0024:0082:1
                THE TWO CASES ARE PROCESSED DIFFERENTLY.  THE FIRST CASE           02710000 T  0024:0082:1
                MERELY PLACES THE CROSSHATCH IN ELBAT.  THE SECOND CASE            02711000 T  0024:0082:1
                CAUSES AN EXIT FROM SCANNING THE ALPHA FOR THE DEFINED ID.         02712000 T  0024:0082:1
                FOR A FULL DISCUSSION SEE DEFINEGEN;                               02713000 T  0024:0082:1
CROSSHATCH:                                                                        02714000 T  0024:0082:1
                IF DEFINECTR≠0 THEN GO COMPLETE;                                   02715000 T  0024:0083:2
                PUTSEQNO(GT1,LCR);                                                 02716000 T  0024:0084:0
                TURNONSTOPLIGHT(0,LCR);                                            02717000 T  0024:0085:2
                IF DEFINEINDEX = 0 THEN GO ARGH;                                   02718000 T  0024:0086:0
```

```
                LCR:=(GT1:=DEFINEARRAY[DEFINEINDEX-1]) DIV 262144;                    02719000 T  0024:0087:3
                NCR:=GT1 MOD 262144;                                                  02720000 T  0024:0090:0
                GT2:=0&(T:=DEFINEARRAY[DEFINEINDEX:=DEFINEINDEX-3])[33:18:15];        02721000 T  0024:0091:2
                LASTUSED:=T.[33:15];                                                  02722000 T  0024:0094:0
                FOR GT1:=1 STEP 1 UNTIL GT2 DO                                        02723000 T  0024:0095:3
                     BEGIN                                                            02723500 T  0024:0098:0
                     STACKHEAD[(T:=TAKE(LASTINFO+1)).[12:36] MOD 125]:=               02724000 T  0024:0098:0
                               TAKE(LASTINFO).LINK;                                   02725000 T  0024:0101:2
                     LASTINFO:=(NEXTINFO:=LASTINFO)-T.PURPT;                          02726000 T  0024:0102:1
                     END;                                                             02727000 T  0024:0105:2
                GO SCANAGAIN;                                                         02728000 T  0024:0107:2
DOLLAR:         COMMENT THIS CODE HANDLES CONTROL CARDS;                              02729000 T  0024:0107:3
                DOLLARCARD;                                                           02730000 T  0024:0108:0
PERCENT:        IF NCR ≠ FCR THEN READACARD;                                          02731000 T  0024:0108:1
                GO SCANAGAIN;                                                         02737000 T  0024:0110:1
COMMENT         MOST PERCENT SIGNS ACTING AS END OF CARD SENTINELS GET TO              02738000 T  0024:0111:2
                PERCENT.  PERCENT READS THE NEXT CARD AND STARTS OVER.  A             02739000 T  0024:0111:2
                SIDE EFFECT IS THAT ALL CHARACTERS ON A CARD ARE IGNORED              02740000 T  0024:0111:2
                AFTER A FREE PERCENT SIGN (ONE NOT IMBEDDED IN A STRING OR            02741000 T  0024:0111:2
                COMMENT);                                                             02742000 T  0024:0111:2
COMMENT         MIGHT BE FUNNY COMMA - HANDLE HERE;                                   02743000 T  0024:0111:2
RTPAREN:        RESULT:=7; SCANNER;                                                   02744000 T  0024:0111:2
                IF EXAMIN(NCR) = """ THEN                                             02745000 T  0024:0113:2
                     BEGIN                                                            02746000 T  0024:0115:2
                     RESULT:=0; SCANNER;                                              02747000 T  0024:0115:3
                     DO   BEGIN                                                       02748000 T  0024:0116:1
                          RESULT:=5; SCANNER                                          02749000 T  0024:0117:2
                          END UNTIL EXAMIN(NCR) = """;                                02750000 T  0024:0117:3
                     RESULT:=0; SCANNER;                                              02751000 T  0024:0120:1
                     RESULT:=7; SCANNER;                                              02752000 T  0024:0121:3
                     IF EXAMIN(NCR) ≠ "(" THEN GO ARGH;                               02753000 T  0024:0123:2
                     RESULT:=0; SCANNER;  Q:=ACCUM[1];                                02754000 T  0024:0125:2
                     T:=SPECIAL[24]                                                   02755000 T  0024:0127:3
                     END;                                                             02756000 T  0024:0127:3
                RESULT:=2; GO COMPLETE;                                               02757000 T  0024:0128:1
IPART:          TCOUNT:=0; C:=CONVERT;                                                02758000 T  0024:0129:3
%               RESULT:=7; SCANNER; % DEBLANK.                                        02759000 T  0024:0131:3
%               IF DEFINECTR=0 THEN                                                   02760000 T  0024:0131:3
%                IF (C=3 OR C=4) AND EXAMIN(NCR)="""" THEN %OCTAL OR HEX STRING.      02761000 T  0024:0131:3
%                     BEGIN    INTEGER SIZ;                                        02762000 T  0024:0131:3
%                     RESULT:=5; SCANNER; % SKIP QUOTE.                            02763000 T  0024:0131:3
%                     COUNT:=0;                                                    02764000 T  0024:0131:3
%                     DO   BEGIN                                                   02765000 T  0024:0131:3
%                          RESULT:=5; SCANNER;                                     02766000 T  0024:0131:3
%                          IF COUNT > SIZ:=48 DIV C THEN % > 1 WORD LONG.          02767000 T  0024:0131:3
%                               BEGIN  ERR(520); GO SCANAGAIN  END;                02768000 T  0024:0131:3
%                          END UNTIL EXAMIN(NCR)="""";                             02769000 T  0024:0131:3
%                     Q:=ACCUM[1]; RESULT:=5; SCANNER; COUNT:=COUNT-1;             02770000 T  0024:0131:3
%                     IF C=3 THEN % OCTAL STRING.                                  02771000 T  0024:0131:3
%                       IF OCTIZE(ACCUM[1],ACCUM[4],16-COUNT,COUNT) THEN           02772000 T  0024:0131:3
%                         FLAG(521) % NON OCTAL CHARACTER IN STRING.               02773000 T  0024:0131:3
%               ELSE ELSE IF HEXIZE(ACCUM[1],ACCUM[4],12-COUNT,COUNT) THEN         02774000 T  0024:0131:3
%                          FLAG(521); % NON CHARACTER IN HEX STRING.               02775000 T  0024:0131:3
%                     IF COUNT < SIZ THEN                                          02776000 T  0024:0131:3
%                          BEGIN                                                   02777000 T  0024:0131:3
%                          C:=ACCUM[4]; GO FINISHNUMBER;                           02778000 T  0024:0131:3
%                          END;                                                    02779000 T  0024:0131:3
```

```
%             T.INCR:=COUNT:=8; T.CLASS:=STRING;                                  02780000 T  0024:0131:3
%             MOVECHARACTERS(8,ACCUM[4],0,ACCUM[1],3);                            02781000 T  0024:0131:3
%             GO COMPLETE;                                                        02782000 T  0024:0131:3
%             END OCTAL OR HEX STRING;                                            02783000 T  0024:0131:3
              IF DPTOG THEN                                                       02784000 T  0024:0131:3
                   BEGIN NHI:=THI; NLO:=TLO; END;                                 02785000 T  0024:0132:0
              IF EXAMIN(NCR)="," THEN                                             02786000 T  0024:0134:0
                   BEGIN                                                          02787000 T  0024:0135:3
                   RESULT:=0; SCANNER;                                            02788000 T  0024:0136:0
                   C:=1.0× C;                                                     02789000 T  0024:0137:3
FPART:             TCOUNT:=COUNT;                                                 02790000 T  0024:0138:1
                   IF EXAMIN(NCR)≤9 THEN                                          02791000 T  0024:0139:3
                        BEGIN                                                     02792000 T  0024:0141:3
                        RESULT:=0; SCANNER;                                       02793000 T  0024:0142:0
                        IF DPTOG THEN                                             02794000 T  0024:0143:2
                             BEGIN                                                02795000 T  0024:0143:3
                             DOUBLE(CONVERT,TLO,TEN[(COUNT-TCOUNT)MOD 12],        02796000 T  0024:0144:0
                                                  0,/,:=,THI,TLO);                02797000 T  0024:0147:2
                             FOR T:=12 STEP 12 UNTIL COUNT = TCOUNT DO            02798000 T  0024:0148:1
                               DOUBLE(THI,TLO,TEN[12],0,/,:=,THI,TLO);            02799000 T  0024:0153:2
                             DOUBLE(THI,TLO,NHI,NLO,+,:=,NHI,NLO);                02800000 T  0024:0158:0
                             C:=NHI                                               02801000 T  0024:0160:0
                             END                                                  02802000 T  0024:0160:0
                        ELSE C:=TEN[TCOUNT-COUNT]×CONVERT+C;                      02803000 T  0024:0161:2
                        END                                                       02804000 T  0024:0164:1
                   END;                                                           02805000 T  0024:0164:1
              RESULT:=7; SCANNER;                                                 02806000 T  0024:0164:1
              IF EXAMIN(NCR)="@" THEN                                             02807000 T  0024:0166:0
                   BEGIN                                                          02808000 T  0024:0167:3
FPART:             RESULT:=0; SCANNER;                                            02809000 T  0024:0168:0
                   TCOUNT:=COUNT;                                                 02810000 T  0024:0169:3
                   C:=C×1.0;                                                      02811000 T  0024:0170:1
                   RESULT:=7; SCANNER;                                            02812000 T  0024:0172:0
                   IF T:=EXAMIN(NCR)>9 THEN                                       02813000 T  0024:0173:2
                        BEGIN                                                     02815000 T  0024:0175:3
                        RESULT:=0; SCANNER;                                       02816000 T  0024:0176:0
                        TCOUNT:=COUNT;                                            02817000 T  0024:0177:2
                        END;                                                      02818000 T  0024:0178:0
                   RESULT:=0; SCANNER;                                            02820000 T  0024:0178:0
                   Q:=ACCUM[1];                                                   02822000 T  0024:0179:2
                   IF GT1:=T:=(IF T="-"THEN -CONVERT ELSE CONVERT)<-46 OR         02823000 T  0024:0180:0
                     T>69 THEN FLAG(269)                                          02824000 T  0024:0185:2
                   ELSE BEGIN                                                     02825000 T  0024:0186:1
                        T:=TEN[T];                                                02826000 T  0024:0189:2
                        IF ABS(0&C[42:3:6]&C[1:2:1]+0&T[42:3:6]&T[1:2:1]          02827000 T  0024:0190:1
                          + 12) >63 THEN FLAG(269)                                02828000 T  0024:0195:2
                   ELSE IF DPTOG THEN                                             02829000 T  0024:0197:2
                     IF GT1<0 THEN                                                02830000 T  0024:0198:1
                        BEGIN                                                     02831000 T  0024:0199:3
                        GT1:=-GT1;                                                02832000 T  0024:0200:0
                        DOUBLE(NHI,NLO,TEN[GT1 MOD 12],0,/,:=,NHI,NLO);           02833000 T  0024:0201:2
                        FOR GT2:=12 STEP 12 UNTIL GT1 DO                          02834000 T  0024:0205:2
                          DOUBLE(NHI,NLO,TEN[12],0,/,:=,NHI,NLO);                 02835000 T  0024:0206:0
                        END                                                       02836000 T  0024:0211:3
                     ELSE BEGIN                                                   02837000 T  0024:0211:3
                        DOUBLE(NHI,NLO,TEN[GT1 MOD 12],0,×,:=,NHI,NLO);           02838000 T  0024:0212:0
                        FOR GT2:=12 STEP 12 UNTIL GT1 DO                          02839000 T  0024:0215:3
```

```
                                DOUBLE( NHI,NLO,TEN[12],0,x,:=,NHI,NLO);
                              END
                        ELSE C:=CxT;
                            END;
                    END;
NUMBERREND:
            Q:=ACCUM[1]; RESULT:=3;
FINISHNUMBER:
            T:=0;
            IF C.[1:37]=0 THEN
                 BEGIN   T.CLASS:=LITNO ; T.ADDRESS:=C  END
            ELSE T.CLASS:=NONLITNO ;
            GO COMPLETE;
COMMENT     THE CODE BETWEEN IDENT AND COMPOST DOES A LOOKUP IN INFO.
            IF QUANTITY IS NOT FOUND THE ELBAT WORD EXPECTS TO BE
            ZERO.  THE SCRAMBLE FOR APPROPRIATE STACK IS FIRST THING
            TO BE DONE. THEN THE LOOP BETWEEN COMPOST AND
            ROSE IS ENTERED.  THE LAST THING DONE FOR ANY
            IDENTIFIER WHICH IS FOUND IS TO STUFF THE LOCATION
            OF THE ELBATWORD IN INFO INTO THE LINK FIELD.  THIS
            ALLOWS REFERENCE BACK TO INFO FOR ADDITIONAL DATA,
            SHOULD THIS BE REQUIRED.                              ;
IDENT:      T:=STACKHEADISCRAM:=(Q:=ACCUM[1])MOD 125];
ROSE:       GT1:=T.LINKR;
            IF(GT2:=T.LINKC)+GT1= 0 THEN
                BEGIN T:=0; GO COMPLETE END;
            IF T = INFO[GT1, GT2] THEN BEGIN                      %101-
              T:= 0; GO TO COMPLETE END;                          %101-
            T:=INFO[GT1,GT2];
            IF INFO[GT1,GT2+1]&O[1:1:11] ≠ Q THEN GO ROSE;
            IF COUNT ≤ 5 THEN GO COMPOST ;
            IF NOT EQUAL(COUNT-5,ACCUM[2],INFO[GT1,GT2+2])THEN GO ROSE;
COMPOST:    T:=T&GT1[35:43:5]&GT2[40:40:8];
            IF GT1 ≠1 AND NOT MACROID THEN % NOT RESERVED WORD    %110-
              XREFIT(T.LINK,CARDNUMBER,NORMALREF); % BUILD XREF ENTRY %110-
COMMENT     CHECK HERE FOR COMMENTS AND DEFINED IDS;
            IF NOT ENDTOG THEN
                 BEGIN
                 IF GT1:=T.CLASS = COMMENTV THEN
                      BEGIN
                      WHILE EXAMIN(NCR) ≠ ";" DO
                           BEGIN RESULT:=6; COUNT:=0; SCANNER END;
                      RESULT:=0;SCANNER;GO SCANAGAIN
                      END
                 END;
            IF STOPDEFINE THEN GO COMPLETE;
            IF GT1 ≠ DEFINEDID THEN GO COMPLETE;
COMMENT     SETUP FOR DEFINED IDS - SEE DEFINEGEN FOR MORE DETAILS;
            IF T.ADDRESS≠0 THEN T:=FIXDEFINEINFO(T);
            IF DEFINEINDEX = 24 THEN
                 BEGIN FLAG(139);GO ARGH END;
            DEFINEARRAY[DEFINEINDEX]:=LASTUSED&T.ADDRESS [18:33:15];
            LASTUSED:=GIT(T);
            DEFINEARRAY[DEFINEINDEX+2]:=262144xLCR+NCR;
            LCR:=(NCR:=MKABS(DEFINEARRAY[DEFINEINDEX+1]))+1;
            PUTSEQNO(GT4,LCR);
            TURNONSTOPLIGHT("%",LCR); DEFINEINDEX:=DEFINEINDEX+3;
```

```
02840000 T  0024:0217:2
02841000 T  0024:0222:1
02842000 T  0024:0222:1
02843000 T  0024:0224:0
02844000 T  0024:0224:0
02845000 T  0024:0224:0
02846000 T  0024:0225:2
02847000 T  0024:0226:1
02848000 T  0024:0227:2
02849000 T  0024:0227:3
02850000 T  0024:0229:3
02851000 T  0024:0233:3
02852000 T  0024:0235:3
02853000 T  0024:0236:0
02854000 T  0024:0236:0
02855000 T  0024:0236:0
02859000 T  0024:0236:0
02860000 T  0024:0236:0
02861000 T  0024:0236:0
02862000 T  0024:0236:0
02863000 T  0024:0236:0
02864000 T  0024:0236:0
02865000 T  0024:0236:0
02875000 T  0024:0239:3
02876000 T  0024:0241:2
02877000 T  0024:0243:3
02877010 C  0024:0245:2
02877020 C  0024:0247:3
02878000 T  0024:0248:1
02879000 T  0024:0250:1
02880000 T  0024:0254:0
02881000 T  0024:0255:3
02882000 T  0024:0260:0
02882100 C  0024:0262:1
02882200 C  0024:0264:0
02883000 T  0024:0267:3
02884000 T  0024:0267:3
02885000 T  0024:0268:0
02886000 T  0024:0268:1
02887000 T  0024:0270:1
02888000 T  0024:0271:2
02889000 T  0024:0273:2
02890000 T  0024:0275:3
02891000 T  0024:0277:3
02892000 T  0024:0277:3
02893000 T  0024:0277:3
02894000 T  0024:0278:1
02895000 T  0024:0279:3
02896000 T  0024:0279:3
02898000 T  0024:0282:1
02899000 T  0024:0283:3
02900000 T  0024:0285:2
02901000 T  0024:0287:3
02902000 T  0024:0288:1
02903000 T  0024:0291:3
02904000 T  0024:0295:2
02905000 T  0024:0296:0
```

```
                GO PERCENT;                                                                    02906000 T  0024:0298:0
COMPLETE:                                                                                      02909000 T  0024:0300:0
                ELBAT[NXTELBT]:=T;                                                             02910000 T  0024:0300:0
        IF NOT DEFINING THEN                                                    %108-          02910100 C  0024:0301:2
        IF T.CLASS = BEGINV THEN                                                %108-          02910200 C  0024:0301:3
                BEGINSTACK[BSPOINT:=BSPOINT+1]:=CARDNUMBER ELSE                 %108-          02910300 C  0024:0303:3
        IF T.CLASS = ENDV THEN                                                  %108-          02910400 C  0024:0306:0
      BEGIN                                                                     %108-          02910500 C  0024:0308:0
                IF LISTER THEN IF BEND THEN BEGINPRINT;                         %108-          02910600 C  0024:0308:1
                BSPOINT:=BSPOINT - REAL(BSPOINT > 0); % PREVENT INVALID INDEX                  02910700 C  0024:0311:2
      END;                                                                      %108-          02910800 C  0024:0312:1
                STOPDEFINE:=FALSE; COMMENT ALLOW DEFINES AGAIN;                                02911000 T  0024:0312:1
                IF NXTELBT:=NXTELBT+1 > 74 THEN                                                02912000 T  0024:0313:3
                  IF NOT MACROID THEN                                                          02913000 T  0024:0315:2
                    BEGIN                                                                      02914000 T  0024:0316:0
COMMENT         ELBAT IS FULL:  ADJUST IT;                                                     02915000 T  0024:0316:1
                    MOVE(10,ELBAT[65],ELBAT);                                                  02916000 T  0024:0316:1
                    I:=I-65; P:=P-65; NXTELBT:=10;                                             02917000 T  0024:0318:1
                    END                                                                        02918000 T  0024:0321:3
                END;                                                                           02919000 T  0024:0321:3
        IF TABLE:=ELBAT[P].CLASS = COMMENTV THEN                                               02920000 T  0024:0322:0
                BEGIN                                                                          02921000 T  0024:0325:2
COMMENT         SPECIAL HANDLING OF CONSTANTS FOR SAKE OF FOR STATEMENTS;                      02922000 T  0024:0325:3
                C:=INFO[0,ELBAT[P].ADDRESS];                                                   02923000 T  0024:0325:3
                ELBAT[P].CLASS:=TABLE:=NONLITNO                                                02924000 T  0024:0328:0
                END;                                                                           02925000 T  0024:0329:2
        STOPDEFINE:=FALSE;  COMMENT ALLOW DEFINE;                                              02926000 T  0024:0331:2
        END TABLE ;                                                                            02927000 T  0024:0331:3

                                                                                        24 IS  337 LONG, NEXT SEG    3
        BOOLEAN PROCEDURE BOOLPRIM;       FORWARD;                                             02955000 T  0003:0242:0
        PROCEDURE BOOLCOMP(B); BOOLEAN B; FORWARD;                                             02955500 T  0003:0242:0
        INTEGER PROCEDURE NEXT;                                                                02956000 T  0003:0242:0
             BEGIN                                                                             02956500 T  0003:0242:0
             LABEL EXIT;                                                                       02957000 T  0003:0242:0
                                                                                        START OF SEGMENT *********   25
             INTEGER T;                                                                        02957500 T  0025:0000:0
             DEFINE ERROR = BEGIN  FLAG(603); GO EXIT  END#;                                   02958000 T  0025:0000:0
             SKAN;                                                                             02958500 T  0025:0000:0
             IF RESULT=3 THEN ERROR; % NUMBERS NOT ALLOWED.                                    02959000 T  0025:0003:3
             IF RESULT=2 THEN % SPECIAL CHARACTER.                                             02959500 T  0025:0006:0
                  BEGIN                                                                        02960000 T  0025:0007:2
                  T:=IF Q="1,0000" OR Q="1%0000" THEN 20 % FAKE OUT BOOLEXP.                   02960500 T  0025:0007:3
                     ELSE ((T:=Q.[18:6]-2) & T[42:41:3]);                                      02961000 T  0025:0010:0
                  IF T=11 OR T=19 OR T=20 THEN BATMAN:=SPECIAL[T] % (,),OR ;                   02961500 T  0025:0013:3
             ELSE FLAG(603);                                                                   02962000 T  0025:0017:2
                  GO EXIT                                                                      02962500 T  0025:0021:3
                  END SPECIAL CHARACTERS;                                                      02963000 T  0025:0022:0
COMMENT      LOOK FOR BOOLEAN OPERATORS, THEN OPTIONS;                                         02963500 T  0025:0022:0
             T:=       IF Q="3NOT00" THEN NOTOP                                                02964000 T  0025:0022:0
                  ELSE IF Q="3AND00" THEN ANDOP                                                02964500 T  0025:0023:3
                  ELSE IF Q="2OR000" THEN OROP                                                 02965000 T  0025:0025:3
                  ELSE IF Q="3EQV00" THEN EQVOP                                                02965500 T  0025:0027:3
                  ELSE 0;                                                                      02966000 T  0025:0029:3
             IF T#0 THEN BATMAN.CLASS:=T                                                       02966500 T  0025:0031:2
        ELSE BATMAN:=1 & BOOID[2:7] & REAL(FINDOPTION(1))[1:1]; % OPTION.                      02967000 T  0025:0032:1
EXIT:                                                                                          02967500 T  0025:0042:0
             NEXT:=MYCLASS:=BATMAN.CLASS;                                                      02968000 T  0025:0043:2
```

```
END NEXT;

BOOLEAN PROCEDURE BOOLEXP;
     BEGIN
     BOOLEAN B;

     B:=BOOLPRIM;
     WHILE MYCLASS≥EQVOP AND MYCLASS≤ANDOP DO BOOLCOMP(B);
     BOOLEXP:=B
     END BOOLEXP;

BOOLEAN PROCEDURE BOOLPRIM;
     BEGIN
     BOOLEAN B,KNOT;

     DEFINE SKIPIT = MYCLASS:=NEXT #;
     IF KNOT:=(NEXT=NOTOP) THEN SKIPIT;
     IF MYCLASS=LEFTPAREN THEN
          BEGIN
          B:=BOOLEXP;
          IF MYCLASS≠RTPAREN THEN FLAG(604);
          END
ELSE IF MYCLASS≠BOOID THEN FLAG(601)
ELSE B:=BATMAN<0;
     IF KNOT THEN B:=NOT B; SKIPIT;
     BOOLPRIM:=B
     END BOOLPRIM;

    PROCEDURE BOOLCOMP(B); BOOLEAN B;
         BEGIN
         REAL OPCLASS;

         BOOLEAN T;
         OPCLASS:=MYCLASS;
         T:=BOOLPRIM;
         WHILE OPCLASS<MYCLASS DO BOOLCOMP(T);
         B:=     IF OPCLASS=ANDOP THEN (B AND T)
            ELSE IF OPCLASS=OROP  THEN (B OR T)
            ELSE                       (B EQV T);
         END BOOLCOMP;

%
COMMENT##########################################################################
               FORWARD DECLARATIONS
##########################################################################;
%
 PROCEDURE AEXP; FORWARD;
 PROCEDURE ARITHSEC; FORWARD;
 PROCEDURE SIMPARITH; FORWARD;
 PROCEDURE ARITHCOMP; FORWARD;
 PROCEDURE PRIMARY; FORWARD;
 DEFINE BEXP = AEXP#;
 INTEGER PROCEDURE EXPRSS; FORWARD;
 PROCEDURE POLISHER(EXPECT); VALUE EXPECT; REAL EXPECT; FORWARD;
 PROCEDURE INLINE; FORWARD;
 PROCEDURE SUBHAND(FROM); VALUE FROM; BOOLEAN FROM; FORWARD;
 PROCEDURE IOSTMT; FORWARD;
```

```
          02968500 T  0025:0044:1
  25 IS   48 LONG, NEXT SEG    3
          02969000 T  0003:0242:0
          02969500 T  0003:0242:0
          02970000 T  0003:0242:0
 START OF SEGMENT **********   26
          02970500 T  0026:0000:0
          02971000 T  0026:0001:2
          02971500 T  0026:0004:1
          02972000 T  0026:0004:1
  26 IS    8 LONG, NEXT SEG    3
          02972500 T  0003:0242:0
          02973000 T  0003:0242:0
          02973500 T  0003:0242:0
 START OF SEGMENT **********   27
          02974000 T  0027:0000:0
          02974500 T  0027:0000:0
          02975000 T  0027:0003:2
          02975500 T  0027:0003:3
          02976000 T  0027:0004:0
          02976500 T  0027:0005:2
          02977000 T  0027:0007:2
          02977500 T  0027:0007:2
          02978000 T  0027:0009:2
          02978500 T  0027:0011:3
          02979000 T  0027:0014:0
          02979500 T  0027:0014:0
  27 IS   18 LONG, NEXT SEG    3
          02980000 T  0003:0242:0
          02980500 T  0003:0242:0
          02981000 T  0003:0242:0
 START OF SEGMENT **********   28
          02981500 T  0028:0000:0
          02982000 T  0028:0000:0
          02982500 T  0028:0000:1
          02983000 T  0028:0001:3
          02983500 T  0028:0004:1
          02984000 T  0028:0006:0
          02984500 T  0028:0008:1
          02985000 T  0028:0011:2
  28 IS   14 LONG, NEXT SEG    3
          02985500 T  0003:0242:0
          02986000 T  0003:0242:0
          02986500 T  0003:0242:0
          02987000 T  0003:0242:0
          02987500 T  0003:0242:0
          03001000 T  0003:0242:0
          03002000 T  0003:0242:0
          03003000 T  0003:0242:0
          03004000 T  0003:0242:0
          03005000 T  0003:0242:0
          03006000 T  0003:0242:0
          03007000 T  0003:0242:0
          03009000 T  0003:0242:0
          03010000 T  0003:0242:0
          03011000 T  0003:0242:0
          03012000 T  0003:0242:0
```

```
 INTEGER PROCEDURE IFEXP; FORWARD;                                                03013000 T  0003:024:0
 PROCEDURE PARSE; FORWARD;                                                        03014000 T  0003:024:0
 PROCEDURE DOT; FORWARD;                                                          03015000 T  0003:024:0
 PROCEDURE IFCLAUSE; FORWARD;                                                     03018000 T  0003:024:0
INTEGER PROCEDURE GET(SYLLABLE);VALUE SYLLABLE; REAL SYLLABLE; FORWARD;           03019000 T  0003:024:0
INTEGER PROCEDURE GNAT(L); VALUE L; REAL L; FORWARD;                              03020000 T  0003:024:0
 PROCEDURE PANA; FORWARD;                                                         03021000 T  0003:024:0
 PROCEDURE IFSTMT; FORWARD;                                                       03022000 T  0003:024:0
 PROCEDURE GOGEN(LABELBAT,BRANCHTYPE);                                            03023000 T  0003:024:0
              VALUE LABELBAT,BRANCHTYPE;                                          03024000 T  0003:024:0
              REAL LABELBAT,BRANCHTYPE; FORWARD;                                  03025000 T  0003:024:0
 BOOLEAN PROCEDURE SIMPGO; FORWARD;                                               03026000 T  0003:024:0
 PROCEDURE STMT; FORWARD;                                                         03027000 T  0003:024:0
 PROCEDURE EMIT(SYLLABLE); VALUE SYLLABLE; REAL SYLLABLE; FORWARD;                03028000 T  0003:024:0
 PROCEDURE PROCSTMT(FROM); VALUE FROM; BOOLEAN FROM; FORWARD;                     03029000 T  0003:024:0
 PROCEDURE STRMPROCSTMT; FORWARD;                                                 03030000 T  0003:024:0
 PROCEDURE CONSTANTCLEAN; FORWARD;                                                03034000 T  0003:024:0
 PROCEDURE SCATTERFLBAT; FORWARD;                                                 03035000 T  0003:024:0
 PROCEDURE EMITB(BRANCH,FROM,TOWARDS); VALUE BRANCH,FROM,TOWARDS;                 03036000 T  0003:024:0
              INTEGER BRANCH,FROM,TOWARDS; FORWARD;                               03037000 T  0003:024:0
 PROCEDURE VARIABLE(FROM); INTEGER FROM; FORWARD;                                 03038000 T  0003:024:0
 PROCEDURE IMPFUN; FORWARD;                                                       03039000 T  0003:024:0
PROCEDURE RIGHT(L); VALUE L; INTEGER L; FORWARD;                                  03039500 T  0003:024:0
 PROCEDURE STREAMSTMT; FORWARD;                                                   03040000 T  0003:024:0
PROCEDURE SEGMENTSTART(B);VALUE B;BOOLEAN B;FORWARD;                              03041000 T  0003:024:0
PROCEDURE SEGMENT(SIZE,FR); VALUE SIZE,FR; INTEGER SIZE,FR; FORWARD;              03042000 T  0003:024:0
                                                                                  03043000 T  0003:024:0
                                                                                  03044000 T  0003:024:0
                                                                                  03045000 T  0003:024:0
INTEGER PROCEDURE BAE; FORWARD;                                                   03046000 T  0003:024:0
     PROCEDURE PROGDESCBLDR(A,B,C,D);VALUE A,B,C,D;                               03047000 T  0003:024:0
                  INTEGER A,C,D; BOOLEAN B; FORWARD;                              03047100 T  0003:024:0
 PROCEDURE BANA; FORWARD;                                                         03048000 T  0003:024:0
 PROCEDURE EMITNUM(A); VALUE A; REAL A; FORWARD;                                  03049000 T  0003:024:0
 PROCEDURE EMITD(A,B,T); VALUE A,B,T; INTEGER A,B,T; FORWARD;                     03050000 T  0003:024:0
INTEGER PROCEDURE GETSPACE(S,L); VALUE S,L;                                       03051000 T  0003:024:0
                                 INTEGER L; BOOLEAN S; FORWARD;                   03051001 T  0003:024:0
 PROCEDURE FORSTMT; FORWARD;                                                      03052000 T  0003:024:0
REAL PROCEDURE TAKE(INDEX); VALUE INDEX; INTEGER INDEX; FORWARD;                  03053000 T  0003:024:0
PROCEDURE E;  FORWARD;                                                            03054000 T  0003:024:0
PROCEDURE ENTRY(TYPE); VALUE TYPE;REAL TYPE; FORWARD;                             03055000 T  0003:024:0
PROCEDURE PUTNBUMP(P1);  VALUE P1;  REAL P1;  FORWARD;                            03057000 T  0003:024:0
PROCEDURE JUMPCHKNX;  FORWARD;                                                    03058000 T  0003:024:0
PROCEDURE JUMPCHKX;  FORWARD;                                                     03059000 T  0003:024:0
 PROCEDURE DBLSTMT; FORWARD;                                                      03060000 T  0003:024:0
 PROCEDURE BLOCK(S); VALUE S; BOOLEAN S; FORWARD;                                 03067000 T  0003:024:0
PROCEDURE PURGE(STOPPER);VALUE STOPPER;REAL STOPPER;FORWARD;                      03068000 T  0003:024:0
PROCEDURE ENTER(TYPEV);                                                           03069000 T  0003:024:0
          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  0003:024:0
                                                                                  03075000 T  0003:024:0
 COMMENT THIS SECTION CONTAINS THE EMITTERS.  THEY ARE THE AGENTS WHICH           04000000 T  0003:024:0
              ACTUALLY PRODUCE CODE AND DEBUGING OUTPUT;                          04001000 T  0003:024:0
 COMMENT EMITL EMITS A LIT CALL;                                                  04002000 T  0003:024:0
```

```
                PROCEDURE EMITL(LITERAL); VALUE LITERAL; INTEGER LITERAL;                04003000 T  0003:0242:0
                        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
                        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
                        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
                        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
                        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
                        VALUE ADDRESS,OPERATOR;                                          04029000 T  0003:0258:0
                        INTEGER ADDRESS,OPERATOR;                                        04030000 T  0003:0258:0
                        BEGIN                                                            04031000 T  0003:0258:0
                            EMITL(ADDRESS);                                              04032000 T  0003:0258:0
                            IF ADDRESS > 1023 THEN EMITO(PRTE);                          04033000 T  0003:0258:1
                            EMITO(OPERATOR) END EMITPAIR;                                04034000 T  0003:0260:1
                COMMENT ADJUST ADJUST L TO THE BEGINING OF A WORD AND FILLS IN THE       04080000 T  0003:0261:3
                        INERVENING SPACE WITH NOPS, IT CHECKS STREAMTOG TO DECIDE        04081000 T  0003:0261:3
                        WHICH SORT OF NOP TO USE;                                        04082000 T  0003:0261:3
                PROCEDURE ADJUST;                                                        04083000 T  0003:0261:3
                        BEGIN                                                            04084000 T  0003:0261:3
                                                                                         04085000 T  0003:0261:3
              WHILE L.[46:2]≠0 DO EMIT(45);                                              04086000 T  0003:0261:3
                        END ADJUST;                                                      04087000 T  0003:0265:2
                PROCEDURE EMITLNG;                                                       04098000 T  0003:0265:2
                        BEGIN LABEL E;                                                   04099000 T  0003:0265:2
                                                                                 START OF SEGMENT *********   29
                          IF NOT LINKTOG THEN GO TO E;                                   04100000 T  0029:0000:0
                   COMMENT  GO TO E IF LAST THING IS A LINK;                             04101000 T  0029:0000:1
                          IF GET(L) ≠ 0 THEN GO TO E;                                    04102000 T  0029:0000:1
                   COMMENT  EITHER LAST EXPRESSION WAS CONDITIONAL OR THERE IS NO        04103000 T  0029:0002:1
                        LNG OR RELATIONAL OPERATOR;                                      04104000 T  0029:0002:1
                          IF GT1 ← GET(L-1) = 77 THEN L ← L-1                            04105000 T  0029:0002:1
                   COMMENT LAST THING WAS AN LNG - SO CANCEL IT;                         04106000 T  0029:0005:3
                       ELSE IF GT1.[42:6]=21 AND GT1.[37:2]=0 THEN % AHA                 04107000 T  0029:0005:3
                   COMMENT LAST THING WAS A RELATIONAL;                                  04108000 T  0029:0009:3
                          BEGIN L←L-1; EMITO(REAL(BOOLEAN(GT1.[36:10]) EQV               04109000 T  0029:0009:3
                            BOOLEAN(IF GT1.[40:2] = 0 THEN 511 ELSE 463)))               04110000 T  0029:0012:1
                   COMMENT NEGATE THE RELATIONAL; END ELSE                               04111000 T  0029:0015:3
                E:        EMITO(LNG) END EMITLNG;                                        04112000 T  0029:0015:3
                                                                                29 IS    18 LONG, NEXT SEG    3
```

```
   COMMENT   EMITB EMITS A BRANCH OPERATOR AND ITS ASSOCIATED NUMBER;
PROCEDURE EMITB(BRANCH,FROM,TOWARDS);
          VALUE BRANCH,FROM,TOWARDS;
          INTEGER BRANCH,FROM,TOWARDS;
          BEGIN
            INTEGER TL;

            TL ← L;
          IF TOWARDS > FOULED THEN FOULED ← TOWARDS;
            L ← FROM-2;
            GT1 ← TOWARDS-FROM;
            IF TOWARDS.[46:2] = 0
              THEN BEGIN
                BRANCH ← BRANCH&1[39:47:1];
                GT1 ← TOWARDS DIV 4 - (FROM-1) DIV 4 END;
            EMITNUM(ABS(GT1));
            EMITO(BRANCH&(REAL(GT1≥ 0)+1)[42:46:2]);

            L ← TL
          END EMITB;

   COMMENT   DEBUGWORD FORMATS TWO FIELDS FOR DEBUGGING OUTPUT IN
          OCTAL, NAMELY :
              1. 4 CHARACTERS FOR THE L REGISTER.
              2.16 CHARACTERS FOR THE WORD BEING EMITTED.    ;
STREAM  PROCEDURE DEBUGWORD( SEQ,CODE,FEIL); VALUE SEQ,CODE ;
          BEGIN
            DI←FEIL; SI← LOC SEQ; SI← SI+4; DS ← 4 CHR;
            DS ← 2 LIT"  ";
            SI ← LOC CODE ;
            16(  DS ← 3 RESET; 3( IF SB THEN DS←SET ELSE
                 DS ← RESET ; SKIP 1 SB));
          29(DS ← 2 LIT"  " );
           END ;
   COMMENT   EMITWORD PLACES THE PARAMETER,"WORD",INTO EDOC. IF
          DEBUGGING IS REQUIRED, "L" AND "WORD" ARE OUTPUT ON
          THE  PRINTER FILE IN OCTAL FORMAT.     ;
PROCEDURE  EMITWORD (WORD); VALUE WORD; REAL WORD;
          BEGIN
          ADJUST;
          IF L≥ 4088 THEN BEGIN ERR(200); L←0; END
              ELSE BEGIN
      MOVE(1,WORD, CODE(L DIV 4+1));
              IF DEBUGTOG THEN
              BEGIN DEBUGWORD(B2D(L),WORD,LIN);
                  WRITELINE  END;
              FOULED ← L ← L+4; END
          END  EMITWORD;
   COMMENT CONSTANTCLEAN IS CALLED AFTER AN UNCONDITIONAL BRANCH HAS
          BEEN EMITTED. IF ANY CONSTANTS HAVE BEEN ACCUMULATED BY
          EMITNUM IN INFO[0,*], CONSTANTCLEAN WILL FIX  THE CHAIN
          OF  C-RELATIVE  OPDC S LEFT BY EMITNUM. IF  C-RELATIVE
          ADDRESSING IS IMPOSSIBLE (I.E. THE ADDRESS
          IF GREATER THAN 127 WORDS) THEN THE CONSTANT ALONG WITH
          THE  1ST  LINK OF THE OPDC CHAIN IS ENTERED IN INFO.
          AT  PURGE  TIME THE REMAINING OPDC S ARE EMITTED WITH
          F -RELATIVE ADDRESSING AND CODE EMITTED TO STORE THE
          CONSTANTS INTO THE PROPER F-RELATIVE CELLS.  ;
```

```
                                                                  04113000 T  0003:0265:2
                                                                  04114000 T  0003:0265:2
                                                                  04115000 T  0003:0265:2
                                                                  04116000 T  0003:0265:2
                                                                  04117000 T  0003:0265:2
                                                                  04118000 T  0003:0265:2
                                                          START OF SEGMENT *********   30
                                                                  04119000 T  0030:0000:0
                                                                  04119500 T  0030:0000:1
                                                                  04120000 T  0030:0002:1
                                                                  04120100 T  0030:0004:0
                                                                  04120200 T  0030:0005:2
                                                                  04120300 T  0030:0006:0
                                                                  04120400 T  0030:0007:2
                                                                  04120500 T  0030:0008:1
                                                                  04121000 T  0030:0011:3
                                                                  04122000 T  0030:0012:1
                                                                  04123000 T  0030:0015:2
                                                                  04124000 T  0030:0015:2
                                                                  04125000 T  0030:0015:2
                                                           30 IS   19 LONG, NEXT SEG    3
                                                                  04126000 T  0003:0265:2
                                                                  04127000 T  0003:0265:2
                                                                  04128000 T  0003:0265:2
                                                                  04129000 T  0003:0265:2
                                                                  04130000 T  0003:0265:2
                                                                  04131000 T  0003:0265:2
                                                                  04132000 T  0003:0266:0
                                                                  04133000 T  0003:0267:2
                                                                  04134000 T  0003:0267:3
                                                                  04135000 T  0003:0267:3
                                                                  04136000 T  0003:0269:3
                                                                  04137000 T  0003:0270:1
                                                                  04138000 T  0003:0271:3
                                                                  04139000 T  0003:0271:3
                                                                  04140000 T  0003:0271:3
                                                                  04141000 T  0003:0271:3
                                                                  04142000 T  0003:0271:3
                                                                  04143000 T  0003:0271:3
                                                                  04144000 T  0003:0271:3
                                                                  04145000 T  0003:0272:1
                                                                  04146000 T  0003:0275:2
                                                                  04147000 T  0003:0277:2
                                                                  04148000 T  0003:0282:0
                                                                  04149000 T  0003:0282:1
                                                                  04150000 T  0003:0285:2
                                                                  04151000 T  0003:0295:3
                                                                  04152000 T  0003:0297:2
                                                                  04153000 T  0003:0297:3
                                                                  04154000 T  0003:0297:3
                                                                  04155000 T  0003:0297:3
                                                                  04156000 T  0003:0297:3
                                                                  04157000 T  0003:0297:3
                                                                  04158000 T  0003:0297:3
                                                                  04159000 T  0003:0297:3
                                                                  04160000 T  0003:0297:3
                                                                  04161000 T  0003:0297:3
                                                                  04162000 T  0003:0297:3
```

```
PROCEDURE CONSTANTCLEAN ;                                                                    04163000 T  0003:0297:3
          IF MRCLEAN THEN                                                                    04164000 T  0003:0297:3
            BEGIN                                                                            04165000 T  0003:0298:0
               INTEGER J,TEMPL,D,LINK;                                                       04166000 T  0003:0298:1
                                                                                  START OF SEGMENT *********   31
               BOOLEAN CREL;                                                                 04167000 T  0031:0000:0
               LABEL  ALLTHU ;                                                               04168000 T  0031:0000:0
                                                                                             04169000 T  0031:0000:0
               FOR J ← 1 STEP 2 UNTIL LASTENTRY DO                                           04170000 T  0031:0000:0
                 BEGIN                                                                       04171000 T  0031:0001:2
                 ADJUST; TEMPL←L; L←INFO[0,255-J+1];                                         04172000 T  0031:0001:2
                    CREL ← FALSE;                                                            04173000 T  0031:0005:2
                    DO BEGIN                                                                 04174000 T  0031:0005:3
                        IF D←(TEMPL-L+3)DIV 4≥128 THEN                                       04175000 T  0031:0006:0
                 IF MODE ≠ 0 THEN                                                            04175500 T  0031:0008:1
BEGIN  FLAG(50);       GO TO ALLTHU END;                                                     04176000 T  0031:0010:0
                                                                                             04177000 T  0031:0011:3
                                                                                             04178000 T  0031:0011:3
                                                                                             04179000 T  0031:0011:3
                                                                                             04180000 T  0031:0011:3
                                                                                             04181000 T  0031:0011:3
                    LINK←GET(L);                                                             04182000 T  0031:0011:3
                    CREL ← TRUE;                                                             04183000 T  0031:0013:2
                      IF MODE ≠ 0 THEN EMITV(D+768) ELSE                                     04184000 T  0031:0013:3
                      EMITV(REAL(TEMPL≥2048)×1024+TEMPL DIV 4);                              04184500 T  0031:0016:0
                  END  UNTIL L←   LINK    = 4095  ;                                          04185000 T  0031:0019:3
ALLTHU:       L ← TEMPL;                                                                     04186000 T  0031:0021:2
              IF  CREL  THEN  EMITWORD( INFO[0,255-J ]);                                     04187000 T  0031:0022:1
                END;                                                                         04188000 T  0031:0025:3
              LASTENTRY ← 0;                                                                 04189000 T  0031:0028:0
            END ;                                                                            04190000 T  0031:0028:1
                                                                                  31 IS    33 LONG, NEXT SEG    3
   COMMENT    EMITNUM HANDLES THE EMISSION OF CODE FOR CONSTANTS,BOTH                        04191000 T  0003:0303:2
                  EXPLICIT AND IMPLICIT, IN EVERY CASE,EMITNUM WILL                          04192000 T  0003:0303:2
                  PRODUCE CODE TO GET THE DESIRED CONSTANT ON TOP OF                         04193000 T  0003:0303:2
                  THE STACK. IF THE NUMBER IS A LITERAL A SIMPLE LITC                        04194000 T  0003:0303:2
                  SYLLABLE IS PRODUCED. HOWEVER,NON-LITERALS ARE KEPT                        04195000 T  0003:0303:2
                  IN THE ZERO-TH ROW OF INFO WITH THE SYLLABLE                               04196000 T  0003:0303:2
                  POSITION,L. THE FIRST EMITNUM ON A PARTICULAR                              04197000 T  0003:0303:2
                  CONSTANT CAUSES THE VALUES OF L AND THE CONSTANT                           04198000 T  0003:0303:2
                  TO BE STORED IN INFO[0,*] (NOTE:ITEMS ARE STORED                           04199000 T  0003:0303:2
                  IN REVERSE STARTING WITH INFO[0,255],ETC.), THEN                           04200000 T  0003:0303:2
                  ITS THE JOB OF CONSTANTCLEAN TO EMIT THE ACTUAL                            04201000 T  0003:0303:2
                  OPDC (SEE CONSTANTCLEAN PROCEDURE FOR DETAILS) ;                           04202000 T  0003:0303:2
PROCEDURE EMITNUM( C ); VALUE C; REAL C;                                                     04203000 T  0003:0303:2
              BEGIN LABEL FINISHED,FOUND ; REAL N;                                           04204000 T  0003:0303:2
                                                                                  START OF SEGMENT *********   32
              IF C[1:37]=0 THEN EMITL(C)                                                     04205000 T  0032:0000:0
              ELSE                                                                           04206000 T  0032:0002:1
                   BEGIN                                                                     04207000 T  0032:0003:2
                   FOULED ← L;                                                               04207500 T  0032:0003:3
                   FOR  N ← 1 STEP 2 UNTIL LASTENTRY DO                                      04208000 T  0032:0004:0
                IF INFO[0,255-N] = C THEN  GO TO FOUND ;                                     04209000 T  0032:0005:2
                   INFO[0,255 -LASTENTRY] ← L;                                               04210000 T  0032:0010:0
                   INFO[0,255 -LASTENTRY-1]← C ;                                             04211000 T  0032:0012:1
                   EMITN(1023);                                                              04212000 T  0032:0015:3
                   IF MODE=0 THEN EMITO(NOP);                                                04212100 T  0032:0016:0
```

```
                    LINKTOG←FALSE;
                    IF LASTENTRY ← LASTENTRY+2 ≥ 128  THEN
                         BEGIN
                         C ← BUMPL;
                         CONSTANTCLEAN;
                         EMITB(BFW,C,L);
                         END;
                    GO TO FINISHED;
      FOUND:        EMIT(INFO[0,255 -N+1]);
                    LINKTOG←FALSE;
                    INFO[0,255-N+1]← L-1;
                    IF MODE=0 THEN EMITO(NOP);
                    END;
   FINISHED:END     EMITNUM  ;

      COMMENT   SEARCH PERFORMS A BINARY SEARCH ON THE COP AND WOP
                ARRAYS. GIVEN THE OPERATOR BITS SEARCH  YIELDS THE BCD
                MNEUMONIC FOR THAT OPERATOR. IF THE OPERATOR CANNOT
                BE FOUND SEARCH YIELDS BLANKS.
                NOTE: DIA,DIB,TRB ARE RETURNED AS BLANKS.       ;
  ALPHA PROCEDURE   SEARCH (Q,KEY); VALUE KEY; ARRAY Q[0]; REAL KEY ;
                BEGIN  LABEL L;

      COMMENT   GT1 AND GT2 ARE INITIALIZED ASSUMMING THAT Q IS ORDERED
                BY PAIRS (ARGUMENT,FUNCTION,ARGUMENT,FUNCTION,ETC.)
                AND THAT THE FIRST ARGUMENT IS IN Q[4].  FURTHERMORE
                THE LENGTH OF Q IS 128.                          ;
                INTEGER N,I ;
                N ← 64 ;
                FOR I ← 66 STEP IF Q[I]<KEY THEN N ELSE - N
                          WHILE N←N DIV 2 ≥ 1 DO
                IF Q[ I ]= KEY THEN GO TO L ;
                 I ←0;   COMMENT ARGUMENT NOT FOUND,SEARCH=Q[1] ;
           L:   SEARCH←Q[  I +1] ;
                END       SEARCH  ;

         COMMENT   B2D  CONVERTS THE FOUR LOW ORDER OCTAL DIGITS TO BCD
                   CODE ;
      ALPHA   PROCEDURE   B2D(B); VALUE B; REAL B;
                   B2D←08B[45:45:3]&B[39:42:3]&B[33:39:3]&B[27:36:3] ;
         COMMENT   PACK IS A STREAM PROCEDURE WHICH INSERTS THE SYLLABLE
                   INTO THE EDOC ARRAY.   THE SPECIFIC ELEMENT OF EDOC
                   IS  PRECISILY = EDOC[(L DIV 4) DIV 128,(< DIV 4)MOD 128]
                   SYLLABLE  POSITION=(L MOD 4 ), WHERE L IS THE SYLLABLE
                   NUMBER RELATIVE TO THE BEGINNING OF THE SEGMENT;
      STREAM PROCEDURE PACK(WORD,POSITION,SYLLABLE);
                   VALUE POSITION,SYLLABLE;
                   BEGIN
                   DI←WORD ; DI ← DI+POSITION ; DI ← DI+POSITION;
                   SI←LOC SYLLABLE ; SI←SI+6;
                   DS←2 CHR ;
                   END  PACK  ;
   COMMENT DEBUG PRINTS OUT OBJECT CODE IF "DEBUGN" IS SET;
   PROCEDURE DEBUG(S); VALUE S; REAL S;
       BEGIN  REAL T1;

       IF SINGLTOG THEN
```

```
04213000 T  0032:0018:0
04214000 T  0032:0019:2
04215000 T  0032:0020:1
04216000 T  0032:0021:2
04217000 T  0032:0023:2
04218000 T  0032:0023:3
04219000 T  0032:0024:1
04220000 T  0032:0024:1
04221000 T  0032:0025:2
04222000 T  0032:0028:1
04223000 T  0032:0029:3
04223100 T  0032:0033:2
04224000 T  0032:0035:2
04225000 T  0032:0035:2
32 IS   38 LONG, NEXT SEG    3
04226000 T  0003:0303:2
04227000 T  0003:0303:2
04228000 T  0003:0303:2
04229000 T  0003:0303:2
04230000 T  0003:0303:2
04231000 T  0003:0303:2
04232000 T  0003:0303:2
START OF SEGMENT *********   33
04233000 T  0033:0000:0
04234000 T  0033:0000:0
04235000 T  0033:0000:0
04236000 T  0033:0000:0
04237000 T  0033:0000:0
04238000 T  0033:0000:0
04239000 T  0033:0000:1
04240000 T  0033:0005:2
04241000 T  0033:0009:2
04242000 T  0033:0011:2
04243000 T  0033:0011:3
04244000 T  0033:0013:3
33 IS   16 LONG, NEXT SEG    3
04245000 T  0003:0303:2
04246000 T  0003:0303:2
04247000 T  0003:0303:2
04248000 T  0003:0303:2
04265000 T  0003:0311:2
04266000 T  0003:0311:2
04267000 T  0003:0311:2
04268000 T  0003:0311:2
04269000 T  0003:0311:2
04270000 T  0003:0311:2
04271000 T  0003:0311:2
04272000 T  0003:0311:2
04273000 T  0003:0312:0
04274000 T  0003:0313:2
04275000 T  0003:0313:3
04276000 T  0003:0314:0
04277000 T  0003:0314:0
04277500 T  0003:0314:0
04278000 T  0003:0314:0
START OF SEGMENT *********   34
04278500 T  0034:0000:0
```

```
      WRITE(LINE,BUG,B2D(L),
            IF STREAMTOG THEN
              SEARCH(COP,S.[42:6])
          ELSE IF T1:=S.[46:2]=1 THEN SEARCH(WOP,S.[36:10])
          ELSE WOP[T1] , IF STREAMTOG THEN
         B2D(S.[36:6]) ELSE IF T1=1 THEN WOP[1]
  ELSE B2D(S.[36:10]),B2D(S))
ELSE WRITE(LINE,BUG,B2D(L),
            IF STREAMTOG THEN
              SEARCH(COP,S.[42:6])
          ELSE IF T1:=S.[46:2]=1 THEN SEARCH(WOP,S.[36:10])
          ELSE WOP[T1] , IF STREAMTOG THEN
         B2D(S.[36:6]) ELSE IF T1=1 THEN WOP[1]
  ELSE B2D(S.[36:10]),B2D(S));
   END DEBUG;

   COMMENT    EMIT PLACES SYLLABLES INTO EDOC, CALLS DEBUG FOR
              DEBUGGING OUTPUT ON THE PRINTER, AND CHECKS FOR SEGMENTS
              GREATER THAN 4093 SYLLABLES,          ;
PROCEDURE EMIT ( S ) ; VALUE S; REAL   S  ;
              BEGIN
              IF L < 4088 THEN
              BEGIN
              LINKTOG ← TRUE;
          PACK(CODE(L DIV 4+1),L.[46:2],S);
              IF DEBUGTOG THEN DEBUG(S);
              L←L+1;
              END  ELSE
              BEGIN ERR(200); L←1; END;
                    COMMENT   200 EMIT - SEGMENT GREATER THAN 4093 SYLLABLES          *;
              END  EMIT ;
                    COMMENT   EMITD EMITS THE DIA,DIB,TRB SEQUENCE OF CODE, THE
              PREVIOUS SETTING OF THE G-H AND K-V REGISTERS IS COMPARED
              THE CURRENT , IF THE G-H,K-V OR BOTH ARE ALREADY SET THEN
              THE APPROIATE SYLLABLE(S) ARE OMITTED
              IF 0 BITS ARE TO BE TRANSFERED THEN NO SYLLABLES ARE
              EMITTED     ;
          PROCEDURE   EMITD(A,B,T); VALUE A,B,T ; INTEGER A,B,T;
        BEGIN LABEL EXIT,NORMAL;

              REAL Q;
              IF T = 15 THEN
              BEGIN
              IF A = 33 THEN Q ← 512
              ELSE IF A ≠ 18 THEN GO TO NORMAL;
              IF B = 18 THEN Q ← Q+256
              ELSE IF B ≠ 33 THEN GO TO NORMAL;
              EMITO(Q+197);  COMMENT -- THIS GETS OUT FIXED FIELD;
              GO TO EXIT;
              END;
          NORMAL:
              IF  T≠ 0 THEN
              BEGIN

               EMIT(((DIALA+A) DIV 6)×512 + ( A← A MOD 6)× 64 + DIA);

               EMIT(((DIALB+B) DIV 6)×512 + ( B← B MOD 6)× 64 + DIB);
```

```
04279000 T  0034:0000:1
04279500 T  0034:0008:0
04280000 T  0034:0009:2
04280500 T  0034:0010:1
04281000 T  0034:0015:2
04281500 T  0034:0019:2
04282000 T  0034:0023:2
04282500 T  0034:0027:2
04283000 T  0034:0038:0
04283500 T  0034:0039:2
04284000 T  0034:0040:1
04284500 T  0034:0045:2
04285000 T  0034:0049:2
04285500 T  0034:0053:2
04286000 T  0034:0061:2
34 IS   64 LONG, NEXT SEG    3
04288000 T  0003:0314:0
04289000 T  0003:0314:0
04290000 T  0003:0314:0
04291000 T  0003:0314:0
04292000 T  0003:0314:0
04293000 T  0003:0314:0
04294000 T  0003:0315:3
04295000 T  0003:0316:0
04296000 T  0003:0317:2
04297000 T  0003:0322:1
04298000 T  0003:0324:1
04299000 T  0003:0325:3
04300000 T  0003:0325:3
04301000 T  0003:0329:3
04302000 T  0003:0329:3
04305000 T  0003:0329:3
04306000 T  0003:0329:3
04307000 T  0003:0329:3
04308000 T  0003:0329:3
04309000 T  0003:0329:3
04310000 T  0003:0329:3
04311000 T  0003:0329:3
04311010 T  0003:0329:3
START OF SEGMENT **********   35
04311020 T  0035:0000:0
04311030 T  0035:0000:0
04311040 T  0035:0000:1
04311050 T  0035:0001:2
04311060 T  0035:0002:1
04311070 T  0035:0005:2
04311080 T  0035:0006:1
04311090 T  0035:0009:2
04311100 T  0035:0010:1
04311110 T  0035:0011:2
04311120 T  0035:0011:2
04312000 T  0035:0011:2
04313000 T  0035:0011:3
04314000 T  0035:0012:0
04315000 T  0035:0012:0
04316000 T  0035:0017:2
04317000 T  0035:0017:2
```

```
                    EMIT(TRB+64×T);


                END       EMITD  ;
EXIT:           END;

PROCEDURE EMITT(E,A,B); VALUE E,A,B; REAL E,A,B;
        BEGIN LABEL EXIT,IS;

                INTEGER S,T1,T2;
                PROCEDURE EMIT21(E,B); VALUE E,B;
                     REAL E;
                     BOOLEAN B;
                     BEGIN IF E = 0 THEN
                               BEGIN IF B THEN EMITO(XCH); END
                          ELSE BEGIN GT1 ← E.ADDRESS;
                                    IF E ← E.CLASS ≤ INTID THEN
                                         EMITV(GT1)
                                    ELSE IF E ≤ INTARRAYID THEN
                                         EMITPAIR(GT1,LOD) ELSE
                                    EMITN(GT1)
                               END
                     END;
                IF B = 0 THEN
                     BEGIN EMIT21(E,FALSE); GO TO EXIT END;
                IF STACKCT ≠ 0 THEN GO TO IS;
                IF B = 15 THEN
                     BEGIN IF A = 33 THEN
                               BEGIN EMIT21(E,FALSE);
                                    EMIT(0); EMITO(INX);
                                    GO TO EXIT;
                               END;
                          IF A = 18 THEN
                               BEGIN EMIT(0);
                                    EMIT21(E,TRUE);
                                    EMITO(197);
                                    GO TO EXIT;
                               END;
                          GO TO IS;
                     END;
                IF B ≤ 10 AND A+B = 48 THEN
                     BEGIN EMIT21(E,FALSE);
                          EMITL(2*B-1);
                          EMITO(LND);
                          GO TO EXIT;
                     END;
IS:             IF (S ← (48-A-B) MOD 6)+B ≤ 39 THEN
                     BEGIN EMIT21(E,FALSE);
                          EMIT(T2←(T1←A DIV 6)×512+(A MOD 6)×64+DIA);
                          EMIT(((A+B-1) DIV 6 -T1+1)×512+64×S+37);
                          GO TO EXIT;
                     END;
                EMIT(0);
                EMIT21(E,TRUE);
                EMITD(A,48-B,B);
```

```
04318000 T  0035:0021:3
04319000 T  0035:0023:3
04320000 T  0035:0023:3
04321000 T  0035:0023:3
04322000 T  0035:0023:3
04322100 T  0035:0023:3
 35 IS   27 LONG, NEXT SEG    3
04500000 T  0003:0329:3
04501000 T  0003:0329:3
START OF SEGMENT **********   36
04502000 T  0036:0000:0
04503000 T  0036:0000:0
04504000 T  0036:0000:0
04505000 T  0036:0000:0
04506000 T  0036:0000:0
04507000 T  0036:0000:1
04508000 T  0036:0002:1
04509000 T  0036:0004:1
04521000 T  0036:0006:0
04522000 T  0036:0007:2
04523000 T  0036:0008:1
04524000 T  0036:0010:0
04525000 T  0036:0011:2
04526000 T  0036:0011:3
04526100 T  0036:0011:3
04526200 T  0036:0012:1
04527000 T  0036:0014:1
04528000 T  0036:0016:0
04529000 T  0036:0016:1
04530000 T  0036:0018:0
04531000 T  0036:0019:3
04532000 T  0036:0021:2
04533000 T  0036:0021:3
04534000 T  0036:0021:3
04535000 T  0036:0022:0
04536000 T  0036:0023:3
04537000 T  0036:0024:1
04538000 T  0036:0025:2
04539000 T  0036:0025:3
04540000 T  0036:0025:3
04541000 T  0036:0026:0
04542000 T  0036:0026:0
04543000 T  0036:0028:1
04544000 T  0036:0030:0
04545000 T  0036:0033:2
04546000 T  0036:0034:0
04547000 T  0036:0034:1
04548000 T  0036:0034:1
04549000 T  0036:0038:0
04550000 T  0036:0039:3
04551000 T  0036:0044:1
04552000 T  0036:0049:3
04553000 T  0036:0050:0
04554000 T  0036:0050:0
04555000 T  0036:0051:2
04556000 T  0036:0052:0
04557000 T  0036:0053:3
```

```
   EXIT: END;

COMMENT THIS SECTION CONTAINS MISCELLANEOUS SERVICE ROUTINES;
COMMENT STEPI AND STEPIT ARE SHORT CALLS ON TABLE;
PROCEDURE STEPIT; ELCLASS ← TABLE(I←I+1);
INTEGER PROCEDURE STEPI; STEPI←ELCLASS←TABLE(I←I+1);
COMMENT TAKE FETCHS A WORD FROM INFO;
REAL PROCEDURE TAKE(INDEX); VALUE INDEX; INTEGER INDEX;
            TAKE ← INFO[INDEX.LINKR,INDEX.LINKC];
COMMENT PUT PLACES A WORD INTO INFO;
PROCEDURE PUT(WORD,INDEX); VALUE WORD,INDEX; REAL WORD,INDEX;
            INFO[INDEX.LINKR,INDEX.LINKC] ← WORD;
COMMENT FLAG FLAGS ERROR MESSAGES, COUNTS THEM AND SUPPRESS FUTURE
            ERROR MESSAGES UNTIL THE COMPILER THINKS IT HAS RECOVERED;
PROCEDURE FLAG(ERRNUM); VALUE ERRNUM; INTEGER ERRNUM;
            BEGIN
   COMMENT WRITERROR IS THE STREAM PROCEDURE WHICH ACTUALLY PRODUCES
            THE ERROR MESSAGE ON THE PRINTER;
     STREAM PROCEDURE WRITERROR(ERRNUM,ACCUM,LINE,COUNT,LSTSEQ);

            VALUE ERRNUM,COUNT;
            BEGIN
                 DI ← LINE; 44(DS←2LIT" "); COMMENT CLEAR BUFFER;
                 SI ←LSTSEQ; SI ← SI-8; DS ←WDS;
                 SI ← LINE; DS ← 2 WDS;
                 4(DS ← 2 LIT "XX"); COMMENT  SET RIGHT MARGIN FLAG;
                 SI ← LSTSEQ; DI ← LSTSEQ; DI ← DI-8; DS ← WDS;
                 DI ← LINE; DI ← DI+8; COMMENT INDENT MESSAGE;
                 DS ← 13 LIT "ERROR NUMBER ";
                 SI ← LOC ERRNUM; DS ← 3 DEC; COMMENT CONVERT ERRNUM;
                 DS ← 4 LIT " -- ";
                 SI ← ACCUM; SI ← SI+3; DS ← COUNT CHR;
                    COMMENT PLACE ALPHA IN BUFFER;
                 DS ← LIT ","
            END WRITERROR;
            IF ERRORTOG THEN % DO NOTHING IF WE SUPPRESS MSSGS,
                 BEGIN
                 SPECTOG := FALSE;
                 ERRORCOUNT := ERRORCOUNT+1; COMMENT COUNT ERRORS;
                 IF NOT LISTER THEN
                      BEGIN
                      EDITLINE(LIN,FCR,L DIV 4,L,[46:2],MEDIUM,0);
                     MOVE(1,INFO[LASTSEQROW,LASTSEQUENCE],LIN[12]);
                     IF NOHEADING THEN DATIME; WRITELINE;
                     END;
    COMMENT    PRINT CARDIMAGE IF WE ARE NOT LISTING;
                 ACCUM[1] ← Q;    COMMENT RESTORE ACCUMULATOR;
                 WRITERROR(ERRNUM,ACCUM[1],LIN[*],Q,[12:6],
                      INFO[LASTSEQROW,LASTSEQUENCE]);
          IF NOT NOHEADING THEN WRITELINE ;
                 ERRORTOG ← FALSE; COMMENT INHIBIT MESSAGES;
IF PUNCHTOG THEN
     BEGIN REAL T1,T2,T3,T4; LABEL L,L1,EXIT;

STREAM PROCEDURE P1(L,P); VALUE P;
      BEGIN DI ← L; 15(DS←8LIT" ");
SI ← LOC P; DI←L; SI←SI+5; SKIP 3 SR;
```

```
                                        04558000 T  0036:0053:3
                                  36 IS   57 LONG, NEXT SEG    3
                                        05000000 T  0003:0329:3
                                        05001000 T  0003:0329:3
                                        05002000 T  0003:0329:3
                                        05003000 T  0003:0332:1
                                        05004000 T  0003:0338:0
                                        05005000 T  0003:0338:0
                                        05006000 T  0003:0338:0
                                        05007000 T  0003:0344:0
                                        05008000 T  0003:0344:0
                                        05009000 T  0003:0344:0
                                        05010000 T  0003:0348:0
                                        05011000 T  0003:0348:0
                                        05012000 T  0003:0348:0
                                        05013000 T  0003:0348:0
                                        05014000 T  0003:0348:0
                                        05015000 T  0003:0348:0
                                        05016000 T  0003:0348:0
                                  START OF SEGMENT **********   37
                                        05017000 T  0037:0000:0
                                        05018000 T  0037:0000:0
                                        05019000 T  0037:0000:0
                                        05020000 T  0037:0001:2
                                        05021000 T  0037:0002:0
                                        05022000 T  0037:0002:1
                                        05023000 T  0037:0003:3
                                        05024000 T  0037:0004:1
                                        05025000 T  0037:0005:2
                                        05026000 T  0037:0007:2
                                        05027000 T  0037:0007:3
                                        05028000 T  0037:0008:0
                                        05029000 T  0037:0009:2
                                        05030000 T  0037:0009:2
                                        05031000 T  0037:0009:3
                                        05032000 T  0037:0010:0
                                        05033000 T  0037:0010:0
                                        05034000 T  0037:0010:1
                                        05035000 T  0037:0011:3
                                        05036000 T  0037:0012:1
                                        05037000 T  0037:0013:2
                                        05038000 T  0037:0013:3
                                        05039500 T  0037:0017:2
                                        05039600 T  0037:0019:3
                                        05041000 T  0037:0031:2
                                        05042000 T  0037:0031:2
                                        05043000 T  0037:0031:2
                                        05044000 T  0037:0032:0
                                        05045000 T  0037:0034:1
                                        05046000 T  0037:0036:0
                                        05047000 T  0037:0047:2
                                        05048000 T  0037:0048:0
                                        05049000 T  0037:0048:1
                                  START OF SEGMENT **********   38
                                        05050000 T  0038:0000:0
                                        05051000 T  0038:0000:0
                                        05052000 T  0038:0002:0
```

```
5
(DS←3 RESET;3(IF SB THEN DS←SET ELSE DS←RESET;SKIP 1 SB));
SI←P;
DI←DI+2; 2(8
(DS←3 RESET;3(IF SB THEN DS←SET ELSE DS←RESET;SKIP 1 SB));
DS←LIT" ") END;
STREAM PROCEDURE P2(L,P); VALUE P;
    BEGIN DI←L; DI ←DI+26;DS←LIT"0";
       SI←P; SI←SI-2; SKIP 1SB;
3
(DS←3 RESET;3(IF SB THEN DS←SET ELSE DS←RESET;SKIP 1 SB));
END;
REAL STREAM PROCEDURE ABS(A); VALUE A;
     BEGIN DI ← LOC ABS; SI ←A; DS← WDS;DI←DI-8; DS←RESET END;
STREAM PROCEDURE  BITEDUST(X,N,ID,LINE,COUNT); VALUE ID,N,COUNT;
 BEGIN  LOCAL  T,F,H;
         DI←LOC F; SI←LINE; DS←WDS;DI←F;
SI←LOC ID;SI←SI+2;DS← 6CHR;  57(DS←2 LIT"  ");
              SI ← LOC COUNT ;SI←SI+8 ;
              DI ←LOC H ; DS ←WDS ;
              DI ← LOC LINE ; SI ← H;
       SI←LOC X;SKIP 2 SB;
      IF SB THEN
        BEGIN   SI←X;  T←SI;
           NC
         DI←LOC F; SI←LINE; DS←WDS;DI←F;
            SI← LOC COUNT; SI←SI+6;
4
(DS←3 RESET;3(IF SB THEN DS←SET ELSE DS←RESET;SKIP 1 SB));
                DS←2 LIT" "; SI←COUNT; SI←SI+48;COUNT←SI;
SI←T;
        6(2(8
(DS←3 RESET;3(IF SB THEN DS←SET ELSE DS←RESET;SKIP 1 SB));
DS← LIT" "); DS←LIT" " );
              DI ← LOC LINE ; SI ← H;
SI←T;SI←SI+48;T←SI);END END;
           BITEDUST(ERRORCOUNT, 63 ,"PRT    ",LINE, 21);
FOR  T1← 0 STEP 1 UNTIL 31DO BITEDUST(INFO[ T1,*],43,"INFO  ",LINE,0);
BITEDUST(      ELBAT[*],13,"ELBAT ",LINE,0);
      BITEDUST(STACKHEAD[*],21,"STHEAD",LINE,0);
         T1←MKABS(ERRNUM)-1; T3←T1;
L:    T2←ABS(T3);
      IF T2.[33:15]=0 THEN BEGIN T3←T2.[18:15];
         IF T3=0 THEN GO TO EXIT ELSE GO TO L END ;
  T4 ← IF T2.[33:15] < 512 THEN 0 ELSE T2.[33:15]&T2[30:10:2];
L1:P1(LIN[0],T1);IF T1=T3 THEN BEGIN IF T≠0 THEN P2(LIN[0],T4);
   T1←T1-1;
    WRITELINE;
                        T3←T2.[18:15];GO TO L  END;
    T1:=T1-1; WRITELINE;
    GO TO L1;
EXIT: END ;

                    END END FLAG;

                    LABEL ENDOFITALL;
COMMENT ERR,IS THE SAME AS FLAG EXCEPT THAT IT MAKES AN ATTEMPT TO
```

```
05053000 T  0038:0003:2
05054000 T  0038:0003:2
05055000 T  0038:0005:3
05056000 T  0038:0006:0
05057000 T  0038:0006:1
05058000 T  0038:0009:2
05059000 T  0038:0010:0
05060000 T  0038:0010:0
05061000 T  0038:0012:0
05062000 T  0038:0012:1
05063000 T  0038:0012:1
05064000 T  0038:0015:3
05065000 T  0038:0015:3
05066000 T  0038:0015:3
05067000 T  0038:0018:0
05068000 T  0038:0018:0
05069000 T  0038:0019:2
05070000 T  0038:0020:0
05071000 T  0038:0021:3
05071100 T  0038:0022:0
05071200 T  0038:0022:1
05072000 T  0038:0023:2
05073000 T  0038:0023:3
05074000 T  0038:0024:0
05075000 T  0038:0025:2
05076000 T  0038:0026:0
05077000 T  0038:0027:2
05078000 T  0038:0027:3
05079000 T  0038:0027:3
05080000 T  0038:0030:1
05081000 T  0038:0031:3
05082000 T  0038:0032:0
05083000 T  0038:0032:1
05084000 T  0038:0035:2
05085000 T  0038:0036:1
05086000 T  0038:0037:2
05087000 T  0038:0038:1
05088000 T  0038:0041:3
05089000 T  0038:0049:3
05090000 T  0038:0052:1
05091000 T  0038:0055:3
05092000 T  0038:0058:1
05093000 T  0038:0060:1
05094000 T  0038:0063:3
05095000 T  0038:0065:3
05096000 T  0038:0070:0
05097000 T  0038:0076:0
05097500 T  0038:0077:2
05098000 T  0038:0091:2
05099000 T  0038:0093:2
05099100 T  0038:0104:1
05100000 T  0038:0105:2
38 IS  106 LONG, NEXT SEG   37
05101000 T  0037:0050:0
37 IS   53 LONG, NEXT SEG    3
05101100 T  0003:0348:0
05102000 T  0003:0348:0
```

```
                    RECOVER FROM ERROR SITUATIONS BY SEARCHING FOR A                    05103000 T   0003:0348:0
                    SEMICOLON, END, OR BEGIN;                                           05104000 T   0003:0348:0
PROCEDURE ERR(ERRNUM); VALUE ERRNUM; INTEGER ERRNUM;                                    05105000 T   0003:0348:0
                    BEGIN FLAG(ERRNUM);                                                 05106000 T   0003:0348:0
                         I ← I-1;                                                       05107000 T   0003:0349:3
                         IF ERRNUM = 200 THEN I := I/0; % SEGMENT TOO LARGE.            05107100 P   0003:0351:2
                         IF ERRNUM = 611 THEN I := I/0; % ERRMAX EXCEEDED.              05107200 P   0003:0353:3
                         DO IF STEPI = BEGINV THEN STMT UNTIL                           05108000 T   0003:0356:0
                            ELCLASS = ENDV OR ELCLASS = SEMICOLON END ERR;              05109000 T   0003:0358:0
DEFINE ERROR = ERR#;  COMMENT ERROR IS A SYNONM FOR ERR;                                05110000 T   0003:0360:1
COMMENT CHECKER IS A SMALL PROCEDURE THAT CHECKS TO SEE THAT THE                        05111000 T   0003:0360:1
                    UPLEVEL ADDRESSING CONVENTIONS ARE OBEYED;                          05112000 T   0003:0360:1
PROCEDURE CHECKER(ELBATWORD); VALUE ELBATWORD; REAL ELBATWORD;                          05113000 T   0003:0360:1
                    BEGIN                                                               05114000 T   0003:0360:1
                         IF MODE ≥ 2 THEN                                               05115000 T   0003:0360:1
                         IF GT1 ← ELBATWORD.LVL ≥ FRSTLEVEL THEN                        05116000 T   0003:0361:3
                         IF GT1 < SUBLEVEL THEN                                         05117000 T   0003:0364:0
                    IF ELBATWORD.[9:2] ≠ 1                                              05118000 T   0003:0365:2
                         THEN BEGIN FLAG(101); ERRORTOG ← TRUE END                      05119000 T   0003:0366:1
                    END CHECKER;                                                        05120000 T   0003:0369:2
   COMMENT GIT IS USED TO OBTAIN THE INDEX TO ADDITIONAL INFORMATION                    05121000 T   0003:0369:2
                    GIVEN THE LINK TO THE ELBAT WORD;                                   05122000 T   0003:0369:2
INTEGER PROCEDURE GIT(L); VALUE L; REAL L;                                              05123000 T   0003:0369:2
                    GIT ← TAKE(L).INCR+L.LINK;                                          05124000 T   0003:0369:2
COMMENT GNAT IS USED TO OBTAIN THE PRT ADDRESS OF A GIVEN DESCRIPTOR.                   05125000 T   0003:0375:2
                    IF THE ADDRESS HAS NOT BEEN ASSIGNED, THEN IT USES                  05126000 T   0003:0375:2
                    GETSPACE TO OBTAIN THE PRT ADDRESS;                                 05127000 T   0003:0375:2
INTEGER PROCEDURE GNAT(L); VALUE L; REAL L;                                             05128000 T   0003:0375:2
                    BEGIN                                                               05129000 T   0003:0375:2
                    REAL A;                                                             05130000 T   0003:0375:2
                                                                          START OF SEGMENT **********   39
                    IF GNAT ←(A←TAKE(L)).ADDRESS=0                                      05131000 T   0039:0000:0
                    THEN PUT(A&(GNAT:=GETSPACE(TRUE,L,LINK+1))[16:37:11],L)             05132000 T   0039:0002:0
                    END GNAT;                                                           05133000 T   0039:0007:2
                                                                            39 IS   10 LONG, NEXT SEG    3
REAL PROCEDURE TAKEFRST;                                                                05188000 T   0003:0375:2
                    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
                    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
                    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
                    BEGIN INTEGER ADRES;                                                05215000 T   0003:0392:0
```

```
                    CHECKER(ELBAT[I]);
                    ADRES ← ELBAT[I].ADDRESS;
                    IF BOOLEAN(ELBAT[I].FORMAL)
                       THEN BEGIN EMITV(ADRES); ADRES ← ADRES-1 END
                       ELSE BEGIN
                         IF TABLE(I) = SUPERFRMTID
                           THEN EMITL(TAKEFRST) ELSE EMITL(ELBAT[I].INCR)
                       END;
                    IF TABLE(I) = SUPERFRMTID
                       THEN BEGIN  BANA; I ← I-1;
                         EMITO(SSP); EMITO(ADD); EMITV(ADRES) END;
                    EMITPAIR(ADRES,LOD) END PASSFORMAT;

COMMENT  STREAMWORDS EITHER RESERVES OR UNRESERVES STREAM RESERVED
         WORDS - IT COMPLEMENTS THEIR STATE;
PROCEDURE STREAMWORDS;
         BEGIN GT1 ← 0;
          DO BEGIN
           INFO[1,GT1].LINK←STACKHEAD[GT2←(T←INFO[1,GT1]).ADDRESS];
           STACKHEAD[GT2] ← T.LINK;
           GT1 ← GT1+2;
          END UNTIL BOOLEAN(T.FORMAL) END STREAMWORDS;


PROCEDURE PROGDESCBLDR(PRTADR,SAV,SIZE,TYPE);
                    VALUE PRTADR,SAV,SIZE,TYPE;
                    INTEGER PRTADR,SIZE,TYPE; BOOLEAN SAV;
              BEGIN PRTADR←PRTADR.[38:10];
              IF SAV THEN BEGIN PRT[PRTADR] ← ( IF TYPE = LDES
                   THEN SIZE ELSE CORADR)
                    &SIZE[8:38:10]&TYPE[1:43:5]&3[6:46:2];
                   IF TYPE≠LDES THEN CORADR←CORADR+SIZE;
              END
         ELSE BEGIN PRT[PRTADR]←0&DISKADR[18:33:15]&SIZE[8:38:10]
                              &TYPE[1:43:5]&1[6:46:2];
                   DISKADR←(SIZE+29) DIV 30+DISKADR;
              END;
              END PROGDESCRLDR;
COMMENT  DOTSYNTAX ANALYSES THE SYNTAX OF A PARTIAL WORD DESIGNATOR.
              IT REPORTS IF AN ERROR IS FOUND.  IT RETURNS WITH THE
              LITERALS INVOLVED;
BOOLEAN PROCEDURE DOTSYNTAX(FIRST,SECOND);
              INTEGER FIRST,SECOND;
              BEGIN
                LABEL EXIT;

                 IF STEPI = FIELDID THEN % GET INFO FROM INFO      %112-
                   BEGIN                                           %112-
                     FIRST := ELBAT[I].SBITF;                      %112-
                     SECOND := ELBAT[I].NBITF;                     %112-
```

```
                                                        START OF SEGMENT **********   40
                                                             05216000 T  0040:0000:0
                                                             05217000 T  0040:0001:2
                                                             05218000 T  0040:0002:1
                                                             05219000 T  0040:0002:1
                                                             05220000 T  0040:0006:0
                                                             05221000 T  0040:0006:1
                                                             05222000 T  0040:0007:2
                                                             05223000 T  0040:0010:0
                                                             05224000 T  0040:0011:2
                                                             05225000 T  0040:0012:0
                                                             05226000 T  0040:0014:1
                                                             05227000 T  0040:0017:2
                                                          40 IS   21 LONG, NEXT SEG    3
                                                             05228000 T  0003:0392:0
                                                             05229000 T  0003:0392:0
                                                             05230000 T  0003:0392:0
                                                             05231000 T  0003:0392:0
                                                             05232000 T  0003:0393:3
                                                             05233000 T  0003:0394:0
                                                             05234000 T  0003:0398:1
                                                             05235000 T  0003:0401:3
                                                             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
                                                             05245000 T  0003:0404:1
                                                             05246000 T  0003:0404:1
                                                             05247000 T  0003:0404:1
                                                             05247500 T  0003:0404:1
                                                             05248000 T  0003:0406:0
                                                             05248500 T  0003:0408:0
                                                             05249000 T  0003:0410:0
                                                             05250000 T  0003:0413:3
                                                             05251000 T  0003:0416:0
                                                             05252000 T  0003:0416:0
                                                             05253000 T  0003:0419:3
                                                             05254000 T  0003:0422:0
                                                             05254500 T  0003:0424:0
                                                             05255000 T  0003:0424:0
                                                             05267000 T  0003:0424:1
                                                             05268000 T  0003:0424:1
                                                             05269000 T  0003:0424:1
                                                             05270000 T  0003:0424:1
                                                             05271000 T  0003:0424:1
                                                             05272000 T  0003:0424:1
                                                             05273000 T  0003:0424:1
                                                        START OF SEGMENT **********   41
                                                             05273100 C  0041:0000:0
                                                             05273200 C  0041:0001:2
                                                             05273300 C  0041:0001:3
                                                             05273400 C  0041:0003:2
```

```
                    GO TO EXIT;                                                  %112-
                  END                                                            %112-
                ELSE                                                             %112-
                  IF ELCLASS = LFTBRKET THEN                                     %112-
                    IF STEPI = FIELDID THEN                                      %112-
                      BEGIN                                                      %112-
                        FIRST := ELBAT[I].SBITF;                                 %112-
                        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;
```

```
BOOLEAN PROCEDURE RANGE(LOWER,UPPER);
                  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;
 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;

           BEGIN DI ← LOC GETSYL; DI ← DI+6;
                 SI ← W; SI ← SI+S; SI ← SI+S; DS ← 2 CHR END;
       GET←GETSYL(CODEC     L DIV 4+1),L.[46:2]); END  GET;

 COMMENT  CALL SWITCH PERFORMS THE FINAL MESS OF GETTING A PROPER DE-
          SCRIPTOR TO THE TOP OF THE STACK;
 PROCEDURE CALLSWITCH(H); VALUE H; REAL H;
       BEGIN EMITV(GNAT(H)); EMITO(PRTE); EMITO(LOD) END CALLSWITCH;
```

```
                05273500 C  0041:0005:2
                05273600 C  0041:0005:3
                05273700 C  0041:0005:3
                05273800 C  0041:0005:3
                05273900 C  0041:0006:1
                05274000 P  0041:0008:0
                05274100 C  0041:0008:1
                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
                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
                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
                05312000 T  0042:0003:2
          42 IS   12 LONG, NEXT SEG    3
                05313000 T  0003:0429:2
                05314000 T  0003:0429:2
                05315000 T  0003:0429:2
                05316000 T  0003:0429:2
```

```
PROCEDURE WRITEPRT(PORS,N,GS); VALUE PORS,N,GS; INTEGER PORS,N,GS;
    BEGIN
    LABEL EXIT;

    STREAM PROCEDURE FILLIT(LIN,PORS,CELL,N,ID);
                            VALUE PORS,CELL,N;
         BEGIN
         LOCAL COUNT;
         LABEL M0,M1,M2,M3,M4,M5,M6,M7,XIT;
         SI:=LOC PORS; SI:=SI+3; DI:=LIN; % "PRT" OR "STACK",
         IF SC="P" THEN
              BEGIN  DS:=3 CHR; DS:=LIT"(";  END
         ELSE BEGIN
              DS:=5 CHR; DS:=LIT"("; SI:=LOC CELL; SI:=SI+5;
              IF SC≥"6" THEN DS:=2 LIT"F-" ELSE DS:=2 LIT"F+";
              COUNT:=DI; DI:=LOC CELL; DI:=DI+4;
              DS:=11 RESET; DI:=COUNT;
              END;
         SI:=LOC CELL; SI:=SI+4; TALLY:=4; % LOCATION.
         3(IF SC="0" THEN % DONT PRINT LEADING ZEROES.
              BEGIN SI:=SI+1; TALLY:=TALLY+63  END ELSE JUMP OUT);
         COUNT:=TALLY; DS:=COUNT CHR; TALLY:=0; COUNT:=TALLY;
         DS:=4 LIT") = "; CELL:=DI; % SAVE OUR PLACE.
         CI:=CI+N;
         GO M0;
         GO M1;
         GO M2;
         GO M3;
         GO M4;
         GO M5;
         GO M6;
         GO M7;
    M0:  SI:=ID; SI:=SI+2; DI:=LOC COUNT;
         DI:=DI+7; DS:=CHR; DI:=CELL;  DS:=COUNT CHR;
         GO XIT;
    M1:  DI:=CELL; DS:=19 LIT"*TEMPORARY STORAGE*"; GO XIT;
    M2:  DI:=CELL;
           DS:=36 LIT"*LIST, LABEL, OR SEGMENT DESCRIPTOR*"; GO XIT;
    M3:  DI:=CELL; DS:=27 LIT"*CASE STATEMENT DESCRIPTOR*"; GO XIT;
    M4:  DI:=CELL; DS:=19 LIT"*FORMAT DESCRIPTOR*"; GO XIT;
    M5:  DI:=CELL; DS:=24 LIT"*OUTER BLOCK DESCRIPTOR*"; GO XIT;
    M6:  DI:=CELL; DS:=20 LIT"*SEGMENT DESCRIPTOR*"; GO XIT;
    M7:  DI:=CELL; DS:=18 LIT"*LABEL DESCRIPTOR*";
    XIT:
         END FILLIT;
    BLANKET(14,LIN);
    IF N=1 THEN FILLIT(LIN,PORS,GS,0,ACCUM[1])
ELSE IF N>1 THEN FILLIT(LIN,PORS,GS,0,INFO[N,LINKR,N,LINKC])
```

```
                                                  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
                                                  05325010 T  0003:0433:2
                                                  05325020 T  0003:0433:2
                                                  05325030 T  0003:0433:2
                                          START OF SEGMENT *********   43
                                                  05325040 T  0043:0000:0
                                                  05325050 T  0043:0000:0
                                                  05325060 T  0043:0000:0
                                                  05325070 T  0043:0000:0
                                                  05325080 T  0043:0000:0
                                                  05325090 T  0043:0000:0
                                                  05325100 T  0043:0000:1
                                                  05325110 T  0043:0001:2
                                                  05325120 T  0043:0002:1
                                                  05325130 T  0043:0002:1
                                                  05325140 T  0043:0004:0
                                                  05325150 T  0043:0005:3
                                                  05325160 T  0043:0006:1
                                                  05325170 T  0043:0007:2
                                                  05325180 T  0043:0007:2
                                                  05325190 T  0043:0007:3
                                                  05325200 T  0043:0008:1
                                                  05325210 T  0043:0011:2
                                                  05325220 T  0043:0012:0
                                                  05325230 T  0043:0013:2
                                                  05325240 T  0043:0013:3
                                                  05325250 T  0043:0014:0
                                                  05325260 T  0043:0014:0
                                                  05325270 T  0043:0014:1
                                                  05325280 T  0043:0014:1
                                                  05325290 T  0043:0015:2
                                                  05325300 T  0043:0015:2
                                                  05325310 T  0043:0015:3
                                                  05325320 T  0043:0015:3
                                                  05325330 T  0043:0016:1
                                                  05325340 T  0043:0018:0
                                                  05325350 T  0043:0018:0
                                                  05325360 T  0043:0022:0
                                                  05325370 T  0043:0023:2
                                                  05325380 T  0043:0028:0
                                                  05325390 T  0043:0033:2
                                                  05325400 T  0043:0037:2
                                                  05325410 T  0043:0041:3
                                                  05325420 T  0043:0045:2
                                                  05325430 T  0043:0048:1
                                                  05325440 T  0043:0048:1
                                                  05325450 T  0043:0049:2
                                                  05325460 T  0043:0051:2
                                                  05325470 T  0043:0055:3
```

```
ELSE FILLIT(LIN,PORS,GS,ABS(N),N);
    IF NOHEADING THEN DATIME; WRITELINE;
    END WRITEPRT;

 COMMENT  GETSPACE MAKES ASSIGNMENTS TO VARIABLES AND DESCRIPTORS IN
               THE STACK AND PRT.  PERMANENT TELLS WHETHER IT IS A
               PERMANENTLY ASSIGNED CELL (ALWAYS IN PRT) OR NOT.  NON
               PERMANENT CELLS ARE EITHER IN STACK OR PRT ACCORDING TO
               MODE.  CARE IS TAKEN TO REUSE NON PERMANENT PRT CELLS;
INTEGER PROCEDURE GETSPACE(PERMANENT,L); VALUE PERMANENT,L;
                                         BOOLEAN PERMANENT; INTEGER L;
        BEGIN LABEL L1,L2,EXIT;

        STREAM PROCEDURE DOIT(C,A,I,S); VALUE C,A;
        BEGIN LOCAL N;
             DI←S; DS←8 LIT" "; SI←S; DS←9 WDS;
             SI←I; SI←SI+2; DI←LOC N; DI←DI+7; DS←CHR;
             DI←S; SI←LOC C; 2(DS←4 DEC);
             SI←I; SI←SI+3; DS←N CHR;
        END;
          BOOLEAN M,Q;
          INTEGER ROW,COL,GS;
IF NOT(STREAMTOG AND (LEVEL>2))THEN
    IF STEPI=RELOP THEN
             BEGIN
                  IF STEPI>IDMAX
                    THEN
                       BEGIN
                            IF ELCLASS=ADOP
                                 THEN
                                      IF ELBAT[I].ADDRESS=SUBOP
                                           THEN GS←FZERO ELSE GS←512
                            ELSE
                       BEGIN GS←0;I←I-1 END;
                            IF STEPI≠LITNO THEN FLAG(51);
                            IF ELBAT[I].ADDRESS≥512 THEN GS←1024;
                            GS←GS+ELBAT[I].ADDRESS
                     END
                       ELSE
                            BEGIN
                                 GS←ELBAT[I].ADDRESS;
                                 IF GS=0 THEN FLAG(51);
                                 IF GS≥FZERO AND GS≤1023 THEN GS←-GS;
                                 IF STEPI≠ADOP THEN I←I-1 ELSE
                                      BEGIN
                                      STEPIT;
                                      GS←ELBAT[I].ADDRESS+
                                           (IF ELBAT[I-1].ADDRESS=SUBOP
                                                THEN -GS ELSE +GS);
                                      END;
                                 GS←ABS(GS);
                            END; Q←GS<512 OR GS>1023;
       GO TO EXIT
       END ELSE I←I-1;
IF MODE = 0 OR PERMANENT
    THEN BEGIN
     IF PRTIMAX > 1023 THEN FLAG(148);
```

```
                05325480 T   0043:0063:2
                05325490 T   0043:0067:3
                05325500 T   0043:0079:2
          43 IS    80 LONG, NEXT SEG    3
                05326000 T   0003:0433:2
                05327000 T   0003:0433:2
                05328000 T   0003:0433:2
                05329000 T   0003:0433:2
                05330000 T   0003:0433:2
                05331000 T   0003:0433:2
                05333000 T   0003:0433:2
                05334000 T   0003:0433:2
       START OF SEGMENT **********   44
                05334100 T   0044:0000:0
                05334200 T   0044:0000:0
                05334300 T   0044:0000:0
                05334400 T   0044:0002:0
                05334500 T   0044:0003:2
                05334600 T   0044:0004:1
                05334700 T   0044:0005:3
                05343000 T   0044:0005:3
                05344000 T   0044:0005:3
                05344400 T   0044:0005:3
                05344500 T   0044:0007:3
                05344510 T   0044:0009:2
                05344520 T   0044:0009:3
                05344530 T   0044:0010:0
                05344540 T   0044:0010:1
                05344550 T   0044:0011:2
                05344560 T   0044:0011:2
                05344570 T   0044:0011:3
                0534458- T   0044:0013:2
                05344590 T   0044:0015:3
                05344600 T   0044:0016:0
                05344610 T   0044:0018:1
                05344615 T   0044:0021:2
                05344620 T   0044:0023:3
                05344630 T   0044:0024:0
                05344640 T   0044:0025:3
                05344650 T   0044:0025:3
                05344660 T   0044:0028:0
                05344661 T   0044:0029:3
                05344662 T   0044:0031:3
                05344670 T   0044:0034:1
                05344680 T   0044:0037:3
                05344690 T   0044:0038:0
                05344700 T   0044:0038:1
                05344710 T   0044:0039:3
                05344720 T   0044:0041:2
                05344730 T   0044:0044:0
                05344740 T   0044:0044:0
                05344750 T   0044:0045:2
                05344760 T   0044:0047:2
                05344770 T   0044:0047:3
                05345000 T   0044:0049:3
                05346000 T   0044:0050:0
                05347000 T   0044:0051:2
```

```
                    IF ASTOG THEN FLAG(505);                                                    05348000 T  0044:0053:2
                    PRTI ←                                                                      05349000 T  0044:0054:1
                    PRTIMAX←(GS←PRT,MAX)+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
      BEGIN REAL   J,K,T,S,M;                                                                   05401000 T  0003:0433:2
                                                                                        START OF SEGMENT **********   45
                    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
                                                                                                05411000 T  0045:0051:3
      END;                                                                                  45 IS   56 LONG, NEXT SEG    3
                                                                                                05411100 T  0003:0433:2
PROCEDURE NESTSORT(L,U); VALUE L,U; REAL L,U; FORWARD;                                          05412000 T  0003:0433:2
PROCEDURE SORTNEST;                                                                             05413000 T  0003:0433:2
      BEGIN ARRAY A[0:14];                                                                  START OF SEGMENT **********   46
                                                                                                05414000 T  0046:0001:3
          REAL I,J,K,T;                                                                         05414100 T  0046:0001:3
          REAL P,Q;                                                                             05415000 T  0046:0001:3
  STREAM PROCEDURE NESTFORM(I,N,L,A); VALUE I,N;                                                05416000 T  0046:0001:3
      BEGIN LOCAL S;                                                                            05417000 T  0046:0003:2
          DI←A; 15(DS←8 LIT " ");                                                               05418000 T  0046:0005:2
          DI←LOC S; DI←DI+7; SI←L; SI←SI+10; DS←CHR;                                            05419000 T  0046:0006:0
          DI←A; DI←DI+1; A←DI;                                                                  05420000 T  0046:0007:2
          DI←DI+6; DS← S CHR;                                                                   05421000 T  0046:0008:0
          DI←A; SI←LOC N; DS←4 DEC;
```

```
                D1←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;
            END;

PROCEDURE NESTSORT(L,U); VALUE L,U; REAL L,U;
            BEGIN REAL I,J,K,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;
BOTTOM:             SORTPRT[K].[18:15]←SORTPRT[I];
                    I←I+1;
                    IF K←K+1≤U THEN GO TO AGAIN ELSE GO TO EXIT;
EXIT:               FOR I←L STEP 1 UNTIL U DO
                    SORTPRT[I]←SORTPRT[I].[18:15];
                END;
            END;

COMMENT      ROUTINES IN THIS SECTION COMPILE CODE FOR ALL EXPRESSIONS;
COMMENT   AEXP IS THE ARITHMETIC EXPRESSION ROUTINE;
PROCEDURE AEXP;
                BEGIN
                    IF ELCLASS = IFV
                      THEN BEGIN IF IFEXP ≠ ATYPE THEN ERR(102) END
                      ELSE BEGIN ARITHSEC; SIMPARITH END
                END AEXP;
COMMENT ARITHSEC COMPILES FIRST PRIMARY IN AN ARITHMETIC EXPRESSION,
```

```
                05422000 T  0046:0008:1
                05423000 T  0046:0009:2
                05424000 T  0046:0009:3
                05425000 T  0046:0011:2
                05425100 T  0046:0012:1
                05425200 T  0046:0017:3
                05425300 T  0046:0019:3
                05425400 T  0046:0021:2
                05426000 T  0046:0023:3
                05427000 T  0046:0028:0
                05428000 T  0046:0032:0
                05429000 T  0046:0037:3
                05430000 T  0046:0039:2
                05430500 T  0046:0042:0
                05431000 T  0046:0043:3
                05432000 T  0046:0048:0
                05433000 T  0046:0052:1
                05434000 T  0046:0054:1
                05435000 T  0046:0058:1
                05436000 T  0046:0061:2
          46 IS   66 LONG, NEXT SEG    3
                05437000 T  0003:0433:2
                05438000 T  0003:0433:2
       START OF SEGMENT **********   47
                05439000 T  0047:0000:0
                05440000 T  0047:0000:0
                05441000 T  0047:0000:1
                05442000 T  0047:0003:2
                05443000 T  0047:0004:0
                05444000 T  0047:0005:3
                05445000 T  0047:0008:0
                05446000 T  0047:0009:2
                05447000 T  0047:0010:1
                05448000 T  0047:0013:2
                05449000 T  0047:0016:0
                05450000 T  0047:0019:2
                05451000 T  0047:0022:1
                05452000 T  0047:0023:2
                05453000 T  0047:0026:0
                05454000 T  0047:0027:3
                05455000 T  0047:0030:0
                05456000 T  0047:0033:2
                05457000 T  0047:0034:1
                05458000 T  0047:0037:2
                05459000 T  0047:0039:2
                05460000 T  0047:0043:3
                05461000 T  0047:0043:3
          47 IS   46 LONG, NEXT SEG    3
                06000000 T  0003:0433:2
                06001000 T  0003:0433:2
                06002000 T  0003:0433:2
                06003000 T  0003:0433:2
                06004000 T  0003:0433:2
                06005000 T  0003:0433:2
                06006000 T  0003:0436:1
                06007000 T  0003:0438:0
                06008000 T  0003:0438:0
```

```
                IN PARTICULAR IT HANDLES P, +P, -P, AND -P*Q WHERE P                    06009000 T  0003:0438:0
                AND Q ARE PRIMARIES;                                                    06010000 T  0003:0438:0
PROCEDURE ARITHSEC;                                                                     06011000 T  0003:0438:0
                BEGIN                                                                   06012000 T  0003:0438:0
                  IF ELCLASS = ADOP                                                     06013000 T  0003:0438:0
                     THEN BEGIN                                                         06014000 T  0003:0439:2
                       STEPIT;                                                          06015000 T  0003:0440:0
                       IF ELBAT[I-1].ADDRESS ≠ SUB THEN PRIMARY                         06016000 T  0003:0440:1
                         ELSE BEGIN                                                     06017000 T  0003:0443:2
                           PRIMARY;                                                     06018000 T  0003:0444:0
                           ENDTOG ← LINKTOG; EMITO(CHS);                                06021000 T  0003:0444:1
                           LINKTOG ← ENDTOG; ENDTOG ← FALSE END END                     06022000 T  0003:0446:0
                     ELSE PRIMARY END ARITHSEC;                                         06023000 T  0003:0447:3
COMMENT SIMPARITH COMPILES SIMPLE ARITHMETIC EXPRESSIONS ON THE                         06024000 T  0003:0449:2
                ASSUMPTION THAT AN ARITHMETIC PRIMARY HAS ALREADY BEEN                  06025000 T  0003:0449:2
                COMPILED.  IT ALSO HANDLES THE CASE OF A CONCATENATE                    06026000 T  0003:0449:2
                WHERE ACTUALPARAPART CAUSED THE VARIABLE ROUTINE TO                     06027000 T  0003:0449:2
                COMPILE ONLY PART OF A PRIMARY.  MOST OF THE WORK OF                    06028000 T  0003:0449:2
                SIMPARITH IS DONE BY ARITHCOMP, AN ARTIFIAL ROUTINE                     06029000 T  0003:0449:2
                WHICH DOES THE HIERARCHY ANALYSIS USING RECURSION.                      06030000 T  0003:0449:2
                ARITHCOMP IS A SUBROUTINE ONLY TO GET THIS RECURSION;                   06031000 T  0003:0449:2
PROCEDURE SIMPARITH;                                                                    06032000 T  0003:0449:2
                BEGIN                                                                   06033000 T  0003:0449:2
                  WHILE ELCLASS = AMPERSAND                                             06034000 T  0003:0449:2
                     DO BEGIN STEPIT; PRIMARY; PARSE END;                               06035000 T  0003:0449:2
                  WHILE ELCLASS ≥ EQVOP DO ARITHCOMP END;                               06036000 T  0003:0452:0
COMMENT ARITHCOMP IS THE GUTS OF THE ARITHMETIC EXPRESSION ROUTINE                      06037000 T  0003:0455:3
                ANALYSIS.  IT CALLS PRIMARY AT APPROPRIATE TIMES AND                    06038000 T  0003:0455:3
                EMITS THE ARITHMETIC OPERATORS.  THE HIERARCHY ANALYSIS                 06039000 T  0003:0455:3
                IS OBTAINED BY RECURSION;                                               06040000 T  0003:0455:3
PROCEDURE ARITHCOMP;                                                                    06041000 T  0003:0455:3
                BEGIN INTEGER OPERATOR, OPCLASS;                                        06042000 T  0003:0455:3
START OF SEGMENT **********   48
                 DO BEGIN                                                               06043000 T  0048:0000:0
                  OPERATOR ← 1 & ELBAT[I] [36:17:10];                                   06044000 T  0048:0000:0
   COMMENT THIS SETS UP THE OPERATOR WHICH WILL BE EMITTED.  THE HIGH                   06045000 T  0048:0002:0
                ORDER TEN BITS OF THE OPERATOR ARE LOCATED IN [17:10]                   06046000 T  0048:0002:0
                OF THE ELBAT WORD;                                                      06047000 T  0048:0002:0
                  OPCLASS ← ELCLASS;                                                    06048000 T  0048:0002:0
                  STEPIT; PRIMARY;                                                      06049000 T  0048:0002:1
                     BEGIN                                                              06051000 T  0048:0003:3
                       WHILE OPCLASS < ELCLASS DO ARITHCOMP;                            06052000 T  0048:0003:3
   COMMENT THE CLASSES ARE ARRANGED IN ORDER OF HIERARCHY;                              06053000 T  0048:0006:0
                       EMIT(OPERATOR);                                                  06054000 T  0048:0006:0
                       EMIT(0); L ← L-1;                                                06054100 T  0048:0007:2
                  STACKCT ← 1;                                                          06054150 T  0048:0009:2
                     END;                                                               06054200 T  0048:0009:3
                 END UNTIL OPCLASS ≠ ELCLASS END ARITHCOMP;                             06055000 T  0048:0009:3
  48 IS   14 LONG, NEXT SEG    3
  INTEGER PROCEDURE EXPRSS; BEGIN AEXP; EXPRSS ← ATYPE END;                             06057000 T  0003:0455:3
PROCEDURE POLISHER(EXPECT); VALUE EXPECT; REAL EXPECT;                                  06060000 T  0003:0459:2
        BEGIN LABEL EXIT;                                                               06061000 T  0003:0459:2
START OF SEGMENT **********   49
                LABEL EL;                                                               06061900 T  0049:0000:0
                REAL COUNT,T1, T2;                                                      06062000 T  0049:0000:0
                BOOLEAN S;                                                              06063000 T  0049:0000:0
                REAL SSS;   INTEGER Z;                                                  06063500 T  0049:0000:0
```

```
STREAM PROCEDURE WRITEOUT(C,N,L); VALUE C,N;                          06064000 T   0049:0000:0
     BEGIN DI ← L; DS ← 2 LIT "S=";                                   06065000 T   0049:0000:0
          SI ← LOC C; SI ← SI+7; DS ← CHR;                            06066000 T   0049:0000:1
          SI ← LOC N; DS ← DEC;                                       06067000 T   0049:0001:3
          58(DS←2LIT " ");                                            06067500 T   0049:0002:0
     END;                                                             06068000 T   0049:0003:2
SSS← STACKCTR;                                                        06068500 T   0049:0003:2
IF STEPI ≠ LEFTPAREN THEN GO TO EXIT;                                 06069000 T   0049:0004:1
DO BEGIN                                                              06070000 T   0049:0006:0
     IF STEPI ≥ OPERATORS THEN                                        06071000 T   0049:0007:2
          BEGIN T1 ← (T2 ← ELBAT[I]).ADDRESS;                         06072000 T   0049:0008:0
               S ← S OR COUNT = T2.[11:3] < 0;                        06074000 T   0049:0010:1
               COUNT ← T2.[14:2]+COUNT-2;                             06075000 T   0049:0013:2
               IF ELCLASS ≥ OPERATOR THEN                             06076000 T   0049:0015:3
               BEGIN IF T1 ≠ 0 THEN EMITO(T1)                         06077000 T   0049:0016:0
               ELSE BEGIN                                             06078000 T   0049:0018:0
                    T1 ← T2.LINK+2;                                   06079000 T   0049:0019:2
                    T2 ← T2.INCR+T1;                                  06080000 T   0049:0021:2
                    FOR T1 ← T1 STEP 1 UNTIL T2 DO                    06081000 T   0049:0022:1
                        EMIT(TAKE(T1));                               06082000 T   0049:0024:0
                    END;                                              06083000 T   0049:0027:3
                END ELSE BEGIN T2 ← ELCLASS;                          06084000 T   0049:0027:3
                    IF STEPI ≠ LITNO THEN                             06085000 T   0049:0028:1
                        BEGIN ERR(500); GO TO EXIT END;               06086000 T   0049:0029:3
                    IF T2 = BITOP THEN EMIT(T1&C                      06087000 T   0049:0031:3
                        [36:42:6]) ELSE                               06088000 T   0049:0033:2
                    IF T2 = HEXOP THEN EMIT(T1&                       06089000 T   0049:0034:1
                        (T2←C DIV 6)[36:45:3]&(C-T2×6)                06090000 T   0049:0036:1
                        [39:45:3]) ELSE                               06091000 T   0049:0039:3
                   IF T2 = ISOLATE THEN                               06092000 T   0049:0041:2
                       BEGIN T2 ← C;                                  06093000 T   0049:0042:0
                            IF STEPI ≠ LITNO                          06094000 T   0049:0043:3
                            THEN BEGIN ERR(500);                      06095000 T   0049:0044:0
                            GO TO EXIT END;                           06096000 T   0049:0045:3
                                                                      06097000 T   0049:0046:0
                                                                      06098000 T   0049:0046:0
                                                                      06099000 T   0049:0046:0
                                                                      06099100 T   0049:0046:0
                            EMIT(Z←((T2+C-1)DIV 6-C DIV               06099200 T   0049:0048:1
                          6+1)×512+(48-T2-C)MOD 6×64+                 06100000 T   0049:0052:1
                            37);                                      06101000 T   0049:0054:0
                       END END;                                       06102000 T   0049:0054:0
                STEPIT;                                               06103000 T   0049:0054:1
                S ← S OR COUNT < 0;                                   06104000 T   0049:0056:0
          END ELSE BEGIN                                              06104100 T   0049:0056:1
               IF ELCLASS = LABELID THEN                              06104200 T   0049:0057:3
                   BEGIN T1:=2;                                       06104300 T   0049:0058:1
FL:                     GT4 ← TAKE(T2←GIT(ELBAT[I]));                 06104400 T   0049:0061:3
                        PUT(L,T2);                                    06104500 T   0049:0062:1
                        IF GT4 = 0 THEN GT4 ← L;                      06104510 T   0049:0064:1
                        IF (GT4:=L-GT4)DIV 4 ≥ 128 THEN               06104520 T   0049:0066:1
                           BEGIN GT4:=0;FLAG(50);END;                 06104600 T   0049:0068:1
                        EMIT(GT4×4+T1);                               06104700 T   0049:0070:1
                        STEPIT;                                       06104800 T   0049:0071:2
                   END ELSE                                           06105000 T   0049:0071:2
               IF ELCLASS ≠ PERIOD THEN AEXP ELSE BEGIN               06106000 T   0049:0073:3
               T2←0;                                                  06106100 T   0049:0074:1
               IF STEPI=PERIOD THEN
```

```
                                BEGIN T2←1; STEPIT END;                             06106200 T  0049:0075:3
                                IF ELCLASS>IDMAX THEN                               06106300 T  0049:0077:2
                                     BEGIN ERR(500); GO TO EXIT END;                06107000 T  0049:0078:0
                                IF ELCLASS = LABELID THEN                           06107100 T  0049:0079:3
                                     BEGIN T1 ← 0; GO TO EL END;                    06107200 T  0049:0080:1
                                IF T1 ← ELBAT[I].ADDRESS = 0 THEN                   06108000 T  0049:0082:0
                                     BEGIN ERR(100); GO TO EXIT END;                06109000 T  0049:0084:0
                                 EMITL(T1);                                         06110000 T  0049:0086:0
                                IF T1>1023 THEN                                     06110100 T  0049:0086:1
                                IF T2=0THEN FLAG(500)                               06110200 T  0049:0087:3
                                ELSE EMITO(PRTE);                                   06110300 T  0049:0089:3
                                 STEPIT;                                            06111000 T  0049:0091:2
                                END; COUNT ← COUNT+1;                               06112000 T  0049:0091:3
                                END;                                                06113000 T  0049:0093:2
                 END UNTIL ELCLASS ≠ COMMA;                                         06114000 T  0049:0093:2
                 IF ELCLASS ≠ RTPAREN THEN                                          06115000 T  0049:0094:0
                      BEGIN ERR(104); GO TO EXIT END;                               06116000 T  0049:0095:2
                 STEPIT;                                                            06117000 T  0049:0096:1
                 IF FALSE THEN                                                      06118000 T  0049:0097:2
                      BEGIN COUNT ← COUNT-EXPECT;                                   06119000 T  0049:0097:3
                           WRITEOUT(IF COUNT < 0 THEN "-" ELSE                      06120000 T  0049:0099:2
                                IF COUNT = 0 THEN " " ELSE "+",                     06121000 T  0049:0101:3
                                ABS(COUNT),LIN[0]);                                 06122000 T  0049:0103:3
                           WRITELINE;                                               06123000 T  0049:0105:2
                      END;                                                          06124000 T  0049:0115:2
           EXIT: STACKCTR ← SSS; END;                                               06125000 T  0049:0115:2
                                                                        49 IS  120 LONG, NEXT SEG     3
     PROCEDURE PRIMARY;                                                             06126000 T  0003:0459:2
          BEGIN LABEL                                                               06127000 T  0003:0459:2
                                                                        START OF SEGMENT **********   50
                 L1, L2, L3, L4, L5, L6, L7, L8, L9, L10,                           06128000 T  0050:0000:0
                 L11, L12, L13, L14, L15, L16, L17, L18, L19, L20,                  06129000 T  0050:0000:0
                 L21, L22, L23, L24, L25, L26, L27, L28, L29, L30,                  06130000 T  0050:0000:0
                 L31, L32, L33, L34, L35, L36, L37, L38, L39;                       06131000 T  0050:0000:0
                 SWITCH S ←                                                         06132000 T  0050:0000:0
                 L1, L2, L3, L4, L5, L6, L7, L8, L9, L10,                           06133000 T  0050:0002:1
                 L11, L12, L13, L14, L15, L16, L17, L18, L19, L20,                  06134000 T  0050:0002:1
                 L21, L22, L23, L24, L25, L26, L27, L28, L29, L30,                  06135000 T  0050:0002:1
                 L31, L32, L33, L34, L35, L36, L37, L38, L39;                       06136000 T  0050:0002:1
                 LABEL EXIT,RP,LDOT,LAMPER;                                         06137000 T  0050:0022:1
                 GO TO S[ELCLASS];                                                  06138000 T  0050:0022:1
                 IF ELCLASS = LFTBRKET THEN                                         06139000 T  0050:0025:2
                      BEGIN STEPIT; VARIABLE(FL);                                   06140000 T  0050:0025:3
                           IF ELCLASS ≠ RTBRKET THEN                                06141000 T  0050:0027:3
                                BEGIN ERR(118); GO TO EXIT END;                     06142000 T  0050:0028:0
                           STEPIT;                                                  06143000 T  0050:0030:0
                           GO TO LDOT;                                              06144000 T  0050:0030:1
                      END;                                                          06145000 T  0050:0031:2
                 IF ELCLASS = NOTOP THEN                                            06146000 T  0050:0031:2
                      BEGIN STEPIT; PRIMARY;                                        06147000 T  0050:0031:3
                           EMITLNG; EMIT(0); L←L-1;                                 06148000 T  0050:0033:2
                           GO TO EXIT;                                              06149000 T  0050:0035:3
                      END;                                                          06150000 T  0050:0036:0
                 IF ELCLASS = UNKNOWNID THEN ERR(100);                              06151000 T  0050:0036:0
     L1:L2:L3:L4:L5:L6:L8:L9:L10:L12:L13:L16:L17:L20:L21:L24:L25:L28:L29:           06152000 T  0050:0038:0
     L32:                                                                           06153000 T  0050:0039:2
                 ERR(103); GO TO EXIT;                                              06154000 T  0050:0039:2
```

```
L7:
     SUBHAND(FALSE); GO TO LDOT;
L11:
     IMPFUN; STACKCT ← STACKCT-1; GO TO LDOT;
L14:L15:
     STRMPROCSTMT; GO TO LDOT;
L18:L19:
     PROCSTMT(FALSE); GO TO LDOT;
L22:L23:L26:L27:L30:L31:
     VARIABLE(FP); GO TO LAMPER;
L33:L35:
     EMIT(0&ELBAT[I] [36:17:10]); STEPIT; GO TO LAMPER;
L34:L36:
     EMITNUM(C); STEPIT; GO TO LAMPER;
L38:
     POLISHER(1); GO TO LDOT;
L39:
     STEPIT; PRIMARY; STACKCT ← STACKCT-1;
     EMITO(LOD); GO TO LDOT;
L37:
     STEPIT; AEXP;
     STACKCT ← STACKCT-1;
     IF ELCLASS ≠ RTPAREN THEN
          BEGIN ERR(104); GO TO EXIT END;
     STEPIT;
LDOT:DOT;
LAMPER:
     STACKCT ← STACKCT +1;
     WHILE ELCLASS = AMPERSAND DO
          BEGIN STEPIT; PRIMARY; PARSE END;
EXIT: END PRIMARY;

PROCEDURE IMPFUN;
     BEGIN REAL T1,T2;

          T1 ← (T2 ← ELBAT[I]).ADDRESS;
          PANA;
          IF T1 ≠ 0 THEN EMITO(T1)
               ELSE BEGIN
               T1 ← T2.LINK+T2.INCR+1;
               T2 ← T2.LINK+2;
               FOR T2 ← T2 STEP 1 UNTIL T1 DO EMIT(TAKE(T2));
               END;
     END;

PROCEDURE SUBHAND(FROM); VALUE FROM; BOOLEAN FROM;
     BEGIN LABEL EXIT;

          REAL T1;
          T1 ← TAKEFRST;
          IF ELCLASS ≠ SUBID AND FROM THEN
               BEGIN IF STEPI ≠ ASSIGNOP THEN
                    BEGIN FLAG(503); GO TO EXIT END;
                    STEPIT;
                    AEXP;
                    EMITO(XCH);
                    GO TO EXIT;
```

```
06155000 T  0050:0040:0
06156000 T  0050:0041:2
06157000 T  0050:0042:0
06158000 T  0050:0043:2
06159000 T  0050:0045:2
06160000 T  0050:0046:0
06161000 T  0050:0047:2
06162000 T  0050:0047:2
06163000 T  0050:0048:0
06164000 T  0050:0049:2
06165000 T  0050:0050:0
06166000 T  0050:0051:2
06167000 T  0050:0054:0
06168000 T  0050:0054:0
06169000 T  0050:0055:3
06170000 T  0050:0056:0
06171000 T  0050:0057:2
06172000 T  0050:0058:0
06172500 T  0050:0060:0
06173000 T  0050:0061:3
06174000 T  0050:0062:0
06174500 T  0050:0063:2
06175000 T  0050:0064:0
06176000 T  0050:0065:2
06177000 T  0050:0066:1
06178000 T  0050:0067:2
06179000 T  0050:0068:1
06179500 T  0050:0069:2
06180000 T  0050:0070:0
06181000 T  0050:0072:0
06182000 T  0050:0074:0
 50 IS   76 LONG, NEXT SEG    3
06183000 T  0003:0459:2
06184000 T  0003:0459:2
START OF SEGMENT **********   51
06185000 T  0051:0000:0
06186000 T  0051:0002:0
06187000 T  0051:0002:1
06188000 T  0051:0004:0
06189000 T  0051:0005:2
06190000 T  0051:0007:3
06191000 T  0051:0009:3
06192000 T  0051:0014:1
06193000 T  0051:0014:1
 51 IS   17 LONG, NEXT SEG    3
06194000 T  0003:0459:2
06195000 T  0003:0459:2
START OF SEGMENT **********   52
06196000 T  0052:0000:0
06197000 T  0052:0000:0
06198000 T  0052:0001:2
06199000 T  0052:0002:0
06200000 T  0052:0003:3
06201000 T  0052:0005:3
06202000 T  0052:0006:0
06203000 T  0052:0006:1
06204000 T  0052:0007:2
```

```
                    END;
                EMITL((L+6) DIV 4-(T1.[24:12]-1) DIV 4);
                EMITB(BBW,BUMPL,T1,[36:12]);
                STEPIT;
                ADJUST;
EXIT:   END SUBHAND;

COMMENT IFEXP COMPILES CONDITIONAL EXPRESSIONS.  IT REPORTS THE TYPE
                OF EXPRESSIONS AS EXPRSS REPORTS;
INTEGER PROCEDURE IFEXP;
                BEGIN INTEGER TYPE,THENBRANCH,ELSEBRANCH;

                    IFCLAUSE;
                STACKCT ← 0;
                    THENBRANCH ← BUMPL;
   COMMENT SAVE L FOR LATER FIXUP;
                    IFEXP ← TYPE ← EXPRSS; COMMENT COMPILE 1ST EXPRSS;
                STACKCT ← 0;
                    ELSEBRANCH ← BUMPL;
                    EMITB(BFC,THENBRANCH,L);
                    IF ELCLASS ≠ ELSEV THEN ERR(155) ELSE BEGIN
                    STEPIT;
                AEXP; STACKCT ← 1;
                    COMMENT THIS COMPILES PROPER TYPE SECOND EXPRSS;
                    EMITB(BFW,ELSEBRANCH,L);
                    EMIT(1); L ← L-1;
   COMMENT THIS IS USED BY EMITLNG TO CLEANUP CODE.  COMPARE WITH
                    BOOSEC, BOOCOMP, AND RELATION;
                    END END IFEXP;

COMMENT PARSE COMPILES CODE FOR THE CONCATENATE;
PROCEDURE PARSE;
                BEGIN INTEGER FIRST,SECOND,THIRD;

                    LABEL EXIT,NEXTCHK;
            IF ELCLASS = FIELDID THEN
              BEGIN
                FIRST := ELBAT[I].SBITF;
                SECOND := 48 - (THIRD := ELBAT[I].NBITF);
                GO TO NEXTCHK;
              END
            ELSE
                    IF ELCLASS = LFTBRKET THEN
            IF STEPI = FIELDID THEN
              BEGIN
                FIRST := ELBAT[I].SBITF;
                SECOND := 48 - (THIRD := ELBAT[I].NBITF);
                IF STEPI ≠ RTBRKET THEN
                  BEGIN
                    ERR(94);
                    GO TO EXIT;
                  END;
                GO TO NEXTCHK;
              END
            ELSE
              IF ELCLASS = LITNO THEN
                    IF STEPI = COLON THEN
```

```
                    06205000 T  0052:0007:3
                    06206000 T  0052:0007:3
                    06207000 T  0052:0011:3
                    06208000 T  0052:0014:0
                    06208500 T  0052:0014:1
                    06209000 T  0052:0015:2
                52 IS   19 LONG, NEXT SEG    3
                    06292000 T  0003:0459:2
                    06293000 T  0003:0459:2
                    06294000 T  0003:0459:2
                    06295000 T  0003:0459:2
              START OF SEGMENT **********   53
                    06296000 T  0053:0000:0
                    06296500 T  0053:0000:1
                    06297000 T  0053:0001:2
                    06298000 T  0053:0003:2
                    06299000 T  0053:0003:2
                    06299500 T  0053:0004:1
                    06300000 T  0053:0005:2
                    06301000 T  0053:0007:2
                    06302000 T  0053:0008:0
                    06303000 T  0053:0010:1
                    06305000 T  0053:0011:2
                    06306000 T  0053:0012:1
                    06307000 T  0053:0012:1
                    06308000 T  0053:0013:3
                    06309000 T  0053:0015:3
                    06310000 T  0053:0015:3
                    06311000 T  0053:0015:3
                53 IS   19 LONG, NEXT SEG    3
                    06312000 T  0003:0459:2
                    06313000 T  0003:0459:2
                    06314000 T  0003:0459:2
              START OF SEGMENT **********   54
%107-               06315000 P  0054:0000:0
%112-               06315100 C  0054:0000:0
%112-               06315200 C  0054:0000:1
%112-               06315300 C  0054:0001:2
%112-               06315400 C  0054:0002:1
%112-               06315500 C  0054:0005:2
%112-               06315600 C  0054:0005:3
%112-               06315700 C  0054:0005:3
                    06316000 T  0054:0005:3
%112-               06316100 C  0054:0007:2
%112-               06316200 C  0054:0008:1
%112-               06316300 C  0054:0009:2
%112-               06316400 C  0054:0010:1
%112-               06316500 C  0054:0013:2
%112-               06316600 C  0054:0014:0
%112-               06316700 C  0054:0014:1
%112-               06316800 C  0054:0015:2
%112-               06316900 C  0054:0015:3
%112-               06317000 P  0054:0015:3
%112-               06317100 C  0054:0016:0
%112-               06317200 C  0054:0016:0
%112-               06317300 C  0054:0016:0
                    06318000 T  0054:0017:3
```

```
                    IF STEPI = LITNO THEN                                                    06319000 T  0054:0019:2
                   IF STEPI = RTBRKET THEN % WE WILL TAKE CARE OF THE RES                    06320000 P  0054:0020:1
                     IF (FIRST := ELBAT[I-3].ADDRESS) ×                  %107-               06320100 C  0054:0022:0
                        (THIRD := ELBAT[I-1].ADDRESS) ≠ 0 THEN           %107-               06320200 C  0054:0024:1
                       BEGIN                                             %107-               06320300 C  0054:0027:2
                         SECOND := 48 - THIRD; % SOURCE IS RIGHT JUST.                       06320400 C  0054:0027:3
                         GO TO NEXTCHK;                                  %107-               06320500 C  0054:0029:2
                       END                                               %107-               06320600 C  0054:0029:3
                     ELSE % BAD BITS, FALL THROUGH TO ERROR.             %107-               06320700 C  0054:0029:3
                   ELSE % MAYBE A COLON.                                 %107-               06320800 C  0054:0029:3
                     IF ELCLASS = COLON THEN %                           %107-               06320900 C  0054:0030:0
                    IF STEPI = LITNO THEN                                                    06321000 T  0054:0031:2
                    IF STEPI = RTBRKET THEN                                                  06322000 T  0054:0032:1
COMMENT IF TEST ARE PASSED THEN SYNTAX IS CORRECT;                                           06323000 T  0054:0034:0
                   IF (FIRST  ← ELBAT[I-5].ADDRESS) ×                                        06324000 T  0054:0034:0
                      (SECOND ← ELBAT[I-3].ADDRESS) ×                                        06325000 T  0054:0036:1
                      (THIRD  ← ELBAT[I-1].ADDRESS) ≠ 0 THEN                                 06326000 T  0054:0039:2
NEXTCHK:          IF FIRST + THIRD ≤ 48 THEN % SO FAR SO GOOD.           %107-               06327000 P  0054:0041:3
                   IF SECOND+ THIRD ≤ 48 THEN                                              06328000 T  0054:0044:0
COMMENT IF TEST ARE PASSED THEN RANGES OF LITERALS ARE O.K.;                                 06329000 T  0054:0046:0
                    BEGIN                                                                    06330000 T  0054:0046:0
                      STEPIT;                                                                06331000 T  0054:0046:1
                       EMITD(SECOND,FIRST,THIRD);                                            06332000 T  0054:0047:2
              STACKCT ← 1;                                                                   06332500 T  0054:0048:0
                       GO TO EXIT END;                                                       06333000 T  0054:0049:2
                   ERR(113); COMMENT ERROR IF SYNTAX OR RANGE FAILS;                         06334000 T  0054:0049:3
EXIT:     END PARSE;                                                                         06335000 T  0054:0050:0
                                                                                  54 IS   54 LONG, NEXT SEG    3
COMMENT DOT COMPILES CODE FOR PARTIAL WORD DESIGNATORS, EXCEPT FOR                           06336000 T  0003:0459:2
          THOSE CASES HANDLED BY THE VARIABLE ROUTINE;                                       06337000 T  0003:0459:2
PROCEDURE DOT;                                                                               06338000 T  0003:0459:2
          BEGIN INTEGER FIRST,SECOND; LABEL EXIT;                                            06339000 T  0003:0459:2
                                                                            START OF SEGMENT **********   55
            IF ELCLASS = PERIOD THEN BEGIN                                                   06340000 T  0055:0000:0
                  IF DOTSYNTAX(FIRST,SECOND) THEN GO TO EXIT;                                06341000 T  0055:0001:2
                                                                                             06342000 T  0055:0003:2
                                                                                             06343000 T  0055:0003:2
          EMITI(0,FIRST,SECOND);                                                             06344000 T  0055:0003:2
                  STEPIT;                                                                    06345000 T  0055:0004:0
   EXIT: END END DOT;                                                                        06346000 T  0055:0004:1
                                                                              55 IS    8 LONG, NEXT SEG    3
PROCEDURE IFCLAUSE;                                                                          06409000 T  0003:0459:2
          BEGIN STEPIT; BEXP;                                                                06410000 T  0003:0459:2
            IF ELCLASS ≠ THENV THEN ERR(116)ELSE STEPIT END IFCLAUS;                         06411000 T  0003:0461:2
COMMENT  PANA COMPILES THE CONSTRUCT: (<ARIT, EXP,>);                                        06412000 T  0003:0464:0
PROCEDURE PANA;                                                                              06413000 T  0003:0464:0
          BEGIN                                                                              06414000 T  0003:0464:0
           IF STEPI ≠ LEFTPAREN THEN ERR(105)                                                06415000 T  0003:0464:0
           ELSE BEGIN STEPIT; AEXP; IF ELCLASS ≠ RTPAREN THEN                                06416000 T  0003:0466:1
              ERR(104) ELSE STEPIT END END PANA;                                             06417000 T  0003:0469:3
COMMENT  BANA COMPILES THE CONSTRUCT:  [<ARITH, EXP,>];                                      06418000 T  0003:0472:0
PROCEDURE BANA;                                                                              06419000 T  0003:0472:0
          BEGIN                                                                              06420000 T  0003:0472:0
           IF STEPI ≠ LFTBRKET THEN ERR(117)                                                 06421000 T  0003:0472:0
              ELSE BEGIN STEPIT; AEXP; IF ELCLASS ≠ RTBRKET THEN                             06422000 T  0003:0473:3
                ERR(118) ELSE STEPIT END END BANA      ;                                     06423000 T  0003:0476:1
COMMENT THIS SECTION CONTAINS THE STATEMENT ROUTINES;                                        07000000 T  0003:0479:2
```

```
 COMMENT COMPOUNDTAIL COMPILES COMPOUNDTAILS.  IT ALSO ELIMINATES                07001000 T  0003:0479:2
              COMMENTS FOLLOWING ENDS.  AFTER ANY ERROR, ERROR MESSAGES          07002000 T  0003:0479:2
              ARE SUPPRESSED.  COMPOUNDTAIL IS PARTIALLY RESPONSIBLE             07003000 T  0003:0479:2
              FOR RESTORING THE ABILITY TO WRITE ERROR MESSAGES.  SOME           07004000 T  0003:0479:2
              CARE IS ALSO TAKEN TO PREVENT READING BEYOND THE "END.";           07005000 T  0003:0479:2
PROCEDURE COMPOUNDTAIL;                                                          07006000 T  0003:0479:2
              BEGIN LABEL ANOTHER;                                               07007000 T  0003:0479:2
                                                                      START OF SEGMENT **********   56
                I ← I-1; BEGINCTR ← BEGINCTR+1;                                  07008000 T  0056:0000:0
 ANOTHER:       ERRORTOG ← TRUE;   COMMENT  ALLOW ERROR MESSAGES;                07009000 T  0056:0002:1
                STEPIT;                                                          07010000 T  0056:0003:3
                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
                   STEPI≤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
PROCEDURE ACTUALPARAPART(SBIT,INDEX); VALUE SBIT,INDEX;                          07035000 T  0003:0479:2
        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
STREAM PROCEDURE WRITEAX(LINE,ACCUM,N,SEQ); VALUE N;                             07038100 T  0057:0000:0
        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
              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(LINE[0],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  0057:0081:2
                    EMITB(BFW,SCLASS,L);                                            07088000 T  0057:0081:3
                    EMITNUM(ACLASS);                                                07089000 T  0057:0083:2
                    EMITPAIR(TAKE(PROINFO),ADDRESS,LOD);                            07090000 T  0057:0083:3
                    EMITO(INX);                                                     07091000 T  0057:0085:3
```

```
                    EMITN(512);
                    EMITD(33,18,15);
                    EMIT(0);
                    EMITD(5,5,1);
          COMMON:   PCTR ← PCTR+1;
                    IF ELCLASS = COMMA THEN GO TO ANOTHER;
                    IF ELCLASS ≠ RTPAREN THEN
                         BEGIN ERR(129); GO TO EXIT END;
                    IF TAKE(INDEX).NODIMPART+1 ≠ PCTR THEN
                         BEGIN ERR(128); GO TO EXIT END;
                    STEPIT;
                    STACKCT ← 0;
     EXIT:    END ACTUAL PARAPART;

     PROCEDURE PROCSTMT(FROM); VALUE FROM; BOOLEAN FROM;
             BEGIN
                    REAL HOLE,ADDRESS;

                    REAL J; LABEL OK;
                    LABEL EXIT;
                    SCATTERELBAT;
                    HOLE← ELBAT[I];
                    ADDRESS ← ADDRSF;
                    IF NESTOG THEN
                    IF MODE≠0 THEN
                    IF TABLE(I+1)≠ASSIGNOP THEN
                    BEGIN FOR J←CALLINFO STEP 1 UNTIL CALLX DO
                         IF CALL[J]=ADDRESS THEN GO TO OK;
                       CALL[CALLX←CALLX+1]←ADDRESS;
          OK:       END;
                    CHECKFR(HOLE);
                    IF ELCLASS ≠ PROCID THEN
                    IF NOT FORMALF THEN
                    IF TABLE(I+1) = ASSIGNOP THEN
                         BEGIN VARIABLE(2-REAL(FROM)); GO TO EXIT END;
          COMMENT   CALL VARIABLE TO HANDLE THIS ASSIGNMENT OPERATION;
                    IF ELCLASS ≠ PROCID EQV FROM
                         THEN BEGIN ERR(159); GO TO EXIT END;
          COMMENT   IT IS PROCEDURE IF AND ONLY WE COME FROM STMT;
                    STEPIT;
                    EMITO(MKS);
                    IF ELCLASS = LEFTPAREN
                         THEN ACTUALPARAPART(FALSE,GIT(HOLE))
                         ELSE IF FORMALF THEN L ← L-1
                           ELSE IF TAKE(GIT(HOLE)).NODIMPART≠0 THEN ERR(128);
                    EMITV(ADDRESS);
     EXIT:    END  PROCSTMT;

     PROCEDURE STRMPROCSTMT;
             BEGIN REAL WHOLE,FIX,T1;



                    WHOLE ← ELBAT[I]; FIX ← -1;
                    IF ELCLASS ≠ STRPROCID THEN EMIT(0);
                    IF WHOLE. LVL  ≠ 1 THEN
                         BEGIN FIX ← L; L ← L+1 END;
```

```
07092000 T  0057:0086:1
07093000 T  0057:0087:2
07093100 T  0057:0088:1
07093200 T  0057:0089:2
07094000 T  0057:0090:1
07095000 T  0057:0092:0
07096000 T  0057:0093:3
07097000 T  0057:0094:0
07098000 T  0057:0096:0
07099000 T  0057:0098:0
07100000 T  0057:0100:0
07100500 T  0057:0100:1
07101000 T  0057:0101:2
57 IS  106 LONG, NEXT SEG    3
07391000 T  0003:0479:2
07392000 T  0003:0479:2
07393000 T  0003:0479:2
START OF SEGMENT **********   58
07393100 T  0058:0000:0
07394000 T  0058:0000:0
07395000 T  0058:0000:0
07396000 T  0058:0000:1
07397000 T  0058:0001:3
07397100 T  0058:0002:0
07397200 T  0058:0003:2
07397210 T  0058:0004:0
07397300 T  0058:0006:1
07397400 T  0058:0008:0
07397500 T  0058:0014:0
07397600 T  0058:0018:1
07398000 T  0058:0019:2
07399000 T  0058:0019:3
07400000 T  0058:0020:1
07401000 T  0058:0021:3
07402000 T  0058:0023:3
07403000 T  0058:0028:1
07404000 T  0058:0028:1
07405000 T  0058:0029:2
07406000 T  0058:0031:3
07407000 T  0058:0031:3
07408000 T  0058:0032:0
07409000 T  0058:0032:1
07410000 T  0058:0033:2
07411000 T  0058:0035:2
07412000 T  0058:0037:2
07413000 T  0058:0042:0
07425000 T  0058:0042:1
58 IS   46 LONG, NEXT SEG    3
07426000 T  0003:0479:2
07427000 T  0003:0479:2
START OF SEGMENT **********   59
07428000 T  0059:0000:0
07429000 T  0059:0000:0
07430000 T  0059:0000:0
07431000 T  0059:0002:0
07432000 T  0059:0004:0
07433000 T  0059:0005:2
```

```
                EMITO(MKS);
                T1 ← TAKEFRST.[1:6];
                FOR GT1 ← 1 STEP 1 UNTIL T1 DO EMIT(0);
                IF STEPI ≠ LEFTPAREN THEN ERR(128)
                ELSE BEGIN ACTUALPARAPART(TRUE,GIT(WHOLE));
                     IF FIX < 0 THEN EMITV(WHOLE.ADDRESS)
                     ELSE BEGIN T1 ← L; L ← FIX;
                          WHOLE ← TAKE(GIT(WHOLE));
                          EMITNUM(T1+2=WHOLE.[16:12]);
                          L ← T1;
                          EMITB(BBW,BUMPL,WHOLE.[28:12]);
                     END;
           END END STRMPROCSTMT;

INTEGER PROCEDURE BAE;
           BEGIN BAE ← BUMPL; CONSTANTCLEAN; ADJUST END BAE;
COMMENT RELSESTMT COMPILES THE RELEASE STATEMENT;
COMMENT  DOSTMT HANDLES THE DO STATEMENT;
PROCEDURE DOSTMT;
           BEGIN INTEGER TL;

                FOULED ← L;

                STEPIT; TL←L; STMT; IF ELCLASS ≠ UNTILV THEN ERR(131)
                ELSE BEGIN
                  STEPIT; BEXP; EMITB(BBC,BUMPL,TL) END
           END DOSTMT;

COMMENT  WHILESTMT COMPILES THE WHILE STATEMENT;
PROCEDURE WHILESTMT;
           BEGIN INTEGER BACK,FRONT;

                FOULED ← L;

                STEPIT; BACK ← L; BEXP; FRONT ← BUMPL;
                IF ELCLASS ≠ DOV THEN ERR(132) ELSE
                  BEGIN STEPIT; STMT; EMITB(BBW,BUMPL,BACK);
                     CONSTANTCLEAN; EMITB(BFC,FRONT,L) END END WHILESTMT;

COMMENT  GOSTMT COMPILES GO TO STATEMENTS.  GOSTMT LOOKS AT THE
           EXPRESSION.  IF IT IS SIMPLE ENOUGH WE GO DIRECTLY,
           OTHERWISE A CALL ON THE MCP IS GENERATED IN ORDER TO GET
           STORAGE RETURNED.  SEE DEXP AND GENGO;
PROCEDURE GOSTMT;
           BEGIN
                REAL ELBW;

                LABEL GOMCP,EXIT;
                IF STEPI = TOV THEN STEPIT;
                IF ELCLASS = LABELID THEN TB1 ← TRUE
                ELSE IF ELCLASS = SWITCHID THEN TB1 ← FALSE
                ELSE BEGIN IF ELCLASS = POLISHV THEN
                          BEGIN POLISHER(1); EMITO(BFW) END
                          ELSE ERR(501);
                          GO TO EXIT
                     END;
                IF NOT LOCAL(ELBAT[I]) THEN
```

```
                                                  07434000 T  0059:0007:3
                                                  07435000 T  0059:0008:1
                                                  07436000 T  0059:0010:0
                                                  07437000 T  0059:0014:0
                                                  07438000 T  0059:0015:3
                                                  07439000 T  0059:0018:0
                                                  07440000 T  0059:0019:3
                                                  07441000 T  0059:0022:1
                                                  07442000 T  0059:0024:1
                                                  07443000 T  0059:0026:1
                                                  07444000 T  0059:0027:3
                                                  07445000 T  0059:0030:0
                                                  07446000 T  0059:0030:0
                                            59 IS    33 LONG, NEXT SEG    3
                                                  07458000 T  0003:0479:2
                                                  07459000 T  0003:0479:2
                                                  07460000 T  0003:0484:0
                                                  07481000 T  0003:0484:0
                                                  07482000 T  0003:0484:0
                                                  07483000 T  0003:0484:0
                                         START OF SEGMENT *********   60
                                                  07483500 T  0060:0000:0
                                                  07484000 T  0060:0000:1
                                                  07485000 T  0060:0000:1
                                                  07486000 T  0060:0004:0
                                                  07487000 T  0060:0005:2
                                                  07488000 T  0060:0008:0
                                            60 IS    11 LONG, NEXT SEG    3
                                                  07489000 T  0003:0484:0
                                                  07490000 T  0003:0484:0
                                                  07491000 T  0003:0484:0
                                         START OF SEGMENT *********   61
                                                  07491500 T  0061:0000:0
                                                  07492000 T  0061:0000:1
                                                  07493000 T  0061:0000:1
                                                  07494000 T  0061:0004:0
                                                  07495000 T  0061:0006:0
                                                  07496000 T  0061:0010:0
                                            61 IS    14 LONG, NEXT SEG    3
                                                  07497000 T  0003:0484:0
                                                  07498000 T  0003:0484:0
                                                  07499000 T  0003:0484:0
                                                  07500000 T  0003:0484:0
                                                  07501000 T  0003:0484:0
                                                  07502000 T  0003:0484:0
                                                  07503000 T  0003:0484:0
                                         START OF SEGMENT *********   62
                                                  07504000 T  0062:0000:0
                                                  07505000 T  0062:0000:0
                                                  07506000 T  0062:0002:0
                                                  07507000 T  0062:0003:3
                                                  07511000 T  0062:0006:0
                                                  07512000 T  0062:0007:3
                                                  07513000 T  0062:0009:3
                                                  07514000 T  0062:0011:2
                                                  07515000 T  0062:0011:3
                                                  07516000 T  0062:0011:3
```

```
                    BEGIN                                                                    07516100 T  0062:0012:1
                         IF TB1 THEN                                                         07516200 T  0062:0013:2
                              BEGIN EMITV(GNAT(ELBAT[I]));                                   07516300 T  0062:0013:3
                                   EMITO(BFW);                                               07516400 T  0062:0015:3
                                   STEPIT;                                                   07516500 T  0062:0016:0
                                   GO TO EXIT END;                                           07516600 T  0062:0016:1
                         BEGIN ERR(501); GO TO EXIT END;                                     07517000 T  0062:0017:2
                         END;                                                                07517500 T  0062:0018:1
               IF TB1 THEN BEGIN GOGEN(ELBAT[I],BFW); STEPIT;                                07518000 T  0062:0018:1
                    CONSTANTCLEAN; GO EXIT END                                             07519000 T  0062:0021:2
               ELSE BEGIN                                                                  07520000 T  0062:0022:0
                    ELBW ← ELBAT[I];                                                       07521000 T  0062:0022:1
                                                                                           07522000 T  0062:0023:3
                    BANA;                                                                  07523000 T  0062:0023:3
                    EMITO(DUP);                                                            07524000 T  0062:0024:0
                    EMITO(ADD);                                                            07525000 T  0062:0024:1
                         EMITO(BFW);                                                       07526000 T  0062:0025:3
                    GT3 ← TAKE(GT4←GIT(ELBW))+GT4;                                         07527000 T  0062:0026:0
                    FOR GT4 ← GT4+1 STEP 1 UNTIL GT3 DO                                    07528000 T  0062:0029:2
                         GOGEN(TAKE(GT4),BFW);                                             07529000 T  0062:0033:2
               END;                                                                        07530000 T  0062:0035:2
EXIT: END GOSTMT;                                                                          07531000 T  0062:0035:2
                                                                                     62 IS    39 LONG, NEXT SEG    3
PROCEDURE GOGEN(LABELBAT,BRANCHTYPE);                                                      07535000 T  0003:0484:0
          VALUE LABELBAT,BRANCHTYPE;                                                       07536000 T  0003:0484:0
          REAL LABELBAT,BRANCHTYPE;                                                        07537000 T  0003:0484:0
          BEGIN                                                                            07538000 T  0003:0484:0
            IF BOOLEAN(GT1←TAKE(GT2←GIT(LABELBAT))),[1:1]                                  07539000 T  0003:0484:0
              THEN EMITB(BRANCHTYPE,BUMPL,GT1,[36:12])                                     07540000 T  0003:0487:3
   COMMENT  LABELR SETS THE SIGN OF THE ADDITIONAL INFO FOR A LABEL                        07541000 T  0003:0490:0
            NEGATIVE WHEN THE LABEL IS ENCOUNTERED.  SO THIS MEANS                         07542000 T  0003:0490:1
            THAT WE NOW KNOW WHERE TO GO;                                                  07543000 T  0003:0490:1
               ELSE BEGIN EMIT(GT1); EMIT(BRANCHTYPE);                                     07544000 T  0003:0490:1
               PUT(GT1&L[36:36:12],GT2) END END GOGEN;                                     07545000 T  0003:0493:2
COMMENT SIMPGO IS USED ONLY BY THE IF STMT ROUTINE.  IT DETERMINES IF                      07546000 T  0003:0495:2
            A STATEMENT IS A SIMPLE GO TO STATEMENT;                                       07547000 T  0003:0495:2
BOOLEAN PROCEDURE SIMPGO;                                                                  07548000 T  0003:0495:2
        BEGIN LABEL EXIT;                                                                  07549000 T  0003:0495:2
                                                                                     START OF SEGMENT **********   63
            IF ELCLASS = GOV                                                               07550000 T  0063:0000:0
              THEN BEGIN                                                                   07551000 T  0063:0000:0
                 IF STEPI = TOV THEN STEPIT;                                               07552000 T  0063:0001:2
                 IF ELCLASS = LABELID THEN                                                 07553000 T  0063:0003:2
                 IF LOCAL(ELBAT[I]) THEN                                                   07554000 T  0063:0004:0
                    BEGIN SIMPGO ← TRUE; GO EXIT END;                                      07555000 T  0063:0005:3
                      I ← I-1; ELCLASS ← GOV END;                                          07556000 T  0063:0007:2
 EXIT:  END SIMPGO;                                                                        07557000 T  0063:0009:2
                                                                                      63 IS    13 LONG, NEXT SEG    3
COMMENT IFSTMT COMPILES IF STATEMENTS.  SPECIAL CARE IS TAKEN TO                           07558000 T  0003:0495:2
            OPTIMIZE CODE IN THE NEIGHBORHOOD OF THE JUMPS.  TO SOME                       07559000 T  0003:0495:2
            EXTENT SUPPERFULOUS BRANCHING IS AVOIDED;                                      07560000 T  0003:0495:2
PROCEDURE IFSTMT;                                                                          07561000 T  0003:0495:2
        BEGIN REAL T1,T2; LABEL EXIT;                                                      07562000 T  0003:0495:2
                                                                                     START OF SEGMENT **********   64
            IFCLAUSE;                                                                      07563000 T  0064:0000:0
            IF SIMPGO                                                                      07564000 T  0064:0000:1
              THEN BEGIN                                                                   07565000 T  0064:0000:1
```

```
                    T1 ← ELBAT[I];                                                  07566000 T  0064:0001:3
               IF STEPI = ELSEV                                                     07567000 T  0064:0002:1
                     THEN BEGIN                                                     07568000 T  0064:0003:2
                        STEPIT;                                                     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
               STEPIT; 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
               STEPIT;                                                              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
                    STEPIT; 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
     BEGIN LABEL EXIT, ROUND;                                                       07594000 T  0003:0495:2
                                                                      START OF SEGMENT **********   65
DEFINE ELBATWORD=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
               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
%*                                                                         *%                              07646080 C  0003:0495:2
%* THE CODE GENERATED FOR THIS STATEMENT IS:                               *%                              07646085 C  0003:0495:2
%*                                                                         *%                              07646090 C  0003:0495:2
%*                            <AEXP>                                       *%                              07646095 C  0003:0495:2
%*                            LITC  ?                                      *%                              07646100 C  0003:0495:2
%*                            BFW         BRANCH TO BRANCH TABLE           *%                              07646105 C  0003:0495:2
%*                            STMT 0                                       *%                              07646110 C  0003:0495:2
%*                            LITC  ?                                      *%                              07646115 C  0003:0495:2
%*                            BFW         BRANCH TO RESUME                 *%                              07646120 C  0003:0495:2
```

```
%*                          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-
BEGIN                                                                   %115-
  ARRAY CASEADDRESS[0:99];     % THIS ARRAY HOLDS THE RELATIVE           %115-
                               % ADDRESS OF THE BEGINNING OF EACH       %115-
                               % CASE.                                  %115-
  INTEGER                                                               %115-
    LINK,                      % HOLDS RELATIVE ADDRESS OF FIRST OF      %115-
                               % BRANCHES THAT MUST BE FIXED UP TO      %115-
                               % BRANCH TO THE RESUME POINT.            %115-
    ADR,                       % HOLDS RELATIVE ADDRESS OF THE BRANCH
                               % AROUND THE CASE STATEMENTS TO THE      %115-
                               % BRANCH TABLE. THIS BRANCH GETS FIXED
                               % UP WHEN WE FIND WHERE THE BRANCH TABLE
                               % GOES.                                  %115-
    N,                         % COUNT OF NUMBER OF CASE STATEMENT      %115-
                               % ENCOUNTERED.                           %115-
```

```
07646125 C  0003:0495:2
07646130 C  0003:0495:2
07646135 C  0003:0495:2
07646140 C  0003:0495:2
07646145 C  0003:0495:2
07646150 C  0003:0495:2
07646155 C  0003:0495:2
07646160 C  0003:0495:2
07646165 C  0003:0495:2
07646170 C  0003:0495:2
07646175 C  0003:0495:2
07646180 C  0003:0495:2
07646185 C  0003:0495:2
07646190 C  0003:0495:2
07646195 C  0003:0495:2
07646200 C  0003:0495:2
07646205 C  0003:0495:2
07646210 C  0003:0495:2
07646215 C  0003:0495:2
07646220 C  0003:0495:2
07646225 C  0003:0495:2
07646230 C  0003:0495:2
07646235 C  0003:0495:2
07646240 C  0003:0495:2
07646245 C  0003:0495:2
07646250 C  0003:0495:2
07646255 C  0003:0495:2
07646260 C  0003:0495:2
07646265 C  0003:0495:2
07646270 C  0003:0495:2
07646275 C  0003:0495:2
07646280 C  0003:0495:2
07646285 C  0003:0495:2
07646290 C  0003:0495:2
07646295 C  0003:0495:2
07646300 C  0003:0495:2
07646305 C  0003:0495:2
07646310 C  0003:0495:2
07646315 C  0003:0495:2
07646320 C  0003:0495:2
07646325 C  0003:0495:2
07646330 C  0003:0495:2
07646335 C  0003:0495:2
                    START OF SEGMENT **********   66
07646340 C  0066:0001:3
07646345 C  0066:0001:3
07646350 C  0066:0001:3
07646355 C  0066:0001:3
07646360 C  0066:0001:3
07646365 C  0066:0001:3
07646370 C  0066:0001:3
07646375 C  0066:0001:3
07646380 C  0066:0001:3
07646385 C  0066:0001:3
07646390 C  0066:0001:3
07646395 C  0066:0001:3
07646400 C  0066:0001:3
```

```
    ENDOFIT,                    % ADDRESS OF RESUME POINT                    %115-   07646405 C  0066:0001:3
    J,                          % TEMPORARY                                  %115-   07646410 C  0066:0001:3
    K;                          % TEMPORARY                                  %115-   07646415 C  0066:0001:3
                                %                                           %115-   07646420 C  0066:0001:3
  BOOLEAN                                                                   %115-   07646425 C  0066:0001:3
    GOTOG;                      % TRUE IF WE ARE COMPILING A SIMPLE          %115-   07646430 C  0066:0001:3
                                % GO TO.                                     %115-   07646435 C  0066:0001:3
  LABEL XIT;                                                                %115-   07646436 C  0066:0001:3
%                                                                           %115-   07646440 C  0066:0001:3
%                                                                           %115-   07646445 C  0066:0001:3
    STEPIT;   % STEP OVER "CASE"                                           %115-   07646450 C  0066:0001:3
    AEXP;     % GENERATE CODE FOR CASE INDEX                                %115-   07646455 C  0066:0002:0
    IF STEPI ≠ BEGINV THEN % NOTICE WE JUST JUMPED OVER "OF"                %115-   07646456 C  0066:0002:1
      BEGIN                                                                 %115-   07646457 C  0066:0003:3
        ERR(70); GO TO XIT;                                                 %115-   07646458 C  0066:0004:0
      END;                                                                  %115-   07646459 C  0066:0005:3
    EMIT(0);  % GENERATE DUMMY BRANCH TO BRANCH TABLE, WILL FIX             %115-   07646460 C  0066:0005:3
    EMITO(BFW);  %    IT UP LATER.                                          %115-   07646465 C  0066:0006:0
    ADR := L;                                                               %115-   07646470 C  0066:0007:2
    WHILE STEPI ≠ ENDV DO                                                   %115-   07646475 C  0066:0007:3
      BEGIN   % PROCESSING CASE STATEMENTS                                  %115-   07646480 C  0066:0009:3
        ERRORTOG := TRUE;                                                   %115-   07646485 C  0066:0009:3
        IF ELCLASS = SEMICOLON THEN % NULL STATEMENT, NO CODE NEEDED                07646490 C  0066:0010:0
          N := N + 1  % THIS LEAVES A ZERO IN CASEADDRESS[N]                %115-   07646495 C  0066:0011:2
        ELSE                                                                %115-   07646500 C  0066:0012:0
          BEGIN                                                             %115-   07646505 C  0066:0012:1
            CASEADDRESS[N] := L; % REMEMBER BEGINNING ADDRESS OF            %115-   07646510 C  0066:0013:2
            N := N + 1;          % THIS CASE.                               %115-   07646515 C  0066:0014:1
            IF (GOTOG := SIMPGO) THEN                                       %115-   07646520 C  0066:0015:3
              ELBAT[I:=I"1] := ELCLASS := GOV; % REMEMBER IF SIMPLE         %115-   07646525 C  0066:0016:1
            STMT; % PROCESS THE STATEMENT                                   %115-   07646530 C  0066:0020:0
            IF ELCLASS = SEMICOLON THEN                                     %115-   07646535 C  0066:0020:1
              IF NOT GOTOG THEN % GENERATE DUMMY BRANCH TO RESUME           %115-   07646540 C  0066:0021:2
                BEGIN                                                       %115-   07646545 C  0066:0022:0
                  EMIT(LINK);                                               %115-   07646550 C  0066:0022:1
                  EMITO(BFW);                                               %115-   07646555 C  0066:0023:3
                  LINK := L;                                                %115-   07646560 C  0066:0024:0
                END                                                         %115-   07646565 C  0066:0025:2
              ELSE % SIMPLE GO TO, NO CODE TO GENERATE                      %115-   07646570 C  0066:0025:2
            ELSE                                                            %115-   07646575 C  0066:0025:2
              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
    STEPI ≤ 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
  EMITO(DUP);           % GENERATE CODE TO MULTIPLY INDEX                   %115-   07646610 C  0066:0036:0
  EMITO(ADD);           % BY TWO AND BRANCH INTO BRANCH TABLE               %115-   07646615 C  0066:0036:1
  EMITO(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
    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
        BEGIN                                                                          07654000 T  0067:0000:0
        SI:=ACC; SI:=SI+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
    LABEL L1;                                                                          07658000 T  0067:0008:1
    STREAM PROCEDURE ZEERO(D);                                                         07658500 T  0067:0008:1
        BEGIN                                                                          07659000 T  0067:0008:1
        DI:=D;DS:=8 LIT"00000000";                                                     07659500 T  0067:0009:2
        SI:=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
                    IF COCT(COUNT-3,48-(COUNT-3)×3,ACCUM[1],
                      CODE(T2))=0 THEN FLAG(303)
                    END
               ELSE BEGIN  ERROR(302); GO TO L1  END ;
               IF STEPI≠COMMA THEN GO TO L1
               END;
          ERROR(54);
          END;
L1:
     RIGHT(SIZE×4);
     STRFAMTOG:=FALSE;
     SEGMENT(SIZE,0);
     PROGDESCBLDR(ADDRSF,TRUE,SIZE,DDES);
     END FILLSTMT;

 PROCEDURE STMT;
          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, L40,
               L41, L42, L43, L44, L45, L46, L47, L48, L49, L50,
               L51, L52, L53, L54;
               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, L40,
               L41, L42, L43, L44, L45, L46, L47, L48, L49, L50,
               L51, L52, L53, L54;
               LABEL AGAIN,EXIT;
               STACKCT ← 0;
     AGAIN:    GO TO S[ELCLASS];
               IF ELCLASS = COLON THEN
                    BEGIN STEPIT; GT1 ← L;
                         IF ELCLASS = COLON THEN
                              BEGIN ADJUST; I ← I-1 END
                         ELSE IF ELCLASS = LITNO THEN L ← 4×C
                         ELSE I ← I-1;
                         IF L < GT1 OR STEPI ≠ COLON THEN
                              BEGIN ERR(504); GO TO EXIT END;
                         STEPIT;
                         GO TO AGAIN;
                    END;
               IF ELCLASS = 0 THEN FLAG(100); FLAG(145);
L1:L2:L3:L4:L5:L6:L9:L11:L13:L14:L15:L16:L17:L20:L21:L25:L28:L29:L24:
L33:L34:L35:L36:L37:L39:
               ERR(144); GO TO EXIT;
     L7:L8:
               SUBHAND(TRUE); GO TO EXIT;
     L10:L18:L19:
               PROCSTMT(TRUE); GO TO EXIT;
     L12:
               STRMPROCSTMT; GO TO EXIT;
     L22:L23:L26:L27:L30:L31:
```

```
07670500 T  0067:0052:0
07671000 T  0067:0052:1
07671500 T  0067:0056:1
07672000 T  0067:0061:2
07672500 T  0067:0061:3
07673000 T  0067:0064:0
07673500 T  0067:0065:2
07674000 T  0067:0068:0
07674500 T  0067:0068:1
07675000 T  0067:0068:1
07675500 T  0067:0069:2
07676000 T  0067:0070:0
07676500 T  0067:0071:2
07677000 T  0067:0072:0
07677500 T  0067:0073:3
  67 IS   76 LONG, NEXT SEG    3
07711000 T  0003:0495:2
07712000 T  0003:0495:2
START OF SEGMENT **********   68
07713000 T  0068:0000:0
07714000 T  0068:0000:0
07715000 T  0068:0000:0
07716000 T  0068:0000:0
07717000 T  0068:0000:0
07718000 T  0068:0000:0
07719000 T  0068:0000:0
07720000 T  0068:0002:1
07721000 T  0068:0002:1
07722000 T  0068:0002:1
07723000 T  0068:0002:1
07724000 T  0068:0002:1
07725000 T  0068:0002:1
07726000 T  0068:0030:0
07726990 T  0068:0030:0
07727000 T  0068:0031:3
07727010 T  0068:0034:0
07727020 T  0068:0034:1
07727030 T  0068:0036:1
07727040 T  0068:0037:2
07727050 T  0068:0039:3
07727060 T  0068:0041:3
07727070 T  0068:0044:0
07727080 T  0068:0046:0
07727090 T  0068:0048:0
07727100 T  0068:0048:1
07727110 T  0068:0049:2
07728000 T  0068:0049:2
07729000 T  0068:0051:3
07730000 T  0068:0052:0
07732000 T  0068:0052:0
07732000 T  0068:0053:2
07733000 T  0068:0054:0
07734000 T  0068:0055:2
07735000 T  0068:0056:0
07736000 T  0068:0057:2
07737000 T  0068:0058:0
07738000 T  0068:0059:2
```

```
                    VARIABLE(FS); GO TO EXIT;                                                    07739000 T  0068:0059:2
L32:                                                                                             07740000 T  0068:0060:0
                    LABELR; GO TO AGAIN;                                                         07741000 T  0068:0061:2
L38:                                                                                             07742000 T  0068:0062:0
                    POLISHER(0); GO TO EXIT;                                                     07743000 T  0068:0062:0
L40:                                                                                             07744000 T  0068:0063:2
                    IF ELBAT[I],ADDRESS = STREAMV THEN                                           07745000 T  0068:0064:0
                         BEGIN INLINE; GO TO EXIT END;                                           07746000 T  0068:0065:3
                    FLAG(146);                                                                   07747000 T  0068:0067:2
                    IF TABLE(I-2) = ENDV AND MODE > 0 THEN                                       07748000 T  0068:0067:3
                         BEGIN I ← I-2; ELCLASS ← ENDV; GO TO EXIT END;                          07749000 T  0068:0070:1
                    I ← I-1; ERRORTOG ← TRUE; BLOCK(FALSE);                                      07750000 T  0068:0073:3
                    ELCLASS ← TABLE(I←I-1); GO TO EXIT;                                          07751000 T  0068:0076:0
L42:                                                                                             07752000 T  0068:0079:2
                    DBLSTMT; GO TO EXIT;                                                         07753000 T  0068:0079:2
L43:                                                                                             07754000 T  0068:0080:0
                    FORSTMT; GO TO EXIT;                                                         07755000 T  0068:0080:0
L44:                                                                                             07756000 T  0068:0081:2
                    WHILESTMT; GO TO EXIT;                                                       07757000 T  0068:0081:2
L45:                                                                                             07758000 T  0068:0082:0
                    DOSTMT; GO TO EXIT;                                                          07759000 T  0068:0082:0
L49:                                                                                %115-        07759100 C  0068:0083:2
                    CASESTATEMENT; GO TO EXIT;                                      %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
                                                                                                 07991000 T  0003:0495:2
PROCEDURE IOSTMT;                                                                                07993000 T  0003:0495:2
         IF STEPI ≠ LITNO OR (GT1←ELBAT[I],ADDRESS)>15 THEN ERR(98)ELSE                          07994000 T  0003:0495:2
         BEGIN EMIT(ELBAT[I-1],ADDRESS&GT1[41:47:1]&GT1[36:44:3]);                               07995000 T  0003:0500:1
              STEPIT                                                                             07996000 T  0003:0505:2
         END SCOPE STATEMENT;                                                                    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
                   BRET;                                                                         08011000 T  0069:0000:0
         OWN BOOLEAN SIGNA,SIGNB,SIGNC, INT,                                                     08012000 T  0069:0000:0
```

```
                    CONSTANA,CONSTANB,CONSTANC;                                  08013000 T  0069:0000:0
               DEFINE SIMPLEB = SIGNC#,          FORMALV = SIGNA#,                08014000 T  0069:0000:0
                      SIMPLEV = CONSTANA#, A = V#, Q = REGO#,                     08015000 T  0069:0000:0
                      OPDC = TRUE#, DESC = FALSE#, K = BRET#;                     08016000 T  0069:0000:0
               LABEL EXIT;                                                        08017000 T  0069:0000:0
COMMENT PLUG EMITS EITHER AN OPERAND CALL ON A VARIABLE OR A CALL ON A            08018000 T  0069:0000:0
               CONSTANT DEPENDING ON THE REQUIREMENTS;                            08019000 T  0069:0000:0
PROCEDURE PLUG(C,A); VALUE C,A; REAL A; BOOLEAN C;                                08020000 T  0069:0000:0
               IF C THEN EMITNUM(A) ELSE EMITV(A.ADDRESS);                        08021000 T  0069:0000:0
COMMENT SIMPLE DETERMINES IF AN ARITHMETIC EXPRESSION IS + OR - A                 08022000 T  0069:0003:3
               CONSTANT OR A SIMPLE VARIABLE.  IT MAKES A THROUGH REPORT          08023000 T  0069:0003:3
               ON ITS ACTIVITY.  IT ALSO MAKES PROVISION FOR THE RESCAN           08024000 T  0069:0003:3
               OF ELBAT (THIS IS THE ACTION WITH K = SEE CODE IN THE              08025000 T  0069:0003:3
               TABLE ROUTINE FOR FURTHER DETAILS);                                08026000 T  0069:0003:3
BOOLEAN PROCEDURE SIMPLE(B,A,S); BOOLEAN B,S; REAL A;                             08027000 T  0069:0003:3
               BEGIN                                                              08028000 T  0069:0003:3
                    S ← IF STEPI ≠ ADOP THEN FALSE ELSE ELBAT[I].ADDRESS          08029000 T  0069:0003:3
                                 = SUB;                                           08030000 T  0069:0006:1
                    IF ELCLASS = ADOP THEN STEPIT;                                08031000 T  0069:0008:1
                    IF ELCLASS ≥ NONLITNO AND ELCLASS ≤ STRNGCON                  08032000 T  0069:0010:0
                      THEN BEGIN K ← K+1; SIMPLE ← TRUE;                          08033000 T  0069:0011:2
                         ELBAT[I] ← 0&COMMENTV[2:41:7]&K[16:37:11];               08034000 T  0069:0014:1
                         INFO[0,K] ← A ← C; B ← TRUE END                          08035000 T  0069:0017:3
                    ELSE BEGIN                                                    08036000 T  0069:0021:3
                     B ← FALSE; A ← ELBAT[I];                                     08037000 T  0069:0022:0
                     SIMPLE ← REALID ≤ ELCLASS AND ELCLASS ≤ INTID END;           08038000 T  0069:0024:0
               STEPIT END SIMPLE;                                                 08039000 T  0069:0026:1
COMMENT TEST EMITS THE STEP-UNTIL ELEMENT TEST;                                   08040000 T  0069:0029:2
PROCEDURE TEST;                                                                   08041000 T  0069:0029:2
          BEGIN                                                                   08042000 T  0069:0029:2
               IF NOT CONSTANB THEN                                               08043000 T  0069:0029:2
                 BEGIN EMITO(SUB); IF SIMPLEB THEN EMITV(B,ADDRESS)               08044000 T  0069:0030:1
                    ELSE BEGIN                                                    08045000 T  0069:0032:1
                      EMITL(2+L-BRET);                                            08046000 T  0069:0034:0
                      EMITB(BBW,BUMPL,B);                                         08047000 T  0069:0036:0
                    END;                                                          08048000 T  0069:0038:0
                    EMITO(MUL); EMIT(0) END;                                      08049000 T  0069:0038:0
                    EMITO(IF SIGNB THEN GEQ ELSE LEQ); EMIT (0); L←L-1            08050000 T  0069:0039:3
          END TEST;                                                               08051000 T  0069:0043:2
BOOLEAN PROCEDURE SIMPI(ALL); VALUE ALL; REAL ALL;                                08052000 T  0069:0044:0
          BEGIN                                                                   08053000 T  0069:0044:0
               CHECKER(VRET,ALL);                                                 08054000 T  0069:0044:0
               ADDRES ← ALL.ADDRESS;                                              08055000 T  0069:0046:0
               FORMALV ← ALL.[9:2] = 2;                                           08056000 T  0069:0047:3
               IF T ← ALL.CLASS > INTARRAYID OR T < BOOID OR                      08057000 T  0069:0049:2
                  GT1 ← (T-BOOID) MOD 4 < 1 THEN                                  08058000 T  0069:0052:0
                    ERR(REAL(T ≠ 0) × 51 + 100);                                  08059000 T  0069:0054:1
               INT ← GT1 = 2;                                                     08060000 T  0069:0057:2
               SIMPI ← T ≤ INTID END SIMPI;                                       08061000 T  0069:0058:1
COMMENT  STORE EMITS THE CODE FOR THE STORE INTO THE FOR INDEX;                   08062000 T  0069:0062:0
PROCEDURE STORE(S); VALUE S; BOOLEAN S;                                           08063000 T  0069:0062:0
               BEGIN                                                              08064000 T  0069:0062:0
                 IF FORMALV THEN BEGIN EMITO(XCH); S ← FALSE END                  08065000 T  0069:0062:0
                  ELSE BEGIN                                                      08066000 T  0069:0065:2
                    EMITL(ADDRES);                                                08067000 T  0069:0065:3
                    IF ADDRES  > 1023 THEN EMITO(PRTE) END;                       08068000 T  0069:0066:1
                     T  ← (REAL(S)+1)×16;                                         08069000 T  0069:0068:1
```

```
                    EMITO((IF INT THEN I+512 ELSE 4×I)+4) END STORE;
COMMENT CALL EFFECTS A CALL ON THE INDEX;
PROCEDURE CALL(S); VALUE S; BOOLEAN S;
              BEGIN
               IF SIMPLEV
                 THEN IF S THEN EMITV(ADDRES) ELSE EMITN(ADDRES)
                 ELSE BEGIN
                   EMITL(2+L-VRET);
                   EMITB(BBW,BUMPL,V);
                   IF S THEN EMITO(LOD) END END CALL;
PROCEDURE FORLIST(NUMLE); VALUE NUMLE; BOOLEAN NUMLE;
        BEGIN
PROCEDURE FIX(STORE,BACK,FORWART,START);

        VALUE STORE,BACK,FORWART,START;
        REAL  STORE,BACK,FORWART,START;
        BEGIN
             EMITB(GET(FORWART-1),FORWART,START);
             IF RETURNSTORE ≠ 0
               THEN BEGIN
                  L ← STORE; EMITNUM(B-BACK);
                  EMITPAIR(RETURNSTORE,STD) END END FIX;
             INTEGER BACKFIX, FORWARDBRANCH, FOOT, STOREFIX;
             LABEL BRNCH,EXIT;
             STOREFIX ← L; Q ← REAL(MODE=0)+3;
             FOR K ← 1 STEP 1 UNTIL Q DO EMITO(NOP);
             IF NUMLE
              THEN BEGIN
                  BACKFIX ← L;
                  IF FORMALV THEN CALL(DESC) END
              ELSE BACKFIX ← V + REAL(SIMPLEV)-1;

             AEXP;
COMMENT      PICK UP FIRST ARITHMETIC EXPRESSION;
             IF  ELCLASS = STEPV
                THEN BEGIN
COMMENT      HERE WE HAVE A STEP ELEMENT;
                  BACKFIX ← BUMPL;
COMMENT      LEAVE ROOM FOR FORWARD JUMP;
                  IF FORMALV THEN CALL(DESC); CALL(OPDC);
COMMENT      FETCH INDEX;
                  IF I > 70 THEN BEGIN NXTELBT ← 1; I ← 0 END
                     ELSE REGO ← I;
                  IF SIMPLEB ← SIMPLE(CONSTANB,B,SIGNB) AND
                   (ELCLASS = UNTILV OR ELCLASS = WHILEV)
                     THEN BEGIN
COMMENT      WE HAVE A SIMPLE STEP FUNCTION;
                       PLUG(CONSTANB ,B);
                     END ELSE BEGIN
COMMENT      THE STEP FUNCTION IS NOT SIMPLE:  WE CONSTRUCT A
                  SUBROUTINE;
                       I ← IF I < 4 THEN 0 ELSE REGO; STEPIT;
                       SIGNB ← CONSTANB ← FALSE;
                       EMIT(0); B ← L;
                       AEXP; EMITO(XCH);
                       BRET ← L;
                       EMITO(BFW) END;
```

```
08070000 T  0069:0070:0
08071000 T  0069:0074:0
08072000 T  0069:0074:0
08073000 T  0069:0074:0
08074000 T  0069:0074:0
08075000 T  0069:0075:2
08076000 T  0069:0078:0
08077000 T  0069:0079:2
08078000 T  0069:0080:1
08079000 T  0069:0083:2
08080000 T  0069:0084:1
08081000 T  0069:0084:1
08082000 T  0069:0084:1
START OF SEGMENT **********   70
08083000 T  0070:0000:0
08084000 T  0070:0000:0
08085000 T  0070:0000:0
08086000 T  0070:0000:0
08087000 T  0070:0002:0
08088000 T  0070:0002:1
08089000 T  0070:0003:3
08090000 T  0070:0005:3
08091000 T  0070:0006:1
08092000 T  0070:0006:1
08093000 T  0070:0006:1
08094000 T  0070:0009:3
08095000 T  0070:0014:0
08096000 T  0070:0014:0
08097000 T  0070:0014:1
08098000 T  0070:0015:3
08099000 T  0070:0017:2
08100000 T  0070:0019:2
08101000 T  0070:0019:2
08102000 T  0070:0019:3
08103000 T  0070:0019:3
08104000 T  0070:0020:0
08105000 T  0070:0021:2
08106000 T  0070:0021:2
08107000 T  0070:0022:1
08108000 T  0070:0022:1
08109000 T  0070:0025:2
08110000 T  0070:0025:2
08111000 T  0070:0027:3
08112000 T  0070:0029:2
08113000 T  0070:0030:0
08114000 T  0070:0031:2
08115000 T  0070:0033:2
08116000 T  0070:0033:2
08117000 T  0070:0034:0
08118000 T  0070:0034:1
08119000 T  0070:0034:1
08120000 T  0070:0034:1
08121000 T  0070:0038:0
08122000 T  0070:0039:2
08123000 T  0070:0040:1
08124000 T  0070:0042:0
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  0070:0060:0
                            IF ELCLASS ≠ COMMA THEN BEGIN                           08145000 T  0070:0060:0
                               EMITB(BFW,BUMPL,L+2); BACKFIX ← L END                08146000 T  0070:0061:3
                            ELSE BACKFIX ← L + 2;                                   08147000 T  0070:0065:2
                           L ← L+1; EMIT(BFW); GO TO BRNCH END END;                 08148000 T  0070:0066:1
COMMENT    THIS IS THE COMMON POINT;                                                08149000 T  0070:0069:2
                        IF ELCLASS = COMMA THEN EMITLNG; L ← L+1;                   08150000 T  0070:0069:2
                        EMIT(BFC);                                                  08151000 T  0070:0072:0
BRNCH:                  FORWARDBRANCH ← L; DIALA ← DIALB ← 0;                       08152000 T  0070:0073:2
                        IF ELCLASS = COMMA                                          08153000 T  0070:0075:2
                          THEN BEGIN                                                08154000 T  0070:0075:2
                             STEPIT;                                                08155000 T  0070:0076:0
                             FORLIST(TRUE);                                         08156000 T  0070:0076:1
                             FIX(STOREFIX,BACKFIX,FORWARDBRANCH,STMTSTART) END      08157000 T  0070:0077:3
                          ELSE BEGIN                                                08158000 T  0070:0079:2
                             IF ELCLASS ≠ DOV                                       08159000 T  0070:0079:3
                                THEN BEGIN ERR(154); REGO←L; GO EXIT END;           08160000 T  0070:0079:3
                             STEPIT;                                                08161000 T  0070:0082:1
                             IF NUMLE THEN FOOT := GETSPACE(FALSE,-1); % TEMP.      08162000 T  0070:0083:2
                        STMT;                                                       08163000 T  0070:0085:3
                                                                                    08164000 T  0070:0086:0
                             IF NUMLE THEN BEGIN                                    08165000 T  0070:0086:0
                               EMITV(RETURNSTORE ← FOOT); EMITO(BBW) END            08166000 T  0070:0087:2
                             ELSE BEGIN                                             08167000 T  0070:0089:2
                               EMITB(BBW,BUMPL,BACKFIX); RETURNSTORE ← 0 END;       08168000 T  0070:0089:3
                             STMTSTART ← FORWARDBRANCH; B ← L;                      08169000 T  0070:0092:1
                             CONSTANTCLEAN; REGO ← L;                               08170000 T  0070:0094:0
                             FIX(STOREFIX,BACKFIX,FORWARDBRANCH,L) END;             08171000 T  0070:0095:2
EXIT:                   END FORLIST;                                                08172000 T  0070:0096:1
                                                                          70 IS  100 LONG, NEXT SEG   69
REAL T1,T2,T3,T4;                                                                   08173000 T  0069:0084:1
                        NXTELBT ← 1; I ← 0;                                         08174000 T  0069:0084:1
                        STEPIT;                                                     08175000 T  0069:0086:1
                        IF SIMPI(VRET←ELBAT[I])                                     08176000 T  0069:0087:2
                          THEN BEGIN                                                08177000 T  0069:0087:3
                             IF STEPI ≠ ASSIGNOP THEN BEGIN ERR(152); GO EXIT END;  08178000 T  0069:0089:2
                             XMARK(ASSIGNREF); % FOR STATEMENT                %110- 08178100 C  0069:0091:3
                             T1 ← L; IF FORMALV THEN EMITN(ADDRES);                 08179000 T  0069:0096:0
                             K ← 0;                                                 08180000 T  0069:0098:0
```

```
        IF SIMPLE(CONSTANA,A,SIGNA) THEN
        IF ELCLASS = STEPV THEN
        IF SIMPLE(CONSTANB,B,SIGNB) THEN
        IF ELCLASS = UNTILV THEN
        IF SIMPLE(CONSTANC,Q,SIGNC) THEN
        IF ELCLASS = DOV THEN
          BEGIN
             PLUG(CONSTANA,A);
             IF SIGNA THEN EMITO(CHS);
             RETURNSTORE  ← BUMPL; ADJUST; CONSTANTCLEAN;
             STMTSTART ← L;
STEPIT;
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 ← REGO;
EXIT:  K ← 0 END FORSTMT;

REAL PROCEDURE REED;
    BEGIN
    LABEL EOF; INTEGER I,J,K;
```

```
08181000 T  0069:0099:2
08182000 T  0069:0100:0
08183000 T  0069:0101:3
08184000 T  0069:0103:2
08185000 T  0069:0104:1
08186000 T  0069:0106:0
08187000 T  0069:0107:3
08188000 T  0069:0108:0
08189000 T  0069:0109:2
08190000 T  0069:0110:1
08191000 T  0069:0113:2
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
08999000 T  0003:0505:3
08999025 T  0003:0505:3
08999050 T  0003:0505:3
START OF SEGMENT **********    71
```

```
        STREAM PROCEDURE MOVE(N,F,T); VALUE N,T;
             BEGIN  SI:=F; DI:=T; DS:=N WDS  END MOVE;
        J:=-1;
        READ(CODISK[NO])[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;

        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;
            BEGIN
    STREAM PROCEDURE MDESC(WD,TOLOC);VALUE WD;

      BEGIN DI←LOC WD; DS←  SET;SI← LOC WD; DI←TOLOC;DS←WDS END;
                DEFINE STARTINTRSC=426#;
                LABEL L1;
                LISTOG:=LISTER:=BOOLEAN(1-ERRORCOUNT,[46:1]);
    COMMENT  LISTOG IS NOT SET BY DEFAULT ON TIMESHARING;
                NOHEADING := TRUE;
                ERRORCOUNT := 0;
                ERRMAX:=999; % MAY BE CHANGED IN DOLLARCARD.
                BASENUM:=10000; ADDVALUE:=1000; NEWBASE:=TRUE;
    COMMENT  DEFAULT VALUES FOR "$SEQ" OPTION;
                LASTUSED := 4; % FOR INITIALIZATION.
                NEXTINFO ← LASTINFO ← LASTSEQROW×256+LASTSEQUENCE+1;
                PUTNBUMP(0);
                GT1 ← -"        ";
                MDESC(GT1,INFO[LASTSEQROW,LASTSEQUENCE]);
                BLANKET(0,INFO[LASTSEQROW,LASTSEQUENCE]); % FOR "$ CHECK".
                READACARD; % INITIALIZATION OF NCR,FCR, AND LCR, AND
```

```
        08999075 T  0071:0000:0
        08999100 T  0071:0000:0
        08999125 T  0071:0001:2
        08999150 T  0071:0003:2
        08999175 T  0071:0008:0
        08999200 T  0071:0014:0
        08999225 T  0071:0019:2
        08999250 T  0071:0020:1
        08999275 T  0071:0020:1
        08999300 T  0071:0027:2
        08999325 T  0071:0031:2
        08999350 T  0071:0031:3
        08999375 T  0071:0036:0
        08999400 T  0071:0041:2
        08999425 T  0071:0042:0
  71 IS    47 LONG, NEXT SEG     3
        08999450 T  0003:0505:3
        08999475 T  0003:0505:3
        08999500 T  0003:0505:3
START OF SEGMENT *********   72
        08999525 T  0072:0000:0
        08999550 T  0072:0001:3
        08999575 T  0072:0005:3
        08999600 T  0072:0012:1
        08999625 T  0072:0016:1
        08999650 T  0072:0017:2
        08999675 T  0072:0019:3
        08999700 T  0072:0019:3
        08999725 T  0072:0026:1
        08999750 T  0072:0030:1
        08999775 T  0072:0031:3
        08999800 T  0072:0032:0
  72 IS    35 LONG, NEXT SEG     3
        09000000 T  0003:0505:3
        09001000 T  0003:0505:3
        09002000 T  0003:0505:3
        09003000 T  0003:0505:3
        09004000 T  0003:0505:3
        09005000 T  0003:0505:3
START OF SEGMENT *********   73
        09006000 T  0073:0000:0
        09024000 T  0073:0001:3
        09025000 T  0073:0001:3
        09028000 T  0073:0001:3
        09028010 T  0073:0005:2
        09028050 T  0073:0005:2
        09028900 T  0073:0006:0
        09028910 T  0073:0006:1
        09028920 T  0073:0007:3
        09028930 T  0073:0009:3
        09029000 T  0073:0009:3
        09033000 T  0073:0010:1
        09034000 T  0073:0013:2
        09034100 T  0073:0014:0
        09034200 T  0073:0015:2
        09034500 T  0073:0017:2
        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
      OCT1771110463422054,    OCT1761332600326467,    OCT1751621340414205,                      09042000 T  0073:0023:3
                                                                                  START OF SEGMENT **********   74
      OCT1742165630517247,    OCT1732623176643120,    OCT1723370036413744,                      09043000 T  0073:0024:1
      OCT1714266046116735,    OCT1705343457542525,    OCT1676634373473252,                      09044000 T  0073:0024:1
      OCT1651040347241213,    OCT1641250441111455,    OCT1631522551333770,                      09045000 T  0073:0024:1
      OCT1622047303622767,    OCT1612461164567564,    OCT1603175421725521,                      09046000 T  0073:0024:1
      OCT1574034726313046,    OCT1565044113775657,    OCT1556255136775233,                      09047000 T  0073:0024:1
      OCT1547730366574502,    OCT1521171646433362,    OCT1511430220142257,                      09048000 T  0073:0024:1
      OCT1501736264172732,    OCT1472325741231521,    OCT1463013331500045,                      09049000 T  0073:0024:1
      OCT1453616220020057,    OCT1444561664024072,    OCT1435716241031111,                      09050000 T  0073:0024:1
      OCT1427301711237333,    OCT1401116227350722,    OCT1371341675243107,                      09051000 T  0073:0024:1
      OCT1361632254513731,    OCT1352200727636717,    OCT1342641115606502,                      09052000 T  0073:0024:1
      OCT1333411341150223,    OCT1324313631402270,    OCT1315376577702746,                      09053000 T  0073:0024:1
      OCT1306676337663537,    OCT1261045602764047,    OCT1251257143561061,                      09054000 T  0073:0024:1
      OCT1241532774515275,    OCT1232061573640554,    OCT1222476132610706,                      09055000 T  0073:0024:1
      OCT1213215561353071,    OCT1204061115645707,    OCT1175075341217270,                      09056000 T  0073:0024:1
      OCT1166314631463146,    OCT1141000000000000,    OCT1131200000000000,                      09057000 T  0073:0024:1
      OCT1121440000000000,    OCT1111750000000000,    OCT1102342000000000,                      09058000 T  0073:0024:1
      OCT1073032400000000,    OCT1063641100000000,    OCT1054611320000000,                      09059000 T  0073:0024:1
      OCT1045753604000000,    OCT1037346545000000,    OCT1011124027620000,                      09060000 T  0073:0024:1
      OCT0001351035564000,    OCT0011643245121000,    OCT0022214116345200,                      09061000 T  0073:0024:1
      OCT0032657142036440,    OCT0043432772446150,    OCT0054341571157602,                      09062000 T  0073:0024:1
      OCT0065432127413543,    OCT0076740555316473,    OCT0111053071060221,                      09063000 T  0073:0024:1
     OCT0121265707274266,    OCT0131543271153343,    OCT0142074147406234,                       09064000 T  0073:0024:1
     OCT0152513201307703,    OCT0163236041571663,    OCT0174105452130240,                       09065000 T  0073:0024:1
     OCT0205126764556310,    OCT0216354561711772,    OCT0231004771627437,                       09066000 T  0073:0024:1
     OCT0241206170175347,    OCT0251447626234641,    OCT0261761573704011,                       09067000 T  0073:0024:1
     OCT0272356132665013,    OCT0303051561442216,    OCT0313664115752661,                       09068000 T  0073:0024:1
     OCT0324641141345435,    OCT0336011371636745,    OCT0347413670206536,                       09069000 T  0073:0024:1
     OCT0361131664625027,    OCT0371360241772234,    OCT0401654312370703,                       09070000 T  0073:0024:1
     OCT0412227375067064,    OCT0422675274304701,    OCT0433454553366062,                       09071000 T  0073:0024:1
     OCT0444367706263476,    OCT0455465667740415,    OCT0467003245730521,                       09072000 T  0073:0024:1
     OCT0501060411731665,    OCT0511274514320242,    OCT0521553637404312,                       09073000 T  0073:0024:1
     OCT0532106607305375,    OCT0542530351166674,    OCT0553256443424453,                       09074000 T  0073:0024:1
     OCT0564132154331566,    OCT0575160607420123,    OCT0606414751324150,                       09075000 T  0073:0024:1
     OCT0621012014361120,    OCT0631214417455344,    OCT0641457523370635,                       09076000 T  0073:0024:1
     OCT0651773450267005,    OCT0662372362344606,    OCT0673071057035747,                       09077000 T  0073:0024:1
     OCT0703707272645341,    OCT0714671151416632,    OCT0726047403722400,                       09078000 T  0073:0024:1
     OCT0737461304707100,    OCT0751137556607072,    OCT0761367512350710,                       09079000 T  0073:0024:1
     OCT0771665435043072;                                                                       09080000 T  0073:0024:1
                                                                                  74 IS  115 LONG, NEXT SEG   73
COMMENT  THIS IS THE FILL FOR THE SECOND ROW OF INFO:                                           09081000 T  0073:0024:1
                    THE FIRST ITEMS ARE STREAM RESERVED WORDS,                                  09082000 T  0073:0024:1
                    THEN ORDINARY RESERVED WORDS,                                               09083000 T  0073:0024:1
                    THEN INTRINSIC FUNCTIONS;                                                   09084000 T  0073:0024:1
 FILL INFO[1,*] WITH                                                                            09085000 T  0073:0024:1
     OCT0670000600000002, "?SI000",                          %256                               09086000 T  0073:0025:3
                                                                                  START OF SEGMENT **********   75
     OCT0700001040000002, "?DI000",                          %258                               09087000 T  0073:0026:1
     OCT0710001460000002, "?CI000",                          %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, "3L0C00",                          %276    09096000 T  0073:0026:1
0CT1030001170000002, "2DC000",                          %278    09097000 T  0073:0026:1
0CT1040001430000002, "5L0CAL",                          %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, "30CT00",                          %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, "7BOOLE", "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, "4HUNT0",                          %348    09113700 T  0073:0026:1
0CT0630000000000002, "2IF000",                          %350    09113800 T  0073:0026:1
0CT0500000040000002, "4REAL0",                          %352    09113900 T  0073:0026:1
0CT0500000050000003, "7INTEG", "ER000000",              %354    09114000 T  0073:0026:1
0CT0500000070000002, "5LABEL",                          %357    09114100 T  0073:0026:1
0CT0360002000020003, "6MEMOR", "Y       ",              %359    09114200 T  0073:0026:1
0CT1410456000000002, "3MOD00",                          %362    09114300 T  0073:0026:1
0CT0500000140000003, "7MONIT", "OR      ",              %364    09114400 T  0073:0026:1
0CT0130301060000002, "4NARS0",                          %367    09114500 T  0073:0026:1
0CT0500000200000002, "4NAME0",                          %369    09114600 T  0073:0026:1
0CT0130304030000002, "5NFLAG",                          %371    09114700 T  0073:0026:1
0CT1320300230000002, "3NOT00",                          %373    09114800 T  0073:0026:1
0CT1350440430000002, "2OR000",                          %375    09114900 T  0073:0026:1
0CT0500000020000002, "4SAVE0",                          %377    09115000 T  0073:0026:1
0CT0500000010000002, "3OWN00",                          %379    09115100 T  0073:0026:1
0CT0460000000000003, "6POLIS", "H       ",              %381    09115200 T  0073:0026:1
0CT0500000160000003, "9PROCE", "DURE    ",              %384    09115300 T  0073:0026:1
```

TOGGLE
FLAG
HUNT
MONITOR
NFLAG
SIGN
SCOPON
SCOPOFF

```
0CT0130300000160011, "4SIGNO",                          %387    09115400 T  0073:0026:1
          0CT2025,   COMMENT  DUP   ;                            09115500 T  0073:0026:1
          0CT0000,   COMMENT  LITC 0;                            09115600 T  0073:0026:1
          0CT0425,   COMMENT  NEQ   ;                            09115700 T  0073:0026:1
          0CT1025,   COMMENT  XCH   ;                            09115800 T  0073:0026:1
          0CT0155,   COMMENT  DIA  1;                            09115900 T  0073:0026:1
          0CT0161,   COMMENT  DIB  1;                            09116000 T  0073:0026:1
          0CT0165,   COMMENT  TRB  1;                            09116100 T  0073:0026:1
0CT1110000000000002, "4STEPO",                          %396    09116200 T  0073:0026:1
0CT0500000220000003, "6STREA", "M      ",               %398    09116300 T  0073:0026:1
0CT0500000110000003, "#SUBRO", "UTINE  ",               %401    09116400 T  0073:0026:1
0CT0500000150000003, "6SWITC", "H      ",               %404    09116500 T  0073:0026:1
0CT1120000000000002, "4THENO",                          %407    09116600 T  0073:0026:1
0CT1130000000000002, "2T0000",                          %409    09116700 T  0073:0026:1
0CT0410000010000002, "4TRUEO",                          %411    09116800 T  0073:0026:1
0CT0560000000000002, "5UNTIL",                          %413    09116900 T  0073:0026:1
0CT1140000000000002, "5VALUE",                          %415    09117000 T  0073:0026:1
0CT0540000000000002, "5WHILE",                          %417    09117100 T  0073:0026:1
0CT1310440200000002, "3ADD00",                          %419    09117200 T  0073:0026:1
0CT1310240270000002, "3BRT00",                          %421    09117300 T  0073:0026:1
0CT1310453050000002, "3CCX00",                          %423    09117400 T  0073:0026:1
0CT1310442500000002, "3CDC00",                          %425    09117500 T  0073:0026:1
0CT1310457050000002, "3CFX00",                          %427    09117600 T  0073:0026:1
0CT1310302060000002, "3CHS00",                          %429    09117700 T  0073:0026:1
0CT1310440500000002, "3C0C00",                          %431    09117800 T  0073:0026:1
0CT1310242020000002, "3COM00",                          %433    09117900 T  0073:0026:1
0CT1310302060000002, "3CSB00",                          %435    09118000 T  0073:0026:1
0CT1310240120000002, "3DEL00",                          %437    09118100 T  0073:0026:1
0CT1260100550000002, "3DIA00",                          %439    09118200 T  0073:0026:1
0CT1260100610000002, "3DIB00",                          %441    09118300 T  0073:0026:1
0CT1310344050000002, "3DUP00",                          %443    09118400 T  0073:0026:1
0CT1310451050000002, "3EQL00",                          %445    09118500 T  0073:0026:1
0CT1310443050000002, "3FCX00",                          %447    09118600 T  0073:0026:1
0CT1310447050000002, "3FFX00",                          %449    09118700 T  0073:0026:1
0CT1310440250000002, "3GEQ00",                          %451    09118800 T  0073:0026:1
0CT1310440450000002, "3GTR00",                          %453    09118900 T  0073:0026:1
0CT1310104420000002, "3HLB00",                          %455    09119000 T  0073:0026:1
0CT1310104420000002, "3HP200",                          %457    09119050 T  0073:0026:1
0CT1310446000000002, "3IDV00",                          %459    09119100 T  0073:0026:1
0CT1310251020000002, "3II000",                          %461    09119200 T  0073:0026:1
0CT1310250220000002, "3INA00",                          %463    09119300 T  0073:0026:1
0CT1310250420000002, "3INB00",                          %465    09119400 T  0073:0026:1
0CT1310100420000002, "3INT00",                          %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, "3ITI00",                          %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",
                0CT1310450450000002, "3LSS00",
                0CT1310101100000002, "3MKS00",
                0CT1310441000000002, "3MUL00",
                0CT1310441050000002, "3NEQ00",
                0CT1310100130000002, "3NOP00",
                0CT0650006550000003, "6SCOP0", "N.......";

            FILL INFO[2,*] WITH
                0CT1310300000020004, "3RDF00",

                          0CT0000,   COMMENT  LITC 0;
                          0CT2141,   COMMENT  FXS   ;
                0CT1310300000020004, "3RDS00",
                          0CT0004,   COMMENT  LITC 1;
                          0CT2141,   COMMENT  FXS   ;
                0CT1310456000000002, "3RDV00",
                0CT1310304030000002, "3RFB00",
                0CT1310240470000002, "3RNO00",
                0CT1310145060000002, "3RRR00",
                0CT1310311060000002, "3RSB00",
                0CT1310242470000002, "3RSP00",
                0CT1310141020000002, "3RTM00",
                0CT1310240470000002, "3RTN00",
                0CT1310141020000002, "3RTR00",
                0CT1310242470000002, "3RTS00",
                0CT1310310030000002, "3SFB00",
                0CT1310442040000002, "3SND00",
                0CT1310301060000002, "3SSB00",
                0CT1310316060000002, "3SSF00",
                0CT1310301060000002, "3SSN00",
                0CT1310311060000002, "3SSP00",
                0CT1310401040000002, "3STD00",
                0CT1310240000020004, "3STF00",
                          0CT0010,   COMMENT  LITC 2;
                          0CT2141,   COMMENT  FXS   ;
                0CT1310442040000002, "3STN00",
                0CT1310240000020004, "3STS00",
                          0CT0014,   COMMENT LITC 3;
                          0CT2141,   COMMENT FXS   ;
                0CT1310440600000002, "3SUB00",
                0CT1310344060000002, "3TFB00",
                0CT1270440650000002, "3TFR00",
                0CT1310155060000002, "3TI000",
                0CT1310344060000002, "3TOP00",
                0CT1270440650000002, "3TRB00",
                0CT1300300000000002, "3VFI00",
                0CT1310502050000002, "3XCH00",
                0CT1310101070000002, "3XIT00",
                0CT1310105020000002, "3ZIP00",
                0CT1310105020000002, "3ZP100",
                0CT1270500750000002, "3CFF00",
                0CT1270500750000002, "3FCF00",
                0CT1270500710000002, "3CFL00",
                0CT1270500710000002, "3FCL00",
                0CT1310440210000002, "3DLA00",
                0CT1310440210000002, "3ADL00",
```

```
%497            09121000 T  0073:0026:1
%499            09121100 T  0073:0026:1
%501            09121200 T  0073:0026:1
%503            09121300 T  0073:0026:1
%505            09121400 T  0073:0026:1
%507            09121500 T  0073:0026:1
%509            09121600 T  0073:0026:1
                      75 IS  256 LONG, NEXT SEG   73
                09121650 T  0073:0026:1
%512            09121700 T  0073:0027:3
                    START OF SEGMENT **********   76
                09121800 T  0073:0028:1
                09121900 T  0073:0028:1
%516            09122000 T  0073:0028:1
                09122100 T  0073:0028:1
                09122200 T  0073:0028:1
%520            09122300 T  0073:0028:1
%522            09122400 T  0073:0028:1
%524            09122500 T  0073:0028:1
%526            09122600 T  0073:0028:1
%528            09122700 T  0073:0028:1
%530            09122800 T  0073:0028:1
%532            09122900 T  0073:0028:1
%534            09123000 T  0073:0028:1
%536            09123100 T  0073:0028:1
%538            09123200 T  0073:0028:1
%540            09123300 T  0073:0028:1
%542            09123400 T  0073:0028:1
%544            09123500 T  0073:0028:1
%546            09123600 T  0073:0028:1
%548            09123700 T  0073:0028:1
%550            09123800 T  0073:0028:1
%552            09123900 T  0073:0028:1
%554            09124000 T  0073:0028:1
                09124100 T  0073:0028:1
                09124200 T  0073:0028:1
%558            09124300 T  0073:0028:1
%560            09124400 T  0073:0028:1
                09124500 T  0073:0028:1
                09124600 T  0073:0028:1
%564            09124700 T  0073:0028:1
%566            09124800 T  0073:0028:1
%568            09124900 T  0073:0028:1
%570            09125000 T  0073:0028:1
%572            09125050 T  0073:0028:1
%574            09125100 T  0073:0028:1
%576            09125200 T  0073:0028:1
%578            09125300 T  0073:0028:1
%580            09125400 T  0073:0028:1
%582            09125500 T  0073:0028:1
%584            09125600 T  0073:0028:1
%586            09125700 T  0073:0028:1
%588            09125800 T  0073:0028:1
%590            09125900 T  0073:0028:1
%592            09126000 T  0073:0028:1
%594            09126100 T  0073:0028:1
%596            09126200 T  0073:0028:1
```

```
0CT1310440610000002, "3DLS00",                                  %598              09126300 T  0073:0028:1
0CT1310440610000002, "3SDL00",                                  %600              09126400 T  0073:0028:1
0CT1310441010000002, "3DLM00",                                  %602              09126500 T  0073:0028:1
0CT1310441010000002, "3MDL00",                                  %604              09126600 T  0073:0028:1
0CT1310442010000002, "3DLD00",                                  %606              09126700 T  0073:0028:1
0CT1310442010000002, "3DDL00",                                  %608              09126800 T  0073:0028:1
0CT0460000000000002, "1P0000",                                  %610              09126900 T  0073:0028:1
0CT0360002000020002, "1M0000",                                  %612              09127000 T  0073:0028:1
0CT1310240000020004, "3PRL00",                                  %614              09127100 T  0073:0028:1
        0CT0111,   COMMENT  PRL;                                                  09127200 T  0073:0028:1
        0CT0055,   COMMENT  NOP;                                                  09127300 T  0073:0028:1
0CT0650006610000003, "7SC0P0", "FF......",                      %618              09127400 T  0073:0028:1
0CT0030000000040003, "2LB000", "[#      ",                      %621              09127500 T  0073:0028:1
0CT0030000000040003, "2RB000", "]#      ",                      %624              09127600 T  0073:0028:1
0CT0030000000040003, "3GTR00", ">#      ",                      %627              09127700 T  0073:0028:1
0CT0030000000040003, "3GEQ00", "≥#      ",                      %630              09127800 T  0073:0028:1
0CT0030000000040003, "3EQL00", "=#      ",                      %633              09127900 T  0073:0028:1
0CT0030000000040003, "3NEQ00", "≠#      ",                      %636              09128000 T  0073:0028:1
0CT0030000000040003, "3LEQ00", "≤#      ",                      %639              09128100 T  0073:0028:1
0CT0030000000040003, "3LSS00", "<#      ",                      %642              09128200 T  0073:0028:1
0CT0030000000040003, "5TIMES", "×#      ",                      %645              09128300 T  0073:0028:1
0CT1310117530000002, "3SC100",                                  %648              09128400 T  0073:0028:1
0CT1310117540000002, "3SAN00",                                  %650              09128500 T  0073:0028:1
0CT1310157730000002, "3SCS00",                                   %652 %112-       09128600 P  0073:0028:1
0CT0500000250000002, "5FIFLD",                                   %654 %115-       09128700 P  0073:0028:1
0CT0610000000000000, "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
                                                                                  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
```

```
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  /;
 OCT1170000000000000,  COMMENT  ,;   OCT0000000000020000,  COMMENT  %;
 OCT1370441050002561,  COMMENT  ≠;   OCT1370451050002460,  COMMENT  =;
 OCT1230000000000000,  COMMENT  ];   OCT0000000000140000,  COMMENT  ";
  0,0;

  FILL MACRO[*] WITH
       OCT0131,          COMMENT SFS A       00          ;

       OCT0116,          COMMENT SFD A       01          ;
       OCT0000,          COMMENT SYNTAX ERROR02          ;
       OCT0140,          COMMENT INC A       03          ;
       OCT0130,          COMMENT SRS A       04          ;
       OCT0117,          COMMENT SRD A       05          ;
       OCT0000,          COMMENT SYNTAX ERROR06          ;
       OCT0000,          COMMENT SYNTAX ERROR07          ;
       OCT00310143,      COMMENT CRF A, SFS 008          ;
       OCT00160143,      COMMENT CRF A, SFD 009          ;
```

```
                                                    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
                                                    09211000 T  0073:0054:1
                                                    09212000 T  0073:0054:1
                                                    09213000 T  0073:0054:1
                                                    09214000 T  0073:0054:1
                                         77 IS   32 LONG, NEXT SEG   73
                                                    09215000 T  0073:0054:1
                                                    09216000 T  0073:0055:2
                                       START OF SEGMENT **********  78
                                                    09217000 T  0073:0056:1
                                                    09218000 T  0073:0056:1
                                                    09219000 T  0073:0056:1
                                                    09220000 T  0073:0056:1
                                                    09221000 T  0073:0056:1
                                                    09222000 T  0073:0056:1
                                                    09223000 T  0073:0056:1
                                                    09224000 T  0073:0056:1
                                                    09225000 T  0073:0056:1
```

```
                0CT00470143,        COMMENT CRF A,  JFW 0  10          ;                09226000 T  0073:0056:1
                0CT00400143,    COMMENT CRF A, INC 011                 ;                09227000 T  0073:0056:1
                0CT00300143,    COMMENT CRF A, SRS 012                 ;                09228000 T  0073:0056:1
                0CT00170143,    COMMENT CRF A, SRD 013                 ;                09229000 T  0073:0056:1
                0CT0000,        COMMENT SYNTAX ERROR14                 ;                09230000 T  0073:0056:1
                0CT0000,        COMMENT SYNTAX ERROR15                 ;                09231000 T  0073:0056:1
                0CT0153,        COMMENT RSA A       16                 ;                09232000 T  0073:0056:1
                0CT0104,        COMMENT RDA A       17                 ;                09233000 T  0073:0056:1
                0CT0150,        COMMENT RCA A       18                 ;                09234000 T  0073:0056:1
          0CT004201430042,      COMMENT SEC 0, CRF A, SEC 0 19         ;                09235000 T  0073:0056:1
                0CT0122,        COMMENT SES A       20                 ;                09236000 T  0073:0056:1
                0CT0106,        COMMENT SED A       21                 ;                09237000 T  0073:0056:1
                0CT0000,        COMMENT SYNTAX ERROR22                 ;                09238000 T  0073:0056:1
                0CT0000,        COMMENT SYNTAX ERROR23                 ;                09239000 T  0073:0056:1
                0CT0056,        COMMENT TSA 0       24                 ;                09240000 T  0073:0056:1
                0CT0000,        COMMENT SYNTAX ERROR25                 ;                09241000 T  0073:0056:1
                0CT0000,        COMMENT SYNTAX ERROR26                 ;                09242000 T  0073:0056:1
                0CT0000,        COMMENT SYNTAX ERROR27                 ;                09243000 T  0073:0056:1
                0CT0000,        COMMENT SYNTAX ERROR28                 ;                09244000 T  0073:0056:1
                0CT0007,        COMMENT TDA 0       29                 ;                09245000 T  0073:0056:1
                0CT0000,        COMMENT SYNTAX ERROR30                 ;                09246000 T  0073:0056:1
                0CT0000,        COMMENT SYNTAX ERROR31                 ;                09247000 T  0073:0056:1
                0CT0115,        COMMENT SSA A       32                 ;                09248000 T  0073:0056:1
                0CT0114,        COMMENT SDA A       33                 ;                09249000 T  0073:0056:1
                0CT0154,        COMMENT SCA A       34                 ;                09250000 T  0073:0056:1
                0CT0141;        COMMENT STC A       35                 ;                09251000 T  0073:0056:1
                                                                                  78 IS   36 LONG, NEXT SEG   73
                                                                                       09251208 T  0073:0056:1
                                                                                START OF SEGMENT **********   79
                                                                                       09251212 T  0073:0058:0
                                                                                       09251214 T  0073:0058:0
                                                                                       09251216 T  0073:0058:0
                                                                                       09251218 T  0073:0058:0
                                                                                       09251220 T  0073:0058:0
                                                                                       09251224 T  0073:0058:0
                                                                                       09251228 T  0073:0058:0
                                                                                       09251230 T  0073:0058:0
                                                                                       09251232 T  0073:0058:0
                                                                                       09251234 T  0073:0058:0
                                                                                       09251236 T  0073:0058:0
                                                                                       09251240 T  0073:0058:0
                                                                                       09251244 T  0073:0058:0
                                                                                       09251248 T  0073:0058:0
                                                                                       09251252 T  0073:0058:0
                                                                                       09251256 T  0073:0058:0
                                                                                       09251260 T  0073:0058:0
                                                                                       09251264 T  0073:0058:0
                                                                                       09251268 T  0073:0058:0
                                                                                       09251272 T  0073:0058:0
                                                                                       09251276 T  0073:0058:0
                                                                                       09251278 T  0073:0058:0
                                                                                       09251280 T  0073:0058:0
                                                                                       09251284 T  0073:0058:0
                                                                                %108-  09251285 C  0073:0058:0
                                                                                %108-  09251286 C  0073:0058:0
                                                                                       09251288 T  0073:0058:0
                                                                                  79 IS   55 LONG, NEXT SEG   73
    FILL OPTIONS[*] WITH "5CHECK",0,        % 0,1
                         "6DEBUG",0,        % 2,3
                         "4DECKO",0,        % 4,5
                         "6FORMA",0,        % 6,7
                         "9INTRI",0,        % 8,9
                         "5LISTA",0,        % 10,11
                         "4LISTO",0,        % 12,13
                         "5LISTP",0,        % 14,15
                         "3MCPOO",0,        % 16,17
                         "4TAPEO",0,        % 18,19
                         "4NESTO",0,        % 20,21
                         "3NEWOO",0,        % 22,23
                         "7NEWIN",0,        % 24,25
                         "40MITO",0,        % 26,27
                         "1$0000",0,        % 28,29
                         "3PRTOO",0,        % 30,31
                         "5PUNCH",0,        % 32,33
                         "5PURGE",0,        % 34,35
                         "4SEGSO",0,        % 36,37
                         "3SEQOO",0,        % 38,39
                         "6SEQER",0,        % 40,41
                         "6SINGL",0,        % 42,43
                         "5STUFF",0,        % 44,45
                         "4VOIDO",0,        % 46,47
                         "5VOIDT",0,        % 48,49
                         "4BENDO",0,        % 50, 51
                         "4XREFO",0,        % 52, 53
    0;
```

```
                    DO UNTIL STEPI = 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
                                                                                    START OF SEGMENT **********   80
      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
                    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
            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,T:=(L+3)DIV 4,T:=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
                                                                                    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
                                                                                    START OF SEGMENT **********   81
    INFO[0:200,0:255];  % FOR LARGE MCP"S.                                                      09392500 T  0081:0003:2
                        INTEGER SAVNDX,NONSAVNDX,N;                                             09393000 T  0081:0005:3
    INTEGER Q,J,K,M;                                                                            09393010 T  0081:0005:3
```

```
BOOLEAN TSSTOG; REAL T;
REAL PROCEDURE PUSHER(GRINCH,GOT,XMAS); VALUE XMAS; REAL XMAS;
                        ARRAY GOT[0]; ARRAY GRINCH [0,0];
     BEGIN
     REAL WHO,WHAT;

     DEFINE LINKR = [32:8]#;

%
     IF WHO:=XMAS,LINKC ≤ 225 THEN
          BEGIN
          MOVE(30,GRINCH[XMAS,LINKR,WHO],GOT[0]);
          PUSHER:=XMAS + 30;
          END
     ELSE BEGIN
          MOVE(WHAT:=256-WHO,GRINCH[XMAS,LINKR,WHO],GOT[0]);
          XMAS:=XMAS + WHAT;
          MOVE(WHO:=30-WHAT, GRINCH[XMAS,LINKR,0], GOT[WHAT]);
          PUSHER:=XMAS + WHO;
          END;
     END PUSHER;

PROCEDURE PUSHEE(GRINCH,N,B,Y); VALUE N,B,Y; REAL N,B,Y;
     ARRAY GRINCH[0,0];
     BEGIN
     REAL I,J,X;

     DEFINE LINKR = [32:8]#;
     J:=Y;
     I:=B + N;
     WHILE B < I DO
          BEGIN
          IF Y:=B,LINKC ≤ 225 THEN
          BEGIN
          MOVE(30,CODE(J),GRINCH[B,LINKR,Y]);
          J:=J + 30;
          B:=B + 30;
          END
     ELSE BEGIN
          MOVE(X:=256-Y,CODE(J),GRINCH[B,LINKR,Y]);
          B:=B + X;
          J:=J + X;
          MOVE(Y:=30-X,CODE(J),GRINCH[B,LINKR,0]);
          B:=B + Y;
          J:=J + Y;
          END;
     END;
END PUSHEE;

STREAM PROCEDURE FIXHDR(F,N); VALUE N;
     BEGIN SI←F; SI←SI+24; DI←LOC F; DS←WDS;
          SI←F; 14(SI←SI+8); DI←LOC F; DS←WDS;
          DI←F; DI←DI+38; SI← LOC N;
          SI←SI+7; DS←CHR;
     END FIXHDR;
                    LABEL EOF;
               IF NOT INTOG THEN
          BEGIN
```

```
09393020 T  0081:0005:3
09393050 T  0081:0005:3
09393060 T  0081:0005:3
09393070 T  0081:0005:3
09393080 T  0081:0005:3
START OF SEGMENT *********   82
09393090 T  0082:0000:0
09393100 T  0082:0000:0
09393110 T  0082:0000:0
09393120 T  0082:0001:3
09393130 T  0082:0002:0
09393140 T  0082:0005:2
09393150 T  0082:0006:1
09393160 T  0082:0006:1
09393170 T  0082:0007:2
09393180 T  0082:0011:2
09393190 T  0082:0012:0
09393200 T  0082:0016:0
09393220 T  0082:0017:3
09393230 T  0082:0017:3
 82 IS   20 LONG, NEXT SEG   81
09393240 T  0081:0005:3
09393250 T  0081:0005:3
09393260 T  0081:0005:3
09393270 T  0081:0005:3
START OF SEGMENT *********   83
09393280 T  0083:0000:0
09393290 T  0083:0000:0
09393300 T  0083:0000:1
09393310 T  0083:0002:0
09393320 T  0083:0003:2
09393330 T  0083:0003:2
09393340 T  0083:0005:2
09393350 T  0083:0005:3
09393360 T  0083:0011:2
09393370 T  0083:0012:0
09393380 T  0083:0013:3
09393390 T  0083:0013:3
09393400 T  0083:0014:0
09393410 T  0083:0020:1
09393420 T  0083:0021:3
09393430 T  0083:0023:2
09393440 T  0083:0029:3
09393450 T  0083:0030:1
09393460 T  0083:0032:0
09393470 T  0083:0032:0
09393480 T  0083:0032:1
83 IS   35 LONG, NEXT SEG   81
09393700 T  0081:0005:3
09393710 T  0081:0005:3
09393720 T  0081:0007:2
09393730 T  0081:0008:1
09393740 T  0081:0009:2
09393750 T  0081:0009:3
09394000 T  0081:0010:0
09394100 T  0081:0010:0
09394200 T  0081:0011:2
```

```
          L←(L+3)DIV 4; COMMENT L←NUM. OF WORDS IN OUTER BLOCK;
          FILL SAVINFO[0,*] WITH
               OCT7700000000000015,

               OCT0253010477527705,
               OCT0051000000000000,
               OCT0441070001000062;

          Q ← -1;
     PUSHEF(SAVINFO,L,4,5);
     SAVNDX:=L;
END;
          REWIND(CODISK);
 DO BEGIN IF   REED=0 THEN GO TO EOF;
           N←FETCH(MKABS(CODE(0)))-1;
           IF BOOLEAN(FETCH(MKABS(CODE(1)))) THEN
BEGIN
     PUSHEF(SAVINFO,N,SAVNDX,1);
     SAVNDX:=SAVNDX +N;
END ELSE BEGIN
     IF DECKTOG THEN
               STACKHEAD[Q←Q+1] ← 1024×NONSAVNDX+N;
          PUSHEF(INFO,N,NONSAVNDX,1);
          NONSAVNDX:=((NONSAVNDX + N + 29)DIV 30)×30;
     END;
     END UNTIL FALSE;
          END;
          END UNTIL FALSE;
EOF:       N←(SAVNDX+29) DIV 30; COMMENT NUMBER OF DISK SEGMENTS
            OCCUPIED BY SAVE PROCEDURES AND ARRAYS;
     IF INTOG AND NOT DECKTOG THEN
BEGIN    % INTRINSIC FUNCTION OPTION
     FOR J:=USEROPINX STEP 2 UNTIL OPARSIZE DO % IS TIMESHARING SET
     IF OPTIONS[J] = "@TIMES" THEN
     BEGIN TSSTOG:=BOOLEAN(OPTIONS[J+1]); J:=OPARSIZE END;
     I ← PRTBASE + 1; J ← 0;
     DO IF GT1 ← PRT[I] ≠0 THEN
     BEGIN
          J ← J+1;
           SAVINFO[J,LINKR,J,LINKC] ←
            0&GT1[8:8:10]
            &GT1[33:18:15];
     END UNTIL I:=I +1 ≥ PRTIMAX;
      SAVINFO[0,0] ← J;      % # OF INTRINSICS
     SAVNDX ← MAXINTRINSIC;
END ELSE BEGIN
               I←PRTBASE; DO IF GT1←PRT[I]≠0 THEN
          BEGIN IF GT1.[1:5]≠LDES THEN
         BEGIN IF (GT1←GT1&(GT1.[33:15]+L)[33:33:15]),[6:2]≠3 THEN
                 GT1←GT1&(GT1.[18:15]+N)[18:33:15];
          END;
               MDESC(GT1,SAVINFO[I,LINKR,I,LINKC]);
          END ELSE SAVINFO[I,LINKR,I,LINKC]:=0 UNTIL I:=I+1≥PRTIMAX;
               MDESC(0&1[2:47:1],SAVINFO[0,PRTBASE-1]);
               SAVNDX ← 30 × N;
        END;
               I ← 0; J ← -1;

               IF NOT DECKTOG THEN
```

```
09395000 T  0081:0011:3
09395100 T  0081:0013:2
09395200 T  0081:0014:0
START OF SEGMENT **********   84
09395300 T  0081:0015:2
09395400 T  0081:0015:2
09395500 T  0081:0015:2
84 IS    4 LONG, NEXT SEG   81
09395700 T  0081:0015:2
09396000 T  0081:0016:0
09397000 T  0081:0018:1
09397100 T  0081:0019:2
09398000 T  0081:0019:2
09399000 T  0081:0021:2
09400000 T  0081:0022:1
09401000 T  0081:0028:1
09402000 T  0081:0033:3
09402100 T  0081:0034:0
09403000 T  0081:0036:0
09404000 T  0081:0037:3
09405000 T  0081:0038:0
09405500 T  0081:0038:1
09406000 T  0081:0042:1
09407000 T  0081:0044:1
09408000 T  0081:0047:3
09412000 T  0081:0047:3
09413000 T  0081:0048:0
09414000 T  0081:0050:1
09414010 T  0081:0050:1
09414020 T  0081:0052:1
09414022 T  0081:0053:2
09414024 T  0081:0055:2
09414026 T  0081:0056:0
09414030 T  0081:0061:2
09414040 T  0081:0063:2
09414050 T  0081:0065:2
09414060 T  0081:0065:3
09414070 T  0081:0066:1
09414080 T  0081:0069:2
09414090 T  0081:0070:0
09414100 T  0081:0071:2
09414110 T  0081:0073:3
09414120 T  0081:0075:3
09414130 T  0081:0076:0
09415000 T  0081:0078:0
09415500 T  0081:0081:2
09416000 T  0081:0082:1
09417000 T  0081:0086:1
09417500 T  0081:0089:2
09418000 T  0081:0089:2
09419000 T  0081:0092:0
09419100 T  0081:0097:2
09420000 T  0081:0101:3
09420010 T  0081:0102:1
09420020 T  0081:0102:1
09420100 T  0081:0104:1
09421000 T  0081:0104:1
```

```
BEGIN
    DO
BEGIN
     I:=PUSHER(SAVINFO,ELBAT,I);
     J:=J + 1;
     WRITE(DISK,30,ELBAT[*]);
END UNTIL I ≥ SAVNDX;
     I:=0;
     WHILE I < NONSAVNDX DO
BEGIN
     I:=PUSHER(INFO,ELBAT,I);
     J:=J + 1;
     WRITE(DISK,30,ELBAT[*]);
END;
     N←IF INTOG THEN IF TSSTOG THEN
          TSSINTYPE ELSE DCINTYPE ELSE MCPTYPE;
     FIXHDR(DISK,N);
     LOCK(DISK,*);
END  ELSE
BEGIN ELBAT[0]←0; I←16;
      DO BEGIN MOVE(8,SAVINFO[I.LINKR,I.LINKC],ELBAT[1]);
           ELBAT[9]←B2D(I+96)&1[11:47:1]&(I+96)[23:35:1];
                WRITE(DECK,10,ELBAT[*]);
         END UNTIL I←I+8≥SAVNDX;
      FILL ELBAT[*] WITH 0,

           0CT7500000000000012,
           0CT0004535530611765,
           0CT7006000404210435,
           0CT7700000000000015,
           0CT0253010477527705,
           0CT0051000004410046,
           0CT0441070001000062,
           0CT0040413100000000,
           0CT0001000000000101;

      WRITE(DECK,10,ELBAT[*]);
     ELBAT[0] ←0&REAL(DECKTOG)[1:19:17];
     FOR I ← 0 STEP 1 UNTIL Q DO
          BEGIN K ← STACKHEAD[I].[23:15];
                M ← STACKHEAD[I].[38:10];
                FOR J ← 0 STEP 8 UNTIL M DO BEGIN
                     MOVE(8,INFO[(J+K).LINKR,(J+K).LINKC],
                          ELBAT [1]);
                     ELBAT[9] ← B2D(J)&"310"[1:31:17];
                     WRITE(DECK,10,ELBAT[*]) END;
          END;
END END END PROGRAM;


COMMENT THIS SECTION CONTAINS GENERATORS USED BY THE BLOCK ROUTINE;
PROCEDURE DEFINEGEN(MACRO,J); VALUE MACRO,J; BOOLEAN MACRO; REAL J;
     BEGIN
        OWN INTEGER CHARCOUNT, REMCOUNT;

  COMMENT CHARCOUNT CONTAINS NUMBER OFCHARACTORS OF THE DEFINE THAT WE
          HAVE PUT INTO INFO.  REMCOUNT CONTAINS NUMBER OF CHARACT-
```

```
09421500 T  0081:0105:3
09422000 T  0081:0106:0
09423000 T  0081:0106:0
09424000 T  0081:0106:0
09425000 T  0081:0109:2
09425900 T  0081:0110:0
09426000 T  0081:0114:1
09427000 T  0081:0115:3
09427100 T  0081:0116:1
09427200 T  0081:0118:0
09427500 T  0081:0118:0
09428000 T  0081:0121:2
09429000 T  0081:0122:1
09430000 T  0081:0126:1
09430050 T  0081:0127:2
09430060 T  0081:0128:1
09430075 T  0081:0131:3
09430100 T  0081:0132:1
09431000 T  0081:0134:1
09432000 T  0081:0134:1
09433000 T  0081:0137:2
09434000 T  0081:0140:1
09435000 T  0081:0145:2
09436000 T  0081:0149:3
09437000 T  0081:0151:3
START OF SEGMENT **********   85
09438000 T  0081:0153:3
09439000 T  0081:0153:3
09440000 T  0081:0153:3
09441000 T  0081:0153:3
09442000 T  0081:0153:3
09443000 T  0081:0153:3
09444000 T  0081:0153:3
09445000 T  0081:0153:3
09446000 T  0081:0153:3
85 IS   10 LONG, NEXT SEG   81
09447000 T  0081:0153:3
09447010 T  0081:0157:3
09447020 T  0081:0160:1
09447030 T  0081:0162:0
09447040 T  0081:0163:3
09447050 T  0081:0165:2
09447060 T  0081:0166:0
09447070 T  0081:0169:3
09447080 T  0081:0170:1
09447090 T  0081:0173:2
09447100 T  0081:0179:3
09448000 T  0081:0182:0
81 IS  186 LONG, NEXT SEG   73
73 IS  101 LONG, NEXT SEG    3
10000000 T  0003:0505:3
10228000 T  0003:0505:3
10229000 T  0003:0505:3
10230000 T  0003:0505:3
START OF SEGMENT **********   86
10231000 T  0086:0000:0
10232000 T  0086:0000:0
```

```
                    ORS REMAINING IN THIS ROW OF INFO;
PROCEDURE PUTOGETHER(CHAR); REAL CHAR;
        BEGIN
STREAM PROCEDURE PACKINFO(INFO,ISKIP,COUNT,ASKIP,ACCUM);

                    VALUE ISKIP,COUNT,ASKIP;
                    BEGIN DI ← INFO; DI ← DI+ISKIP;
                          SI ← ACCUM;SI ← SI+ASKIP; SI ← SI+3;
                          DS ← COUNT CHR END PACKINFO;
                  INTEGER COUNT,SKIPCOUNT;
                    IF (COUNT ← CHAR.[12:6]) + CHARCOUNT > 2047
                      THEN BEGIN FLAG(142);  TB1← TRUE  END
                      ELSE BEGIN
                          IF COUNT > REMCOUNT
                            THEN BEGIN
                               SKIPCOUNT ← COUNT-(COUNT←REMCOUNT);
                               REMCOUNT ← 2047 END
                            ELSE REMCOUNT ← REMCOUNT-COUNT;
                          GT1 ← CHARCOUNT DIV 8 + NEXTINFO;
                          PACKINFO(INFO[GT1.LINKR,GT1.LINKC],CHARCOUNT,[45:3],
                          COUNT,0,CHAR);
                          IF SKIPCOUNT ≠ 0 THEN
                              PACKINFO(INFO[NEXTINFO.LINKR+1,0],0,SKIPCOUNT,
                                COUNT,CHAR);
                          CHARCOUNT ← CHARCOUNT+SKIPCOUNT+COUNT END
                END PUTOGETHER;

STREAM PROCEDURE SCAN(D,S,Q,N,J); VALUE J,N,Q;
                BEGIN DI←D;DI←DI+11;SI←S;SI←SI+3;
                      IF N SC=DC THEN
                      IF SC>"0" THEN
                      BEGIN DI←LOC J; DI←DI+7;
                            IF SC≤DC THEN
                            BEGIN J←SI;DI←J;SI←LOC Q;SI←SI+6;DS←CHR;
                                  DI←S;DI←DI+2;DS←CHR;
                END  END  END;
                   INTEGER LASTRESULT;
                      REAL K,N,ELCLASS;
                      DEFINE I=NXTELBT#;
                      LABEL FINAL,PACKIN;
                      LABEL BACK,SKSC,EXIT;
                      TB1← FALSE;
                      CHARCOUNT←(NEXTINFO-LASTINFO)×8;
                      DEFINECTR ← 1; LASTRESULT ← 2;
                      REMCOUNT ← (256 - NEXTINFO MOD 256) × 8;
                      NEXTINFO←LASTINFO;
                      IF J≠0 THEN N←TAKE(LASTINFO+1),[12:6];
                      K←0;
BACK:                 STOPDEFINE←TRUE;
                      ELCLASS←TABLE(NXTELBT);
SKSC:                 NXTELBT←NXTELBT-1;
                      IF MACRO THEN
                      BEGIN IF ELCLASS=COMMA THEN
                            IF K=0 THEN
FINAL:                      BEGIN PUTOGETHER("1#0000"); GO TO EXIT END
                            ELSE GO PACKIN;
                            IF ELCLASS=LEFTPAREN OR ELCLASS=LFTBRKET THEN
```

```
                                                                10233000 T  0086:0000:0
                                                                10234000 T  0086:0000:0
                                                                10235000 T  0086:0000:0
                                                                10236000 T  0086:0000:0
                                                      START OF SEGMENT *********   87
                                                                10237000 T  0087:0000:0
                                                                10238000 T  0087:0000:0
                                                                10239000 T  0087:0000:1
                                                                10240000 T  0087:0001:3
                                                                10241000 T  0087:0002:1
                                                                10242000 T  0087:0002:1
                                                                10243000 T  0087:0004:1
                                                                10244000 T  0087:0007:2
                                                                10245000 T  0087:0009:2
                                                                10246000 T  0087:0009:2
                                                                10247000 T  0087:0010:0
                                                                10248000 T  0087:0012:0
                                                                10249000 T  0087:0012:1
                                                                10250000 T  0087:0016:0
                                                                10251000 T  0087:0018:0
                                                                10252000 T  0087:0021:2
                                                                10253000 T  0087:0022:0
                                                                10254000 T  0087:0023:2
                                                                10255000 T  0087:0026:1
                                                                10256000 T  0087:0027:2
                                                                10257000 T  0087:0029:2
                                                       87 IS   32 LONG, NEXT SEG   86
                                                                10257100 T  0086:0000:0
                                                                10257200 T  0086:0000:0
                                                                10257300 T  0086:0001:2
                                                                10257400 T  0086:0001:3
                                                                10257500 T  0086:0002:1
                                                                10257600 T  0086:0003:3
                                                                10257700 T  0086:0004:0
                                                                10257800 T  0086:0006:0
                                                                10257900 T  0086:0006:1
                                                                10258000 T  0086:0007:2
                                                                10258100 T  0086:0007:2
                                                                10258200 T  0086:0007:2
                                                                10258300 T  0086:0007:2
                                                                10259000 T  0086:0007:2
                                                                10260000 T  0086:0007:2
                                                                10261000 T  0086:0007:3
                                                                10262000 T  0086:0009:3
                                                                10263000 T  0086:0011:2
                                                                10263100 T  0086:0013:2
                                                                10263110 T  0086:0014:0
                                                                10263200 T  0086:0017:3
                                                                10263300 T  0086:0018:0
                                                                10263400 T  0086:0019:3
                                                                10263500 T  0086:0021:2
                                                                10263600 T  0086:0022:0
                                                                10263700 T  0086:0022:1
                                                                10263800 T  0086:0023:3
                                                                10263900 T  0086:0025:2
                                                                10264000 T  0086:0029:2
                                                                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;              10278000 T  0086:0062:1
                    IF ELCLASS ≠ CROSSHATCH THEN GO BACK;                           10279000 T  0086:0067:2
                    IF DEFINECTR ≠ 1                                                10280000 T  0086:0068:0
                      THEN BEGIN STOPDEFINE ← TRUE;                                 10281000 T  0086:0068:1
                          IF ELCLASS←TABLE(I)≠COMMA THEN                            10282000 T  0086:0070:0
                          DEFINECTR←DEFINECTR-1; GO SKSC END;                       10283000 T  0086:0072:0
EXIT:           DEFINECTR← 0;                                                       10284000 T  0086:0074:0
                NEXTINFO ←(CHARCOUNT+7) DIV 8+NEXTINFO;                             10285000 T  0086:0075:3
          END  DEFINEGEN;                                                           10286000 T  0086:0078:0
                                                                              86 IS   81 LONG, NEXT SEG     3
PROCEDURE DBLSTMT;                                                                  12002000 T  0003:0505:3
          BEGIN                                                                     12003000 T  0003:0505:3
          REAL S,T;                                                                 12004000 T  0003:0505:3
                                                                          START OF SEGMENT **********   88
          LABEL L1,L2,L3,EXIT;                                                      12005000 T  0088:0000:0
          S←0;                                                                      12006000 T  0088:0000:0
          IF STEPI≠LEFTPAREN THEN ERR(281)                                          12007000 T  0088:0000:1
          ELSE                                                                      12008000 T  0088:0002:1
     L1:  BEGIN                                                                     12009000 T  0088:0003:2
               IF STEPI=COMMA THEN                                                  12010000 T  0088:0004:0
               BEGIN                                                                12011000 T  0088:0005:2
               DPTOG←TRUE;                                                          12012000 T  0088:0005:3
               IF STEPI=ADOP THEN STEPIT;                                           12013000 T  0088:0006:0
               EMITNUM(NL0);                                                        12014000 T  0088:0008:0
               EMITNUM(IF ELBAT[I-1].ADDRESS =SUB THEN -NHI ELSE NHI);              12015000 T  0088:0009:2
               DPTOG←FALSE;                                                         12016000 T  0088:0013:2
               STEPIT;                                                              12017000 T  0088:0014:0
                    GO TO L2;                                                       12018000 T  0088:0014:1
               END;                                                                 12019000 T  0088:0015:2
               IF TABLE(I+1)=COMMA THEN                                             12020000 T  0088:0015:2
                    BEGIN                                                           12021000 T  0088:0016:1
                    IF ELCLASS=ADOP  OR ELCLASS=MULOP THEN                          12022000 T  0088:0017:2
```

```
          BEGIN                                                                 12023000 T  0088:0019:2
          EMITO(ELBAT[I].ADDRESS+1);                                            12024000 T  0088:0019:3
          IF S←S-1≤0 THEN FLAG(282); STEPIT;                                    12025000 T  0088:0021:3
          GO TO L3;                                                             12026000 T  0088:0025:2
          END;                                                                  12027000 T  0088:0025:3
     IF ELCLASS=ASSIGNOP THEN                                                   12028000 T  0088:0025:3
          BEGIN                                                                 12029000 T  0088:0026:0
          IF S≠1 THEN FLAG(283); S←0;    STEPIT;                                12030000 T  0088:0026:1
          DO                                                                    12031000 T  0088:0030:0
          BEGIN                                                                 12032000 T  0088:0030:0
      IF ELCLASS  ≠COMMA THEN BEGIN ERR(284);GO EXIT END;                       12033000 T  0088:0030:0
          STEPIT;                                                               12034000 T  0088:0032:1
          IF ELCLASS≤INTID AND ELCLASS≥REALID THEN                              12035000 T  0088:0033:2
          BEGIN  EMITN(ELBAT[I].ADDRESS); STEPIT END                            12036000 T  0088:0034:1
                                        ELSE VARIABLE(FL);                      12037000 T  0088:0037:2
                              EMITO(STD) END UNTIL S←S+1=2 ;                    12038000 T  0088:0038:1
          IF ELCLASS≠RTPAREN THEN ERR(285) ELSE STEPIT;                         12039000 T  0088:0041:3
          GO TO EXIT;                                                           12040000 T  0088:0044:1
          END;                                                                  12041000 T  0088:0045:2
     IF ELCLASS≤INTID AND ELCLASS≥BOOID THEN                                    12042000 T  0088:0045:2
          BEGIN                                                                 12043000 T  0088:0046:1
          CHECKFR(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
     STREAM PROCEDURE SET(S,D,K,E); VALUE K,E;                                  12103000 T  0089:0000:0
          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;
                        ACCUM[0]←0;
                        ACCUM[0].CLASS←DEFINEDID;
                        COUNT←K+2;
                        SCRAM←ACCUM[1] MOD 125;
                        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;

                    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;
                IF GTA1[J←J-1]≠0 THEN FLAG(23);
        DEFINE
                ADDC=532480#,
                SUBC=1581056#,
                EMITSTORE=EMITPAIR#;
        PROCEDURE PURGF(STOPPER);
                    VALUE STOPPER;
                    REAL STOPPER;
         BEGIN
           INTEGER POINTER;

           LABEL RECOV; DEFINE ELCLASS = KLASSF#;
           REAL J,N,OCR,TL,ADD;
              POINTER←LASTINFO;
            WHILE POINTER ≥ STOPPER
            DO
              BEGIN
                        IF ELCLASS←(GT1←TAKE(POINTER)).CLASS=NONLITNO
                        THEN BEGIN
                        NCII←NCII-1;
                             EMITNUM(TAKE(POINTER+1));
                             EMITSTORE(MAXSTACK,STD);
                             MAXSTACK←(G←MAXSTACK)+1;
                             J←L; L←GT1.LINK;
                             DO
                               BEGIN
                                  GT4←GET(L);
                                  EMITV(G)
                               END
```

```
                12118000 T  0089:0022:0
                12119000 T  0089:0025:2
                12120000 T  0089:0026:0
                12121000 T  0089:0028:1
                12122000 T  0089:0030:0
                12123000 T  0089:0031:3
                12124000 T  0089:0032:0
                12125000 T  0089:0033:2
                12126000 T  0089:0035:3
                12127000 T  0089:0039:3
                12127100 T  0089:0040:0
                12128000 T  0089:0042:0
         89 IS   46 LONG, NEXT SEG    3
                13197000 T  0003:0505:3
                13198000 T  0003:0505:3
                13199000 T  0003:0505:3
        START OF SEGMENT *********   90
                13200000 T  0090:0000:0
                13201000 T  0090:0001:2
                13202000 T  0090:0002:0
                13203000 T  0090:0003:3
                13204000 T  0090:0004:1
                13205000 T  0090:0006:0
                13206000 T  0090:0007:2
                13207000 T  0090:0008:1
                13208000 T  0090:0009:3
         90 IS   12 LONG, NEXT SEG    3
                13209000 T  0003:0505:3
                13210000 T  0003:0505:3
                13211000 T  0003:0509:3
                13212000 T  0003:0509:3
                13213000 T  0003:0509:3
                13214000 T  0003:0509:3
                13215000 T  0003:0509:3
                13216000 T  0003:0509:3
                13217000 T  0003:0509:3
                13218000 T  0003:0509:3
                13219000 T  0003:0509:3
        START OF SEGMENT *********   91
                13220000 T  0091:0000:0
                13221000 T  0091:0000:0
                13222000 T  0091:0000:0
                13223000 T  0091:0000:1
                13224000 T  0091:0001:2
                13225000 T  0091:0002:0
                13226000 T  0091:0002:0
                13227000 T  0091:0004:1
                13228000 T  0091:0005:3
                13229000 T  0091:0006:1
                13230000 T  0091:0008:1
                13231000 T  0091:0009:3
                13232000 T  0091:0011:2
                13233000 T  0091:0013:2
                13234000 T  0091:0014:0
                13235000 T  0091:0014:0
                13236000 T  0091:0015:2
                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  0091:0086:0
               END;                                                         13283000 T  0091:0089:2
     XREFDUMP(POINTER); % DUMP XREF INFO                          %110-     13283500 C  0091:0089:2
     GT2←TAKE(POINTER+1);                                                   13284000 T  0091:0091:2
     GT3←GT2.PURPT;                                                         13285000 T  0091:0092:1
   STACKHEAD[(0&GT2[12:12:36])MOD 125]←TAKE(POINTER).LINK;                  13286000 T  0091:0094:0
     POINTER←POINTER-GT3                                                    13287000 T  0091:0097:3
          END                                                               13288000 T  0091:0098:0
  END ;                                                                     13289000 T  0091:0099:2
 LASTINFO←POINTER;                                                          13290000 T  0091:0099:3
 NEXTINFO←STOPPER                                                           13291000 T  0091:0100:0
END;                                                                        13292000 T  0091:0100:0
                                                                  91 IS  105 LONG, NEXT SEG    3
PROCEDURE F;                                                                13293000 T  0003:0509:3
```

```
COMMENT                                                                         13294000 T  0003:0509:3
       F IS THE PROCEDURE WHICH PLACES AN ENTRY IN INFO AND                     13295000 T  0003:0509:3
       HOOKS IT INTO STACKHEAD. THE PREVIOUS STACKHEAD LINK                     13296000 T  0003:0509:3
       IS SAVED IN THE LINK OF THE ELBAT WORD IN THE NEW ENTRY                  13297000 T  0003:0509:3
       F PREVENTS AN ENTRY FROM OVERFLOWING A ROW,STARTING AT THE               13298000 T  0003:0509:3
       BEGINNING OF THE NEXT ROW IF NECESSARY                          ;        13299000 T  0003:0509:3
 BEGIN                                                                          13300000 T  0003:0509:3
    REAL WORDCOUNT,RINX;                                                        13301000 T  0003:0509:3
                                                             START OF SEGMENT **********   92
       IF RINX←(NEXTINFO+WORDCOUNT←(COUNT+18)DIV 8    ).LINKR ≠                 13302000 T  0092:0000:0
              NEXTINFO.LINKR                                                    13303000 T  0092:0003:2
THEN BEGIN PUT(125&(RINX×256-NEXTINFO)[27:40:8],NEXTINFO);                      13304000 T  0092:0003:2
       NEXTINFO←256×RINX END;                                                   13305000 T  0092:0007:3
       IF SPECTOG THEN                                                          13305100 T  0092:0009:2
       IF NOT MACROID THEN                                                      13305200 T  0092:0009:2
           UNHOOK;                                                              13305300 T  0092:0010:0
                                                                                13306000 T  0092:0011:2
       ACCUM[0].INCR←WORDCOUNT;                                                 13307000 T  0092:0011:2
       IF NOT INLINETOG OR MACROID THEN BEGIN                                   13307500 T  0092:0013:3
        ACCUM[0].LINK ←STACKHEAD[SCRAM];STACKHEAD[SCRAM]←NEXTINFO;              13308000 T  0092:0015:2
                         END;                                                   13308500 T  0092:0019:2
        ACCUM[1].PURPT←NEXTINFO-LASTINFO;                                       13309000 T  0092:0019:2
MOVE(WORDCOUNT,ACCUM,INFO[NEXTINFO.LINKR,NEXTINFO.LINKC]);                      13310000 T  0092:0022:0
       IF XREF THEN % MAKE DECLARATION REFERENCE                     %110-      13310050 C  0092:0025:3
%       IF (ACCUM[0].CLASS ≠ DEFINEDID OR NOT                        %110-      13310075 C  0092:0026:1
%           BOOLEAN(ACCUM[0].FORMAL)) THEN % NOT DEFINE PARAMETER%110-          13310080 C  0092:0026:1
         BEGIN                                                       %110-      13310100 C  0092:0026:1
           XREFINFO[NEXTINFO] :=                                     %110-      13310200 C  0092:0027:2
             IF SPECTOG THEN                                         %110-      13310300 C  0092:0029:3
               XREFINFO[ELBAT[I]]                                    %110-      13310350 C  0092:0030:0
             ELSE                                                    %110-      13310400 C  0092:0033:3
               ((XLUN := XLUN + 1) & SGNO SEGNOF);                   %110-      13310450 C  0092:0034:0
           IF SPECTOG THEN % JUST GO BACK AND FIX UP XREF ENTRY      %110-      13310500 C  0092:0037:2
             XMARK(DECLREF)                                          %110-      13310525 C  0092:0037:2
           ELSE                                                      %110-      13310550 C  0092:0042:0
             XREFIT(NEXTINFO,CARDNUMBER,IF PTOG AND NOT STREAMTOG%110-          13310575 C  0092:0042:0
                      THEN NORMALREF ELSE DECLREF);                  %110-      13310580 C  0092:0042:0
         END                                                         %110-      13310600 C  0092:0047:2
%       ELSE % DEFINE PARAMETERS - DONT CROSS REF.                   %110-      13310700 C  0092:0047:2
%         XREFINFO[NEXTINFO] := 0                                    %110-      13310750 C  0092:0047:2
       ELSE                                                          %110-      13310800 C  0092:0047:2
         IF DEFINING.[1:1]  THEN % WE ARE DOING XREFING              %110-      13310900 C  0092:0047:2
           XREFINFO[NEXTINFO] := 0;                                  %110-      13310950 C  0092:0048:1
        LASTINFO←NEXTINFO;                                                      13311000 T  0092:0052:1
         NEXTINFO←NEXTINFO+WORDCOUNT                                            13312000 T  0092:0053:2
 END;                                                                           13313000 T  0092:0053:3
                                                          92 IS   57 LONG, NEXT SEG    3
PROCEDURE ENTRY(TYPE);                                                          13314000 T  0003:0509:3
        VALUE TYPE;                                                             13315000 T  0003:0509:3
        REAL TYPE;                                                              13316000 T  0003:0509:3
COMMENT                                                                         13317000 T  0003:0509:3
       ENTRY ASSUMES THAT I IS POINTING AT AN IDENTIFIER WHICH                  13318000 T  0003:0509:3
       IS BEING DECLARED AND MAKES UP THE ELBAT ENTRY FOR IT                    13319000 T  0003:0509:3
       ACCORD TO TYPE        .IF THE ENTRY IS AN ARRAY AND NOT                  13320000 T  0003:0509:3
       A SPECIFICATION THEN A DESCRIPTOR IS PLACED IN THE STACK                 13321000 T  0003:0509:3
       FOR THE UPCOMING COMMUNICATE TO GET STORAGE FOR THE ARRAY(S)    ;        13322000 T  0003:0509:3
 BEGIN                                                                          13323000 T  0003:0509:3
```

```
                J←0;I←I-1;                                                                                    13324000 T  0003:0509:3
                DO                                                                                            13325000 T  0003:0512:0
                  BEGIN                                                                                       13326000 T  0003:0512:0
                        STOPDEFINE ←TRUE; STEPIT; SCATTERELBAT;                                               13327000 T  0003:0512:0
                        IF FORMALF←SPECTOG                                                                    13328000 T  0003:0513:3
                           THEN                                                                               13329000 T  0003:0513:3
                            BEGIN                                                                             13330000 T  0003:0514:1
                              IF ELCLASS≠SECRET                                                               13331000 T  0003:0515:2
                                 THEN FLAG(002);                                                              13332000 T  0003:0515:2
                                      BUP←BUP+1                                                               13333000 T  0003:0517:2
;               KLASSF←TYPE;MAKEUPACCUM; E;J←J+1;                                                             13333500 T  0003:0517:2
                            END                                                                               13334000 T  0003:0521:2
                           ELSE                                                                               13335000 T  0003:0521:2
                            BEGIN                                                                             13336000 T  0003:0521:2
                              IF ELCLASS>IDMAX AND ELCLASS≤MULOP                          %112-               13337000 P  0003:0521:3
                           THEN IF ELCLASS= POLYSHV THEN ELCLASS←TYPE ELSE FLAG(3);                           13338000 T  0003:0522:1
                              IF LEVELF=LEVEL                                                                 13339000 T  0003:0527:2
                                THEN FLAG(001);                                                               13340000 T  0003:0527:3
                              VONF←P2;                                                                        13341000 T  0003:0529:2
                            FORMALF←PTOG;                                                                     13341100 T  0003:0530:0
                        IF XREF AND NOT SPECTOG THEN % ERASE PREVIOUS XREF ENTRY.                             13341200 C  0003:0530:1
                          XREFPT←XREFPT-REAL(ELBAT[I]≠0); % GET RID OF LAST CREF                              13341300 C  0003:0532:0
                KLASSF←TYPE; MAKEUPACCUM;E; J←J+1;                                                            13342000 T  0003:0534:1
                IF ((FORMALF←PTOG) OR(STREAMTOG AND NOT STOPGSP)) AND NOT P2                                  13343000 T  0003:0537:3
                  THEN ADDRSF←PJ←PJ+1                                                                         13344000 T  0003:0539:3
                  ELSE IF STOPGSP                                                                             13345000 T  0003:0541:2
                         THEN ADDRSF←0                                                                        13346000 T  0003:0543:2
                         ELSE ADDRSF:=GETSPACE(P2,LASTINFO+1);                                                13347000 T  0003:0544:0
                    PUT(TAKE(LASTINFO)& ADDRSF[16:37:11],LASTINFO);                                           13348000 T  0003:0547:2
                  END END                                                                                     13349000 T  0003:0549:3
                                                                                                              13350000 T  0003:0549:3
                   UNTIL STEPI≠COMMA OR STOPENTRY; GTA1[0]←J                                                  13351000 T  0003:0549:3
                END;                                                                                          13352000 T  0003:0552:0
                 PROCEDURE UNHOOK;                                                                            13353000 T  0003:0553:2
                COMMENT                                                                                       13354000 T  0003:0553:2
                  UNHOOK ASSUMES THAT THE WORD IN ELBAT[I] POINTS TO A PSUEDO ENTRY                           13355000 T  0003:0553:2
                  FOR APARAMETER,ITS JOB IS TO UNHOOK THAT FALSE ENTRY SO THAT                                13356000 T  0003:0553:2
                  E WILL WORK AS NORMAL.                                                      ;               13357000 T  0003:0553:2
                  BEGIN                                                                                       13358000 T  0003:0553:2
                    REAL LINKT,A,LINKP;                                                                       13359000 T  0003:0553:2
                                                                                                      START OF SEGMENT **********   93
                    LABEL L;                                                                                  13360000 T  0093:0000:0
                                LINKT←STACKHEAD[SCRAM]     ; LINKP←ELBAT[I],LINK;                             13361000 T  0093:0000:0
                                IF LINKT=LINKP THEN STACKHEAD[SCRAM]←TAKE(LINKT),LINK                         13362000 T  0093:0002:1
                                               ELSE                                                           13363000 T  0093:0005:3
                           L: IF A←TAKE(LINKT),LINK=LINKP                                                     13364000 T  0093:0006:0
                              THEN PUT((TAKE(LINKT))&(TAKE(A))[35:35:13],LINKT)                               13365000 T  0093:0008:1
                              ELSE BEGIN LINKT←A; GO TO L END;                                                13366000 T  0093:0012:0
                END;                                                                                          13367000 T  0093:0014:1
                                                                                                    93 IS   17 LONG, NEXT SEG    3
                PROCEDURE MAKEUPACCUM;                                                                        13368000 T  0003:0553:2
                  BEGIN                                                                                       13369000 T  0003:0553:2
                    IF PTOG                                                                                   13370000 T  0003:0553:2
                      THEN GT1←LEVELF ELSE GT1←LEVEL;                                                         13371000 T  0003:0554:0
                    ACCUM[0]← ABS(ELBAT[I] & KLASSF[2:41:7] & REAL(FORMALF)[9:47:1]                           13372000 T  0003:0556:1
                                & REAL(VONF)[10:47:1] & GT1[11:43:5] &ADDRSF[16:37:11]                        13373000 T  0003:0559:3
                                )                                                                             13374000 T  0003:0562:0
```

```
  END;
PROCEDURE ARRAE;
 BEGIN
     INTEGER SAVEINFO;

     LABEL BETA1;
           TYPEV←REALARRAYID;
            IF T1←GTA1[J←J+1]=0 THEN J←J+1
                               ELSE
                                 IF T1=OWNV THEN
                                       BEGIN
                                            P2←TRUE;IF SPECTOG THEN
                                            FLAG(13)
                                      END
                                        ELSE
                                       TYPEV←REALARRAYID+T1-REALV;
 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)
```

```
                13375000 T  0003:0562:1
                13376000 T  0003:0563:3
                13377000 T  0003:0563:3
                13378000 T  0003:0563:3
START OF SEGMENT **********   94
                13379000 T  0094:0000:0
                13380000 T  0094:0000:0
                13381000 T  0094:0000:1
                13382000 T  0094:0004:0
                13383000 T  0094:0005:2
                13384000 T  0094:0006:0
                13385000 T  0094:0006:1
                13386000 T  0094:0007:3
                13387000 T  0094:0008:1
                13388000 T  0094:0009:2
                13389000 T  0094:0009:2
                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
```

```
      END ELSE
            IF FIRSTX=4095
              THEN
                BEGIN
                  ADJUST;
                  FIRSTX←L;
                END;
          AJUMP←FALSE
    END;
     PROCEDURE JUMPCHKNX;
    COMMENT JUMPCHKNX DETERMINES WHETHER ANY EXECUTABLE CODE HAS BEEN
            EMITTED AND IF SO WHETHER IT WAS JUST PREVIOUS TO THE
            NON EXECUTABLE ABOUT TO BE EMITTED.IF BOTH THEN L IS BUMPED
            AND SAVED FOR A LATER BRANCH;
    IF NOT SPECTOG THEN
    BEGIN
       IF FIRSTX≠4095
         THEN
          BEGIN
           IF NOT AJUMP
             THEN
               SAVEL←BUMPL;
          AJUMP←TRUE
          END;ADJUST
    END;
    PROCEDURE SEGMENTSTART(SAVECODE);VALUE SAVECODE;BOOLEAN SAVECODE;
        BEGIN
        STREAM PROCEDURE PRINT(SAVECODE,ADR,FIEL); VALUE SAVECODE,ADR;

             BEGIN
             LABEL L1;
             DI:=FIEL; DS:=8 LIT" ";
             SI:=FIEL; DS:=9 WDS; DI:=DI-3;
             SAVECODE(DS:=38 LIT "START OF SAVE SEGMENT; BASE ADDRESS = ";
               JUMP OUT TO L1);
             DS:=38 LIT " START OF REL SEGMENT; DISK ADDRESS = ";
    L1;
             SI:=LOC ADR; DS:=5 DEC;
             END PRINT;
        MOVE(1,SAVECODE,CODE(0));
        IF SAVECODE AND INTOG AND NOT DECKTOG THEN FLAG(57);
        IF LISTER OR SEGSTOG THEN
             BEGIN
             PRINT(SAVECODE,IF SAVECODE THEN CORADR ELSE DISKADR,LIN[*]);
             IF NOHEADING THEN DATIME; WRITELINE;
             END;
        END SEGMENTSTART;

    PROCEDURE SEGMENT(SIZE,FR); VALUE SIZE,FR; INTEGER SIZE,FR;
        BEGIN
        STREAM PROCEDURE PRINT(SIZE,FIEL); VALUE SIZE;

             BEGIN
             DI:=FIEL; DS:=8 LIT" ";
             SI:=FIEL; DS:=14 WDS;
             DI:=DI-16; DS:=6 LIT"SIZE= ";
             SI:=LOC SIZE; DS:=4 DEC; DS:=6 LIT" WORDS"
```

```
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
13615000 T  0003:0575:3
13616000 T  0003:0578:0
13617000 T  0003:0578:0
13618000 T  0003:0578:0
13619000 T  0003:0578:0
13620000 T  0003:0578:0
13621000 T  0003:0578:0
13622000 T  0003:0578:1
13623000 T  0003:0579:2
13624000 T  0003:0579:2
13625000 T  0003:0579:3
13626000 T  0003:0580:0
13627000 T  0003:0580:0
13628000 T  0003:0580:1
13629000 T  0003:0583:2
13630000 T  0003:0583:2
13631000 T  0003:0583:3
13632000 T  0003:0586:0
13632100 T  0003:0586:0
13633000 T  0003:0586:0
START OF SEGMENT **********   95
13634000 T  0095:0000:0
13635000 T  0095:0000:0
13636000 T  0095:0000:0
13637000 T  0095:0001:3
13638000 T  0095:0002:0
13639000 T  0095:0008:0
13640000 T  0095:0009:2
13641000 T  0095:0014:0
13642000 T  0095:0014:0
13643000 T  0095:0014:1
13651000 T  0095:0014:1
13651100 T  0095:0019:2
13652000 T  0095:0022:1
13652500 T  0095:0024:0
13653000 T  0095:0024:1
13653500 T  0095:0027:3
13654000 T  0095:0039:2
13655000 T  0095:0039:2
 95 IS   40 LONG, NEXT SEG    3
13657000 T  0003:0586:0
13660000 T  0003:0586:0
13661000 T  0003:0586:0
START OF SEGMENT **********   96
13663000 T  0096:0000:0
13665000 T  0096:0000:0
13667000 T  0096:0001:3
13668000 T  0096:0002:0
13670000 T  0096:0003:2
```

```
        END PRINT;
    STREAM PROCEDURE DOIT(C,A,I,S,F,W); VALUE C,A,F,W;
        BEGIN  LOCAL N;
        DI:=S; DS:=8 LIT" "; SI:=S; DS:=9 WDS;
        DI:=DI-8; SI:=LOC W; DS:=4 DEC;
        SI:=I; SI:=SI+10;DI:=LOC N; DI:=DI+7; DS:=CHR;
        DI:=S; SI:=LOC F; SI:=SI+7; DS:=CHR; SI:=LOC C;
        DS:=3 DEC; DS:=4 DEC;SI:=I; SI:=SI+11;DS:=N CHR;
        END DOIT;
    IF LISTER OR SEGSTOG THEN
        BEGIN
        PRINT(SIZE,LIN[*]);
        IF NOHEADING THEN DATIME; WRITELINE;
        END;
    IF STUFFTOG THEN IF FR>0 THEN IF LEVEL>1 THEN
        BEGIN
        KLASSF:=TAKE(PROINFO),CLASS;
       IF FR > 1024 THEN FR←FR-1024;
        DOIT(KLASSF,FR,INFO[PROINFO,LINKR,PROINFO,LINKC],
            TWXA[0],SAF,SIZE);
        WRITE(STUFF,10,TWXA[*]);
        END;
    IF SIZE>SEGSIZEMAX THEN SEGSIZEMAX:=SIZE;
    END SEGMENT;

STREAM PROCEDURE MOVECODE(EDOC,TEDOC);
 BEGIN LOCAL T1,T2,T3;
    SI←EDOC;T1←SI;
    SI←TEDOC;T2←SI;
    SI←LOC EDOC;
    SI←SI+3;
    DI←LOC T3;
    DI←DI+5;
    SKIP 3 DB;
  15(IF SB THEN DS← 1 SET ELSE DS←1 RESET;SKIP 1 SB);
    SI← LOC EDOC;
    DI← LOC T2;
    DS← 5 CHR;
    3(IF SB THEN DS←1 SET ELSE DS←1 RESET;SKIP 1 SB);
    DI←T3;
    SI←LOC T2;
    DS←WDS;
    DI←LOC T3;
    DI←DI+5;
    SKIP 3 DB;
    SI←LOC TEDOC;
    SI←SI+3;
  15(IF SB THEN DS←1 SET ELSE DS← 1 RESET;SKIP 1 SB);
    SI← LOC TEDOC;
    DI← LOC T1;
    DS← 5 CHR;
   3(IF SB THEN DS←1 SET ELSE DS←1 RESET;SKIP 1 SB);
    DI←T3;
    SI←LOC T1;
    DS←WDS
  END;
PROCEDURE ENTER(TYPE);
```

```
13673000 T  0096:0004:1
13673100 T  0096:0005:2
13673150 T  0096:0005:2
13673200 T  0096:0005:2
13673250 T  0096:0007:2
13673300 T  0096:0007:3
13673350 T  0096:0009:2
13673400 T  0096:0010:0
13673450 T  0096:0011:3
13674000 T  0096:0012:0
13674500 T  0096:0013:2
13675000 T  0096:0013:3
13676000 T  0096:0015:2
13677000 T  0096:0026:1
13677100 T  0096:0026:1
13677150 T  0096:0029:3
13677200 T  0096:0030:0
13677250 T  0096:0032:0
13677300 T  0096:0034:1
13677400 T  0096:0037:3
13677500 T  0096:0039:2
13677600 T  0096:0043:2
13678000 T  0096:0043:2
13681000 T  0096:0045:2
96 IS   47 LONG, NEXT SEG   3
13683000 T  0003:0586:0
13684000 T  0003:0586:0
13685000 T  0003:0586:0
13686000 T  0003:0586:1
13687000 T  0003:0587:2
13688000 T  0003:0587:2
13689000 T  0003:0587:3
13690000 T  0003:0587:3
13691000 T  0003:0588:0
13692000 T  0003:0588:0
13693000 T  0003:0590:0
13694000 T  0003:0590:1
13695000 T  0003:0590:1
13696000 T  0003:0591:2
13697000 T  0003:0593:2
13698000 T  0003:0593:2
13699000 T  0003:0593:3
13700000 T  0003:0593:3
13701000 T  0003:0594:0
13702000 T  0003:0594:0
13703000 T  0003:0594:1
13704000 T  0003:0594:1
13705000 T  0003:0595:2
13706000 T  0003:0597:2
13707000 T  0003:0597:2
13708000 T  0003:0597:3
13709000 T  0003:0597:3
13710000 T  0003:0599:3
13711000 T  0003:0600:0
13712000 T  0003:0600:0
13713000 T  0003:0600:1
13714000 T  0003:0600:1
```

```
                VALUE TYPE;
             REAL TYPE;
 BEGIN
        G←GTA1[J←J-1];
        IF NOT SPECTOG
          THEN
            BEGIN
              IF NOT P2
                 THEN IF P2←(G=OWNV)
                         THEN G←GTA1[J←J-1];
              IF NOT P3
                 THEN IF P3←(G=SAVEV)
                         THEN G←GTA1[J←J-1]
            END;
        IF G≠0 THEN FLAG(25) ELSE ENTRY(TYPE)
 END;
PROCEDURE HTTEOAP(GOTSTORAGE,RELAD,STOPPER,PRTAD);
             VALUE GOTSTORAGE,RELAD,STOPPER,PRTAD;
             BOOLEAN GOTSTORAGE;
             REAL RELAD,STOPPER,PRTAD;
  BEGIN
              IF FUNCTOG
                THEN
                   BEGIN
                       EMITV(513);
                       EMITO(RTN)
                   END
                ELSE
                       EMITO(XIT);
                CONSTANTCLEAN;
                PURGE(STOPPER);
      MOVE(1,CODE(0),Z); PROGDFSCBLDR(PRTAD,BOOLEAN(Z),(L+3)DIV 4,PDFS);
      END HTTEOAP;
PROCEDURE INLINE;
 BEGIN
  INTEGER SN,LN,P,LS,J; BOOLEAN MKST;

             BOOLEAN FLIPFLOP;
     INTEGER PN;
   LABEL L1,L2,L3;
     PN←1 ;
             FLIPFLOP←INLINETOG←TRUE;P←0;MKST←FALSE;LS←L;EMITO(NOP);
    IF STEPI≠LEFTPAREN THEN FLAG(59);
    IF TABLE(I+1)=COLON THEN BEGIN STEPI;GO TO L2 END ;
L1: IF STEPI>IDMAX THEN BEGIN FLAG(465); GO TO L2 END ;
       ACCUM[0]←0&P[16:37:11]&LOCLID[2:41:7]&SCRAM[35:35:13];
          E;IF FLIPFLOP THEN BEGIN FLIPFLOP←FALSE;LN←SN←LASTINFO END;
          IF STEPI=COMMA OR ELCLASS=COLON OR ELCLASS=RTPAREN
      THEN BEGIN I←I-2;STEPIT END
      ELSE IF ELCLASS≠ASSIGNOP THEN FLAG(60) ELSE STEPIT;
      AEXP;
 L2:  IF ELCLASS=COLON THEN
      BEGIN IF MKST THEN FLAG(99); MKST←TRUE; EMITO(MKS); P←P+2;
            IF TABLE(I+1)≠RTPAREN THEN GO TO L1; STEPIT
            ;PN←2;
      END ELSE P←P+1;
      IF ELCLASS=COMMA THEN GO TO L1;
```

```
13715000 T  0003:0600:1
13716000 T  0003:0600:1
13717000 T  0003:0600:1
13718000 T  0003:0600:1
13719000 T  0003:0603:2
13720000 T  0003:0603:2
13721000 T  0003:0603:3
13722000 T  0003:0604:0
13723000 T  0003:0604:0
13724000 T  0003:0605:2
13725000 T  0003:0608:1
13726000 T  0003:0608:1
13727000 T  0003:0610:0
13728000 T  0003:0612:1
13729000 T  0003:0613:3
13730000 T  0003:0616:0
13731000 T  0003:0617:2
13732000 T  0003:0617:2
13733000 T  0003:0617:2
13734000 T  0003:0617:2
13735000 T  0003:0617:2
13736000 T  0003:0617:2
13737000 T  0003:0617:2
13738000 T  0003:0617:2
13739000 T  0003:0617:3
13740000 T  0003:0618:1
13741000 T  0003:0618:1
13742000 T  0003:0619:2
13743000 T  0003:0619:2
13744000 T  0003:0620:1
13745000 T  0003:0621:2
13746000 T  0003:0621:3
13747000 T  0003:0628:0
13748000 T  0003:0628:1
13749000 T  0003:0628:1
13750000 T  0003:0628:1
START OF SEGMENT **********   97
13750500 T  0097:0000:0
13750600 T  0097:0000:0
13751000 T  0097:0000:0
13751100 T  0097:0000:0
13752000 T  0097:0000:1
13753000 T  0097:0005:2
13753100 T  0097:0007:2
13754000 T  0097:0010:1
13755000 T  0097:0013:3
13755500 T  0097:0018:0
13756000 T  0097:0021:2
13757000 T  0097:0023:3
13758000 T  0097:0026:1
13759000 T  0097:0030:0
13760000 T  0097:0030:1
13761000 T  0097:0031:3
13761100 T  0097:0036:1
13761110 T  0097:0038:1
13761200 T  0097:0040:0
13762000 T  0097:0041:3
```

```
                IF FLCLASS≠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
        INTEGER PINFOO,BLKAD;                                                                        14026000 T  0098:0013:3
                 COMMENT LOCAL TO BLOCK TO SAVE WHERE A PROCEDURE IS EMTERED                         14027000 T  0098:0013:3
                 IN INFO;                                                                            14028000 T  0098:0013:3
       REAL MAXSTACKO,LASTINFOT,RELAD,LO,TSUBLEVEL,STACKCTRO;                                        14029000 T  0098:0013:3
       INTEGER SGNOO,LOLD,SAVELO,PRTIO,NINFOO;                                                       14030000 T  0098:0013:3
         INTEGER NCIIO;                                                                              14031000 T  0098:0013:3
          INTEGER PROAD ;                                                                            14032000 T  0098:0013:3
            INTEGER FIRSTXO;                                                                         14033000 T  0098:0013:3
```

```
                BOOLEAN FUNCTOGO,AJUMPO;                                                                14034000 T  0098:0013:3
                        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[1].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
                                                                                        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  0099:0004:0
                                   IF NOT STREAMTOG THEN                                                14082000 T  0099:0004:0
                                        BUP←LASTINFO;                                                   14083000 T  0099:0004:1
                                        GETLP: G←TAKE(BUP);                                             14084000 T  0099:0005:3
                                               IF K←G.ADDRESS≠PJ                                        14085000 T  0099:0007:2
                                                 THEN                                                   14086000 T  0099:0008:1
                                                   BEGIN                                                14087000 T  0099:0009:2
IF BUP ≠ BUP:=BUP- TAKE(BUP + 1).PURPT THEN                                             %102-           14088000 P  0099:0009:3
                                                     GO TO GETLP                                        14089000 T  0099:0012:1
                                                   END;                                                 14090000 T  0099:0012:1
```

```
                    TYPEV←G.CLASS;
                                                    G.ADDRESS←F←F+1;
                                              PUT(G,BUP); G.INCR←GT1;
                                              PUT(G,MARK+PJ)
                                             ;BUP←BUP-TAKE(BUP+1),PURPT
                                 END
                                UNTIL PJ←PJ-1=0
                               END;

                                    SPECTOG←FALSE;
                             GO TO HF
                            END
                      END;
                        STACKCT ← 0;
                    WHILE STEPI=DECLARATORS
                    DO
                     BEGIN
                       GTA1[J←J+1]←ELBAT[I].ADDRESS;
                       STOPDEFINE←ERRORTOG←TRUE;
                     END;
IF J =0 THEN GO TO CALLSTATEMENT;
                    P2←P3←FALSE;
                    GO TO DECLSW[GTA1[J]];
OWNERR:FLAG(20);J←J+1;GO TO REALDEC;
SAVERR:FLAG(21);J←J+1;GO TO REALDEC;
AUXMEMERR:FLAG(618);J:=J+1;GO TO REALDEC;
STREAMERR:      IF ELCLASS = LEFTPAREN THEN
                BEGIN
                     I ← I - 1;
                     GO TO CALLSTATEMENT;
                END;
                     FLAG(22);
                     J ← J + 1;
                     GO TO PROCEDUREDEC;
REALDEC:P3←TRUE;ENTER(REALID);GO TO START;
ALPHADEC:P3←TRUE;ENTER(ALFAID);GO TO START;
BOOLEANDEC:P3←TRUE;ENTER(BOOID);GO TO START;
INTEGERDEC:P3←TRUE;ENTER(INTID);GO TO START;
        MONITORDEC:IF SPECTOG
                     THEN BEGIN COMMENT ERROR 463 MEANS THAT A MONITOR
                                  DECLARATION APPEARS IN THE SPECIFICATION
                                  PART OF A PROCEDURE;
                                FLAG(463);
                          END;
                   DO UNTIL FALSE;
              DUMPDEC:IF SPECTOG
                     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;
```

```
                                                                  14091000 T  0099:0013:2
                                                                  14115000 T  0099:0014:1
                                                                  14116000 T  0099:0017:2
                                                                  14117000 T  0099:0020:0
                                                                  14118000 T  0099:0020:1
                                                                  14119000 T  0099:0023:3
                                                                  14120000 T  0099:0024:0
                                                                  14121000 T  0099:0025:3
                                                          99 IS    27 LONG, NEXT SEG   98
                                                                  14122000 T  0098:0046:0
                                                                  14123000 T  0098:0046:1
                                                                  14124000 T  0098:0047:2
                                                                  14125000 T  0098:0047:2
                                                                  14125500 T  0098:0047:2
                                                                  14126000 T  0098:0048:0
                                                                  14127000 T  0098:0048:1
                                                                  14128000 T  0098:0049:3
                                                                  14129000 T  0098:0049:3
                                                                  14130000 T  0098:0052:1
                                                                  14131000 T  0098:0053:3
                                                                  14132000 T  0098:0054:0
                                                                  14133000 T  0098:0055:3
                                                                  14134000 T  0098:0056:1
                                                                  14135000 T  0098:0059:2
                                                                  14136000 T  0098:0061:3
                                                          %112-   14136100 C  0098:0064:1
                                                           % 6    14137000 T  0098:0067:3
                                                           % 6    14137100 T  0098:0068:1
                                                           % 6    14137200 T  0098:0069:2
                                                           % 6    14137300 T  0098:0070:1
                                                           % 6    14137400 T  0098:0071:2
                                                           % 6    14137500 T  0098:0071:2
                                                           % 6    14137600 T  0098:0071:3
                                                           % 6    14137700 T  0098:0073:2
                                                                  14138000 T  0098:0073:3
                                                                  14139000 T  0098:0076:0
                                                                  14140000 T  0098:0078:0
                                                                  14141000 T  0098:0080:0
                                                                  14142000 T  0098:0082:0
                                                                  14143000 T  0098:0082:0
                                                                  14144000 T  0098:0082:1
                                                                  14145000 T  0098:0082:1
                                                                  14146000 T  0098:0082:1
                                                                  14147000 T  0098:0083:3
                                                                  14148000 T  0098:0083:3
                                                                  14149000 T  0098:0084:1
                                                                  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
```

```
                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;

     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;

     GO TO START;

                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 FLCLASS≠COMMA;
                         STOPENTRY←STOPGSP←FALSE;
                         GO TO START;
                SWITCHDEC:
```

```
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
14186000 T  0098:0109:3
14187000 T  0098:0109:3
14188000 T  0098:0112:0
14189000 T  0098:0113:2
14190000 T  0098:0114:1
14191000 T  0098:0115:2
14192000 T  0098:0115:2
14193000 T  0098:0115:3
14194000 T  0098:0116:0
14195000 T  0098:0117:2
14196000 T  0098:0117:2
14197000 T  0098:0117:3
14198000 T  0098:0119:2
14199000 T  0098:0120:0
14200000 T  0098:0120:1
```

```
BEGIN
  LABEL START;

    INTEGER GT1,GT2,GT4,GT5;
   BOOLEAN 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;

GO TO START;
  DEFINEDEC:
    BEGIN LABEL START;

        REAL J,K;
    BOOLEAN STREAM PROCEDURE PARM(S,D,K,J); VALUE K,J;
      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;
            PARM←TALLY;
      END;
STOPENTRY←STOPGSP←TRUE;I←I-1;
      DEFINING := BOOLEAN(REAL(DEFINING) & 1[47:47:1]);          %108-
   DO
    BEGIN
            STOPDEFINE←TRUE;
      STEPIT; MOVE(9,ACCUM[1],GTA1);
      K←COUNT+1; J←GTA1[0]; ENTRY(DEFINEDID);
      GTA1[0]←J+"100000"; J←0;
      IF ELCLASS=LEFTPAREN OR ELCLASS=LFTBRKET THEN
      BEGIN
            DO BEGIN STOPDEFINE←TRUE;
               STEPIT;
               IF (J←J+1)>9 OR PARM(ACCUM[1],GTA1,K,J) OR
                    K>62 THEN BEGIN ERR(141); GO TO START END;
               STOPDEFINE←TRUE;
            END UNTIL STEPI≠COMMA;
            IF ELCLASS≠RTPAREN AND ELCLASS≠RTBRKET THEN ERR(141);
            STOPDEFINE←TRUE;
            STEPIT;
            PUT(TAKE(LASTINFO)&J[16:37:11],LASTINFO);
      END;
     IF ELCLASS≠RELOP
      THEN
         BEGIN
```

```
        14201000 T  0098:0121:2
        14202000 T  0098:0121:2
START OF SEGMENT **********  101
        14203000 T  0101:0000:0
        14204000 T  0101:0000:0
        14205000 T  0101:0000:0
        14206000 T  0101:0001:3
        14207000 T  0101:0003:3
        14217000 T  0101:0004:1
        14218000 T  0101:0006:0
        14219000 T  0101:0006:0
        14220000 T  0101:0006:0
        14221000 T  0101:0010:0
        14222000 T  0101:0011:2
        14222500 T  0101:0012:0
        14223000 T  0101:0012:0
        14223500 T  0101:0013:3
        14224000 T  0101:0013:3
        14251000 T  0101:0014:1
        14252000 T  0101:0016:0
 101 IS   17 LONG, NEXT SEG   98
        14253000 T  0098:0122:0
        14254000 T  0098:0122:1
        14254050 T  0098:0123:2
START OF SEGMENT **********  102
        14254100 T  0102:0000:0
        14254200 T  0102:0000:0
        14254300 T  0102:0000:0
        14254400 T  0102:0001:2
        14254500 T  0102:0002:0
        14254600 T  0102:0002:1
        14254700 T  0102:0003:2
        14254800 T  0102:0003:3
        14255000 T  0102:0004:1
        14255500 C  0102:0007:3
        14256000 T  0102:0009:2
        14257000 T  0102:0010:0
        14258000 T  0102:0010:0
        14259000 T  0102:0010:1
        14259010 T  0102:0013:2
        14259015 T  0102:0016:0
        14259020 T  0102:0018:1
        14259030 T  0102:0020:0
        14259060 T  0102:0020:1
        14259070 T  0102:0021:3
        14259080 T  0102:0022:0
        14259090 T  0102:0026:1
        14259100 T  0102:0031:2
        14259110 T  0102:0031:3
        14259120 T  0102:0033:2
        14259130 T  0102:0036:0
        14259140 T  0102:0037:2
        14259150 T  0102:0037:3
        14259160 T  0102:0040:0
        14260000 T  0102:0040:0
        14261000 T  0102:0040:0
        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
                                                                                                                  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
                                                                                                         START OF SEGMENT **********  103
            BOOLEAN FOUNDLB; % TRUE IF LEFT-BRACKET WAS USED IN FIELD SPEC,%112-               14269080 C  0103:0000:0
            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 = LFTBRKET 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
                                ELSE % MISSING RIGHT BRACKET.                    %112-          14269715 C  0103:0036:0
                              ELSE                                               %112-          14269720 C  0103:0036:0
                                GO TO EXIT;                                      %112-          14269740 C  0103:0036:0
                            END;                                                 %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
                                                                                 %112-          14269840 C  0103:0037:2
    EXIT:                                                                        %112-          14269860 C  0103:0040:1
      END                                                                        %112-          14269880 C  0103:0041:2
    UNTIL                                                                        %112-          14269900 C  0103:0041:2
      ELCLASS ≠ COMMA;                                                           %112-          14269920 C  0103:0041:2
    STOPENTRY := STOPGSP := FALSE;                                               %112-          14269940 C  0103:0042:0
  END;                                                                           %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
                                                                                               START OF SEGMENT ********** 104
      LABEL START2;                                                                              14273000 T  0104:0000:0
    BOOLEAN FWDTOG; COMMENT THIS TOGGLE IS THE FORWARD DEC INDICATOR;                            14274000 T  0104:0000:0
        IF NOT SPECTOG THEN FUNCTOG←FALSE;                                                       14275000 T  0104:0000:0
   FWDTOG←FALSE  ;                                                                             14276000 T  0104:0001:3
              MAXSTACKO← MAXSTACK;                                                             14277000 T  0104:0002:1
   IF G←GTA1[J←J-1]=STREAMV                                                                    14278000 T  0104:0003:2
    THEN                                                                                         14279000 T  0104:0005:2
      BEGIN STREAMTOG←TRUE;                                                                      14280000 T  0104:0005:3
        IF G←GTA1[J←J-1]=0 THEN TYPEV←STRPROCID                                                  14281000 T  0104:0007:2
                          ELSE                                                                   14282000 T  0104:0010:0
                           BEGIN                                                                 14283000 T  0104:0010:1
                            IF TYPEV←PROCID +G>INTSTRPROCID OR                                   14284000 T  0104:0011:2
                               TYPEV <BOOSTRPROCID                                               14285000 T  0104:0013:2
                             THEN FLAG(004);                                                     14286000 T  0104:0013:2
                        IF NOT SPECTOG THEN                                                      14287000 T  0104:0015:2
                             FUNCTOG←TRUE;                                                       14288000 T  0104:0015:3
                             CHKSOB                                                              14289000 T  0104:0017:2
    END                      END                                                                 14290000 T  0104:0017:2
    ELSE                                                                                         14291000 T  0104:0017:3
   IF G=SAVEV OR G=0 THEN TYPEV←PROCID                                                           14292000 T  0104:0017:3
             ELSE                                                                                14293000 T  0104:0020:1
              IF TYPEV←REALSTRPROCID+G<BOOPROCID OR TYPEV>INTPROCID                              14294000 T  0104:0021:2
                THEN FLAG(005)                                                                   14295000 T  0104:0023:3
          ELSE BEGIN FUNCTOG←TRUE;G←GTA1[J←J-1];                                                 14296000 T  0104:0025:2
                     END;                                                                        14297000 T  0104:0028:1
   IF NOT STREAMTOG THEN SEGMENTSTART(G=SAVEV);                                                  14298000 T  0104:0028:1
   SAF ← G=SAVEV;                                                                                14299000 T  0104:0031:2
                                                                                                 14300000 T  0104:0032:0
                                                                                                 14301000 T  0104:0032:0
                                                                                                 14302000 T  0104:0032:0
   MODE←MODE+1;                                                                                  14303000 T  0104:0032:0
   LO←PROINFO;                                                                                   14304000 T  0104:0033:3
   SCATTERLBAT;                                                                                  14305000 T  0104:0034:0
COMMENT CHECK TO SEE IF DECLARED FORWARD PREVIOUSLY                                ;             14306000 T  0104:0034:1
   IF LEVELF=LEVEL                                                                               14307000 T  0104:0034:1
     THEN                                                                                        14308000 T  0104:0035:2
       BEGIN                                                                                     14309000 T  0104:0035:3
```

```
            IF G←TAKE(LINKF+1)≥0                                                                14310000 T  0104:0036:0
              THEN FLAG(006);                                                                   14311000 T  0104:0037:3
            FWDTOG←TRUE;                                                                        14312000 T  0104:0039:3
          XMARK(DECLREF); % PROCEDURE DECLARED FORWARD. MARK LAST      %110-                  14312100 C  0104:0040:0
                          % XREF ENTRY AS A DECLARATION,              %110-                  14312101 C  0104:0044:1
            PROAD←ADDRSF;                                                                       14313000 T  0104:0044:1
            PROINFO←ELBAT[I];MARK←LINKF+INCRF;STEPIT                                            14314000 T  0104:0045:2
                 ;PUT(-G,LINKF+1);                                                              14315000 T  0104:0047:3
          END                                                                                   14316000 T  0104:0049:3
        ELSE                                                                                    14317000 T  0104:0049:3
          BEGIN STOPENTRY←TRUE; P2←TRUE;                                                        14318000 T  0104:0049:3
            STOPGSP←LEVEL>1 AND STREAMTOG;                                                      14318500 T  0104:0051:3
            ENTRY(TYPEV);    MARK←NEXTINFO;PUTNBUMP(0);                                         14319000 T  0104:0053:3
            STOPGSP←FALSE;                                                                      14319500 T  0104:0055:3
            PROINFO←TAKE(LASTINFO)& LASTINFO[35:35:13];PROAD←ADDRSF;                            14320000 T  0104:0056:1
         P2←STOPENTRY←FALSE                                                                     14321000 T  0104:0059:3
          END;                                                                                  14322000 T  0104:0059:3
                    PJ←0; LEVEL←LEVEL+1;                                                        14323000 T  0104:0060:1
           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:PINFOO←NEXTINFO;                                                                                       14375000 T  0104:0114:1
START2: END;                                                                                                  14376000 T  0104:0115:3
                                                                                                    104 IS  117 LONG, NEXT SEG   98
       IF SPECTOG OR STREAMTOG                                                                                14377000 T  0098:0128:0
          THEN                                                                                                14378000 T  0098:0128:0
                GO TO START;                                                                                  14379000 T  0098:0128:1
COMMENT IF SPECTOG IS ON THEN THE BLOCK WILL PROCESS THE SPECIFICATION                                        14380000 T  0098:0129:3
        PART SIMILARY TO DECLARATIONS WITH A FEW NECESSARY VARIATIONS;                                        14381000 T  0098:0129:3
HF:                                                                                                           14382000 T  0098:0129:3
BEGIN                                                                                                         14383000 T  0098:0130:0
  LABEL START,STOP;                                                                                           14384000 T  0098:0130:0
                                                                                                  START OF SEGMENT **********  105
      DEFINE  TESTLEV = LEVEL>2 #;                                                                            14384100 T  0105:0000:0
IF STREAMTOG                                                                                                  14385000 T  0105:0000:0
  THEN BEGIN                                                                                                  14386000 T  0105:0000:0
   IF TESTLEV THEN JUMPCHKNX ELSE SEGMENTSTART(TRUE);PJ←P;                                                    14387000 T  0105:0000:1
                 PTOG←FALSE;                                                                                  14388000 T  0105:0004:1
      PUT(TAKE(GIT(PROINFO))&LL[28:36:12],GIT(PROINFO));                                                      14388100 T  0105:0005:2
  IF TESTLEV THEN BEGIN EMITO(584);          END;                                                           14389000 T  0105:0008:1
          IF STEPI=BEGINV                                                                                     14393000 T  0105:0010:1
            THEN                                                                                              14394000 T  0105:0011:2
               BEGIN                                                                                          14395000 T  0105:0011:3
                 WHILE STEPI=DECLARATORS OR ELCLASS=LOCALV                                                    14396000 T  0105:0012:0
                    DO                                                                                        14397000 T  0105:0014:0
                     BEGIN                                                                                    14398000 T  0105:0015:3
                       IF ELBAT[I].ADDRESS=LABELV                                                             14399000 T  0105:0015:3
                        THEN                                                                                  14400000 T  0105:0016:1
                           BEGIN                                                                              14401000 T  0105:0017:2
                             STOPDEFINE←STOPGSP←STOPENTRY←TRUE;                                               14402000 T  0105:0017:3
  DO BEGIN STOPDEFINE←TRUE;STEPIT;ENTRY(STLABID);PUTNBUMP(0) END UNTIL                                        14403000 T  0105:0019:2
                ELCLASS≠COMMA;STOPGSP←STOPENTRY←FALSE                                                         14404000 T  0105:0022:1
                                    END                                                                       14405000 T  0105:0024:0
                              ELSE                                                                            14406000 T  0105:0025:2
                                BEGIN                                                                         14407000 T  0105:0025:2
                                        I←I+1;                                                                14408000 T  0105:0025:3
                                  ENTRY(LOCLID)                                                               14409000 T  0105:0027:2
                                END                                                                           14410000 T  0105:0027:2
                             END;                                                                             14411000 T  0105:0027:3
      IF FUNCTOG  THEN                                                                                        14411100 T  0105:0028:0
```

```
                    PUT((Z←TAKE(PROINFO))&LOCLID[2:41:7] &                                  14411200 T  0105:0028:1
                              (PJ+2+REAL(TESTLEV))[16:37:11],PROINFO);                      14411300 T  0105:0031:3
                         COMPOUNDTAIL                                                       14412000 T  0105:0034:1
                       END                                                                  14413000 T  0105:0034:1
                     ELSE                                                                   14414000 T  0105:0035:2
               BEGIN                                                                        14415000 T  0105:0035:2
               IF   FUNCTOG  THEN                                                           14415100 T  0105:0035:3
                    PUT( Z←TAKE(PROINFO))& LOCLID[2:41:7]&                                  14415200 T  0105:0035:3
                              (PJ+2+REAL(TESTLEV))[16:37:11],PROINFO);                      14415300 T  0105:0038:1
               STREAMSTMT;                                                                  14415400 T  0105:0041:3
               END;                                                                         14415500 T  0105:0042:0
COMMENT THE FOLLOWING BLOCK CONSTITUTES THE STREAM PROCEDURE PURGE;                         14416000 T  0105:0042:0
               BEGIN                                                                        14417000 T  0105:0042:0
               REAL NLOC,NLAB;                                                              14418000 T  0105:0042:0
                                                                                  START OF SEGMENT **********  106
               DEFINE SES=18#,SED=6#,TRW=5#;                                                14419000 T  0106:0000:0
               DEFINE LOC=[36:12]#,LASTGT=[24:12]#;                                         14420000 T  0106:0000:0
               J←  LASTINFO;                                                                14421000 T  0106:0000:0
                    NLOC←NLAB←0;                                                            14422000 T  0106:0000:1
               DO                                                                           14423000 T  0106:0002:0
               BEGIN                                                                        14424000 T  0106:0002:0
               IF(GT1←TAKE(J)).CLASS=LOCLID THEN                                            14425000 T  0106:0002:0
                    BEGIN                                                                   14426000 T  0106:0004:0
                    IF BOOLEAN(GT1.FORMAL) THEN                                             14427000 T  0106:0004:1
                         BEGIN                                                              14428000 T  0106:0005:3
                         IF GT1<0 THEN                                                      14429000 T  0106:0006:0
                         PUT(TAKE(GT2←MARK+P-GT1.ADDRESS+1)&FILEID[2:41:7]                  14430000 T  0106:0006:1
                              ,GT2);                                                        14431000 T  0106:0011:3
                         END                                                                14432000 T  0106:0012:0
                    ELSE  NLOC←NLOC+1;                                                      14433000 T  0106:0012:0
                    END                                                                     14434000 T  0106:0014:0
               ELSE                                                                         14435000 T  0106:0014:0
                    BEGIN                                                                   14436000 T  0106:0014:0
                    IF  GT1.ADDRESS≠0 THEN NLAB←NLAB+1;                                     14437000 T  0106:0014:1
                    IF(GT3←TAKE(GT(J))).LASTGT≠0 AND GT3.LOC =0 THEN                        14438000 T  0106:0017:3
                         BEGIN                                                              14439000 T  0106:0021:3
                         MOVE(9,INFO[0,J],ACCUM[0]);                                        14440000 T  0106:0022:0
                         Q←ACCUM[1];                                                        14441000 T  0106:0024:1
                         FLAG(267);                                                         14442000 T  0106:0025:3
                         ERRORTOG←TRUE;                                                     14443000 T  0106:0026:1
                         END;                                                               14444000 T  0106:0027:2
                    END;                                                                    14445000 T  0106:0027:2
               XREFDUMP(J); % DUMP XREF INFO                                        %110-  14445100 C  0106:0027:2
               G←(GT2←TAKE(J+1)).PURPT;                                                     14446000 T  0106:0029:2
               IF  GT1.[2:8] ≠ STLABID×2+1  THEN                                            14447000 T  0106:0032:0
                    STACKHEAD[(0&GT2[12:12:36])MOD 125]←TAKE(J).LINK;                       14448000 T  0106:0034:0
               END UNTIL J←J-G≤1;                                                           14449000 T  0106:0038:1
                                                                                            14450000 T  0106:0040:1
               IF   TESTLEV   THEN BEGIN EMITC(1,0); EMITO(BFW) END                         14451000 T  0106:0040:1
                              ELSE  EMIT(0);                                                14451100 T  0106:0043:3
PUT(TAKE(MARK)&NLOC[1:42:6]&L[16:36:12]&P[40:40:8],MARK);                                   14451200 T  0106:0045:2
               IF FUNCTOG THEN                                                              14452000 T  0106:0049:3
                    PUT(Z, PROINFO);                                                        14457000 T  0106:0049:3
               STREAMWORDS;                                                                 14460000 T  0106:0051:2
               STREAMTOG←FALSE;                                                             14461000 T  0106:0051:3
          IF NOT TESTLEV THEN BEGIN PROGDESCBLDR(PROAD,TRUE,(L+3)DIV 4,CHAR);               14461100 T  0106:0052:1
                                    SEGMENT((L+3)DIV 4,PROINFO,ADDRESS);                    14461200 T  0106:0056:0
```

```
                                        RIGHT(L); L←0;                                                  14461300 T  0106:0058:1
                        END;                                                                            14461400 T  0106:0060:0
    IF LISTER AND FORMATOG THEN SPACEITDOWN;                                                            14461500 T  0106:0060:0
    END;                                                                                                14462000 T  0106:0066:0
                                                                                                106 IS    67 LONG, NEXT SEG  105
        LASTINFO←LASTINFOT;NEXTINFO←MARK+P+1;                                                           14463000 T  0105:0043:2
    END                                                                                                 14464000 T  0105:0045:3
ELSE                                                                                                    14465000 T  0105:0045:3
    BEGIN                                                                                               14466000 T  0105:0045:3
 IF STEPI=FORWARDV                                                                                      14467000 T  0105:0046:0
   THEN                                                                                                 14468000 T  0105:0046:1
    BEGIN                                                                                               14469000 T  0105:0047:2
      XREFIT(PROINFO,0,FORWARDREF); % WE NEED THIS SO WE CAN FIND                                       14469100 C  0105:0047:3
                                    % THE FORWARD DECL. DURING XREF                                    14469101 C  0105:0050:0
      PUT(-TAKE(G←PROINFO,LINK+1),G);                                                                   14470000 T  0105:0050:0
          PURGE(PINFOO);                                                                                14471000 T  0105:0053:2
      STEPIT                                                                                            14472000 T  0105:0054:0
    END                                                                                                 14473000 T  0105:0054:0
   ELSE                                                                                                 14474000 T  0105:0054:1
    BEGIN                                                                                               14475000 T  0105:0054:1
 PROADO←PROAD;                                                                                          14476000 T  0105:0055:2
     TSUBLEVEL←SUBLEVEL;SUBLEVEL←LEVEL ;STACKCTRO←STACKCTR;                                             14477000 T  0105:0055:3
IF MODE=1 THEN FRSTLEVEL←LEVEL;STACKCTR←513+REAL(FUNCTOG);                                              14478000 T  0105:0058:0
      IF ELCLASS = BEGINV THEN                                                                          14479000 T  0105:0061:2
 BEGIN                                                                                                  14481000 T  0105:0062:0
         CALLINFO←(CALLX←CALLX+1)+1;                                                                    14481100 T  0105:0062:1
         NESTCTR←STACKCTR;                                                                              14481200 T  0105:0064:1
                  BLOCK(TRUE)                                                                           14482000 T  0105:0065:3
                ; PURGE(PINFOO);                                                                        14483000 T  0105:0065:3
                IF NESTOG THEN                                                                          14483100 T  0105:0067:2
                BEGIN GT1←TAKE(PROINFO),ADDRESS;                                                        14483200 T  0105:0067:3
                     NESTPRT[GT1]←0&PROINFO[35:35:13]&CALLINFO[22:35:13];                               14483300 T  0105:0070:0
                     CALL[CALLINFO-1]←(TAKE(GIT(PROINFO))+NESTCTR-511)&                                 14483400 T  0105:0073:3
                          CALLX[22:35:13];                                                              14483500 T  0105:0079:2
                END;                                                                                    14483600 T  0105:0080:1
                L←0;                                                                                    14483700 T  0105:0080:1
                  GO TO STOP END;                                                                       14484000 T  0105:0081:3
                   BEGIN                                                                                14485000 T  0105:0082:0
FLAG(052);                                                                                              14486000 T  0105:0082:0
                    RELAD←L  ;                                                                          14487000 T  0105:0082:1
          STMT;                                                                                         14488000 T  0105:0083:3
               HITEOAP(FALSE,RELAD,PINFOO,PROAD);                                                       14489000 T  0105:0084:0
                END;                                                                                    14490000 T  0105:0085:3
  STOP:                                                                                                 14491000 T  0105:0085:3
      SUBLEVEL←TSUBLEVEL;                                                                               14492000 T  0105:0086:0
      STACKCTR←STACKCTRO;                                                                               14493000 T  0105:0086:1
      IF LISTER AND FORMATOG THEN SPACEITDOWN;                                                          14493500 T  0105:0087:3
        END;                                                                                            14494000 T  0105:0093:2
        END;                                                                                            14495000 T  0105:0093:2
     PROINFO←LO;                                                                                        14496000 T  0105:0093:2
     IF JUMPCTR=LEVEL                                                                                   14497000 T  0105:0094:0
        THEN                                                                                            14498000 T  0105:0094:0
                JUMPCTR←LEVEL-1;                                                                        14499000 T  0105:0094:1
           LEVEL←LEVEL-1;                                                                               14500000 T  0105:0096:1
         MODE←MODE-1;                                                                                   14501000 T  0105:0097:3
        MAXSTACK←MAXSTACKO;                                                                             14502000 T  0105:0099:2
START:END;                                                                                              14503000 T  0105:0099:3
```

```
                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
                        BEGIN
                         STMT;
                         IF ELCLASS←TABLE(I+1)=DECLARATORS
                           THEN
                             BEGIN
                               ELBAT[I].CLASS←SEMICOLON;
                               BEGINCTR←BEGINCTR+1;
                               GO TO START
                             END
                           ELSE
                             COMPOUNDTAIL
                        END;
            FUNCTOG←FUNCTOG0;
            IF SOP THEN HTTEOAP(FALSE,FIRSTX,NINF00,BLKAD)
                ELSE BEGIN IF NESTOG THEN SORTNEST; PURGE(NINF00); END;
            SEGMENT((L+3)DIV 4,PROAD0);
                IF LEVEL>1 THEN RIGHT(L);
            IF LEVEL ← LEVEL-1 = 0 THEN CONSTANTCLEAN;

            AJUMP←AJUMP0;

               FIRSTX←FIRSTX0;
            SAVEL←SAVEL0;
            STACKCTR←STACKCTR0;


        END BLOCK;

         COMMENT THIS SECTION CONTAINS THE VARIABLE ROUTINE AND ITS SIDEKICKS;
```

```
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
      14514000 T  0098:0144:0
      14515000 T  0098:0144:1
      14516000 T  0098:0145:2
      14517000 T  0098:0146:1
      14518000 T  0098:0147:2
      14519000 T  0098:0147:3
      14520000 T  0098:0150:0
      14521000 T  0098:0151:3
      14522000 T  0098:0152:0
      14523000 T  0098:0152:0
      14524000 T  0098:0152:0
      14525000 T  0098:0152:1
      14599000 T  0098:0153:2
      14600000 T  0098:0153:3
      14601000 T  0098:0155:3
      14602000 T  0098:0159:2
      14603000 T  0098:0161:2
      14604000 T  0098:0163:2
      14605000 T  0098:0165:3
      14606000 T  0098:0165:3
      14607000 T  0098:0166:1
      14608000 T  0098:0166:1
      14609000 T  0098:0167:2
      14610000 T  0098:0168:0
      14611000 T  0098:0168:1
      14612000 T  0098:0168:1
      14613000 T  0098:0168:1
98 IS  176 LONG, NEXT SEG    3
      15000000 T  0003:0628:1
      15001000 T  0003:0628:1
      15002000 T  0003:0628:1
      15003000 T  0003:0628:1
      15004000 T  0003:0628:1
      15005000 T  0003:0628:1
      15006000 T  0003:0628:1
      15007000 T  0003:0628:1
      15008000 T  0003:0628:1
      15009000 T  0003:0628:1
      15012000 T  0003:0628:1
      15013000 T  0003:0628:1
      15014000 T  0003:0628:1
      15015000 T  0003:0628:1
      15016000 T  0003:0628:1
      15017000 T  0003:0628:1
      15018000 T  0003:0628:1
```

```
                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;
                              T1   ,  COMMENT 1ST INTEGER OF PARTIAL WORD SYNTAX;
                              T2   ,  COMMENT 2ND INTEGER OF PARTIAL WORD SYNTAX;
```

```
15019000 T   0003:0628:1
15020000 T   0003:0628:1
15021000 T   0003:0628:1
15022000 T   0003:0628:1
15023000 T   0003:0628:1
15024000 T   0003:0628:1
15025000 T   0003:0628:1
15026000 T   0003:0628:1
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
```

```
                J   ;  COMMENT SUBSCRIPT COUNTER ;                                          15075000 T  0107:0000:0
          REAL REMEMBERSEQNO; % REMEMBERS SEQUENCE NUMBER OF VARIABLE %110-                15075550 C  0107:0000:0
                              % ON LEFT HAND SIDE OF ASSIGNMENT SO WE %110-                15075551 C  0107:0000:0
                              % CAN XREF IT CORRECTLY.                %110-                15075552 C  0107:0000:0
                LABEL EXIT, NEXT, LASS;                                %114-                15076000 P  0107:0000:0
                DEFINE                                                 %114-                15076100 P  0107:0000:0
                  FORMALNAME = [9:2] = 2#,                             %114-                15076110 C  0107:0000:0
                  LONGID = NAMEID#,                                    %114-                15076120 C  0107:0000:0
                  PARTIALWORD = (T1 ≠ 0)#,                             %114-                15076130 C  0107:0000:0
                  ERREXIT(NUM) = BEGIN ERR(NUM); GO TO EXIT; END#;     %114-                15076140 C  0107:0000:0
                BOOLEAN                                                %114-                15076200 P  0107:0000:0
                  SPCLMON,                                             %114-                15076300 C  0107:0000:0
                  DUPIT;  % TRUE IF WE ARE DOING UPDATE TYPE ASSIGN-   %114-                15076400 C  0107:0000:0
                          % MENT, I.E., I := * + 1..                    %114-                15076500 C  0107:0000:0
                TALL←ELBAT[I] ;                                                              15077000 T  0107:0000:0
                IF   ELCLASS ≤ INTPROCID  THEN                                               15078000 T  0107:0001:2
                   BEGIN                                                                     15079000 T  0107:0001:3
                     IF TALL.LINK ≠PROINFO.LINK THEN                                         15080000 T  0107:0002:0
                       BEGIN ERR(211); GO TO EXIT END;                                       15081000 T  0107:0004:0
COMMENT  211  VARIABLE-FUNCTION IDENTIFIER USED OUTSIDE OF ITS SCOPE*;                       15082000 T  0107:0005:3
                TALL←TALL &(ELCLASS+4)[2:41:7] & 513[16:37:11];                              15083000 T  0107:0005:3
                   END                                                                       15084000 T  0107:0009:2
                ELSE  CHECKER(TALL);                                                         15085000 T  0107:0009:2
                REMEMBERSEQNO := CARDNUMBER;                           %110-                15085100 C  0107:0010:0
                IF TALL.CLASS ≤ INTID THEN % SIMPLE VAR OR FORMAL NAME                       15085200 C  0107:0011:2
                  BEGIN                                                %114-                15085400 C  0107:0012:0
                    IF STEPI ≠ ASSIGNOP THEN                           %114-                15085600 C  0107:0012:1
                      IF P1 = FL THEN                                  %114-                15085800 C  0107:0013:3
                  BEGIN                                                %114-                15086000 P  0107:0015:2
                    IF ELCLASS < AMPERSAND THEN                        %114-                15086200 C  0107:0015:3
                      EMITN(TALL.ADDRESS)                              %114-                15086400 C  0107:0016:0
                    ELSE                                               %114-                15086600 C  0107:0017:3
                      EMITV(TALL.ADDRESS);                             %114-                15086800 C  0107:0018:0
                    GO TO EXIT;                                        %114-                15087000 P  0107:0019:3
                  END                                                  %114-                15087200 C  0107:0020:0
                ELSE                                                   %114-                15087400 C  0107:0020:0
                  IF ELCLASS = PERIOD THEN % PARTIAL WORD              %114-                15087600 C  0107:0020:0
                    IF DOTSYNTAX(T1,T2) THEN % ERROR                   %114-                15087800 C  0107:0021:3
                      GO TO EXIT                                       %114-                15088000 P  0107:0023:2
                    ELSE                                               %114-                15088200 C  0107:0023:2
                      STEPIT;                                          %114-                15088400 C  0107:0023:2
              IF ELCLASS = ASSIGNOP THEN                               %114-                15088600 C  0107:0024:0
                BEGIN                                                  %114-                15088800 C  0107:0025:2
                  STACKCT := 1;                                        %114-                15089000 P  0107:0025:3
                  IF PARTIALWORD THEN % MAKE SURE LEFT-MOST            %114-                15089200 C  0107:0026:0
                    BEGIN                                              %114-                15089400 C  0107:0027:2
                      IF P1 ≠ FS THEN % NOT LEFT-MOST                  %114-                15089600 C  0107:0027:3
                        ERREXIT(201);                                  %114-                15089800 C  0107:0028:0
                      XREFIT(TALL,REMEMBERSEQNO,ASSIGNREF);            %114-                15090600 C  0107:0030:0
                    END                                                %114-                15090800 C  0107:0032:1
                  ELSE                                                 %114-                15091000 P  0107:0032:1
                    XMARK(ASSIGNREF);                                  %114-                15091200 C  0107:0032:1
                  IF TABLE(I+1) = ASTRISK THEN % MIGHT BE UPDATE                             15091400 C  0107:0037:2
                    IF (DUPIT:=(TABLE(I+2) ≥ EQVOP AND TABLE(I+2)                            15091600 C  0107:0039:2
                        ≤ MULOP) OR TABLE(I+2) = AMPERSAND) THEN                             15091800 C  0107:0042:0
                      STEPIT; % STEP OVER ASTERISK                     %114-                15092000 P  0107:0045:3
                  IF TALL.FORMALNAME THEN % FORMAL PARAMETER           %114-                15092200 C  0107:0046:1
```

```
  BEGIN                                                          %114-      15092400 C  0107:0048:0
    EMITN(TALL.ADDRESS);                                         %114-      15092600 C  0107:0048:1
    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  0107:0088:0
   6. V[R].[S:L] ←  ,ASSIGNMENT TO PARTIAL WORD,LEFT-MOST.                 15136000 T  0107:0088:0
R IS A K-ORDER SUBSCRIPT LIST,I.E. R= R1,R2,...,RK.                       15137000 T  0107:0088:0
IN THE CASE OF NO MONITORING ON V, THE FOLLOWING CODE                     15138000 T  0107:0088:0
IS EMITTED FOR THE ABOVE CASES:                                           15139000 T  0107:0088:0
      1. CASE #1 IS A SPECIAL CASE OF #2,NAMELY,SINGLE                    15140000 T  0107:0088:0
         DIMENSION. THE CODE EMITTED IS:                                  15141000 T  0107:0088:0
            VL,LOD .                                                      15142000 T  0107:0088:0
         EXECUTION: PLACES ARRAY DESCRIPTER IN REG A.                     15143000 T  0107:0088:0
      2. THIS CODE IS BASIC TO THE SUBSCRIPTION PROCESS.                  15144000 T  0107:0088:0
         EACH SUBSCRIPT GENERATES THE FOLLOWING SEQUENCE                  15145000 T  0107:0088:0
         OF CODE:                                                         15146000 T  0107:0088:0
            AEXP,L*,IF FIRST SUBSCRIPT THEN VN ELSE CDC                   15147000 T  0107:0088:0
              ,LOD.                                                       15148000 T  0107:0088:0
         FOR A K-ORDER SUBSCRIPTION,K-1 SEQUENCE ARE                      15149000 T  0107:0088:0
         PRODUCED. THE AEXP IN EACH SEQUENCE REFERS TO                    15150000 T  0107:0088:0
         THE CODE PRODUCED BY THE ARITHMETIC EXPRESSION                   15151000 T  0107:0088:0
         PROCEDURE FOR THE ACTUAL SUBSCRIPT EXPRESSIONS,                  15152000 T  0107:0088:0
         L* REFERS TO THE CODE PRODUCED FOR SUBTRACTING                   15153000 T  0107:0088:0
         NON-ZERO LOWER BOUNDS FROM THE SUBSCRIPT                         15154000 T  0107:0088:0
         EXPRESSION(L* YIELDS NO CODE FOR ZERO BOUNDS).                   15155000 T  0107:0088:0
         EXECUTION: PLACES ARRAY ROW DESCRIPTOR IN REG A                  15156000 T  0107:0088:0
                    . THE SPECIFIC ROW DEPENDS UPON THE                   15157000 T  0107:0088:0
                    VALUES OF THE K-1 SUBSCRIPTS.                         15158000 T  0107:0088:0
FOR THE REMAINING CASES,                                                  15159000 T  0107:0088:0
         SEQUENCES OF CODE ARE EMITED AS IN CASE #2.                      15160000 T  0107:0088:0
         HOWEVER,THE ACTUAL SEQUENCES ARE:                                15161000 T  0107:0088:0
          ONE SEQUENCE ,(AEXP,L*),FOR THE 1ST SUBSCRIPT.                  15162000 T  0107:0088:0
          K-1 SEQUENCES,(IF FIRST SUBSCRIPT THEN VN                       15163000 T  0107:0088:0
          ELSE CDC,LOD,AEXP,L*), FOR THE REMAINING                        15164000 T  0107:0088:0
          SUBSCRIPTS,IF K>1.                                              15165000 T  0107:0088:0
         AT THIS POINT, CASES #3-6 ARE DIFFERENTIATED                     15166000 T  0107:0088:0
         AND ADDITION CODE,PARTICULAR TO EACH CASE,IS                     15167000 T  0107:0088:0
         EMITTED.                                                         15168000 T  0107:0088:0
      3. ADD THE SEQUENCE:                                                15169000 T  0107:0088:0
           IF FIRST SUBSCRIPT THEN VV ELSE COC.                           15170000 T  0107:0088:0
         EXECUTION: THE ARRAY ELEMENT IS PUT IN REG A.                    15171000 T  0107:0088:0
      4. ADD THE SEQUENCE:                                                15172000 T  0107:0088:0
           IF FIRST SUBSCRIPT THEN VV ELSE COC,ZEROL,                     15173000 T  0107:0088:0
             XCH,T.                                                       15174000 T  0107:0088:0
      5. ADD THE SEQUENCE:                                                15175000 T  0107:0088:0
           IF FIRST SUBSCRIPT THEN VN ELSE CDC,EXP,                       15176000 T  0107:0088:0
             XCH,←.                                                       15177000 T  0107:0088:0
      6. ADD THE SEQUENCE:                                                15178000 T  0107:0088:0
           IF FIRST SUBSCRIPT THEN VN ELSE CDC,DUP,LOD,                   15179000 T  0107:0088:0
             EXP,T, XCH,←.                                                15180000 T  0107:0088:0
    EXP,T,←,ZEROL,ETC. HAVE SAME MEANINGS AS DEFINED IN                   15181000 T  0107:0088:0
    SIMPLE VARIABLE BLOCK.                                    ;           15182000 T  0107:0088:0
BEGIN                                                                     15183000 T  0107:0088:0
  IF STEP1 ≠ LFTBRKET THEN % ARRAY ITEM NOT FOLLOWED BY          %114-    15183100 C  0107:0088:1
    BEGIN                  % A SUBSCRIPT                         %114-    15183200 C  0107:0089:3
      IF ELCLASS = PERIOD THEN                                   %114-    15183300 C  0107:0090:0
        IF DOTSYNTAX(T1,T2) THEN % ERROR IN PARTIAL WORD         %114-    15183400 C  0107:0091:2
          GO TO EXIT                                             %114-    15183500 C  0107:0092:1
        ELSE                                                     %114-    15183600 C  0107:0092:1
```

```
    STEPIT;                                                          %114-
IF ELCLASS = ASSIGNOP THEN                                           %114-
  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.          *;
```

```
15183700 C  0107:0092:1
15183800 C  0107:0093:3
15183900 C  0107:0094:1
15184000 P  0107:0095:2
15184100 P  0107:0095:3
15184200 P  0107:0096:0
15184300 P  0107:0097:2
15184400 P  0107:0098:1
15184500 C  0107:0101:2
15184600 C  0107:0101:2
15184700 C  0107:0101:2
15184750 C  0107:0106:0
15184800 C  0107:0107:3
15184850 C  0107:0110:1
15185000 C  0107:0114:1
15185100 C  0107:0115:3
15185200 C  0107:0116:1
15185300 C  0107:0118:1
15185400 C  0107:0120:0
15185500 C  0107:0120:1
15185600 C  0107:0122:0
15185700 C  0107:0123:3
15185800 C  0107:0124:0
15185900 C  0107:0124:1
15186000 C  0107:0126:0
15186100 C  0107:0126:0
15186200 C  0107:0128:0
15186300 C  0107:0128:1
15186400 C  0107:0129:2
15186500 C  0107:0129:3
15186600 C  0107:0130:0
15186700 C  0107:0131:2
15186800 C  0107:0132:1
15186900 C  0107:0135:3
15187000 C  0107:0136:0
15187100 C  0107:0137:2
15187200 C  0107:0137:2
15187300 C  0107:0137:2
15187400 C  0107:0138:1
15187500 C  0107:0139:2
15187600 C  0107:0139:2
15234000 T  0107:0139:2
15234500 T  0107:0140:0
15235000 T  0107:0140:1
15253000 T  0107:0140:1
15254000 T  0107:0142:0
15255000 T  0107:0142:1
15256000 T  0107:0144:0
15257000 T  0107:0146:0
15258000 T  0107:0146:0
15259000 T  0107:0146:0
15260000 T  0107:0146:0
15261000 T  0107:0147:2
15262000 T  0107:0148:1
15263000 T  0107:0148:1
15264000 T  0107:0148:1
15265000 T  0107:0148:1
```

```
                    IF  J=0  THEN                                                  15266000 T  0107:0148:1
                             EMITPAIR(TALL.ADDRESS,LOD);                           15267000 T  0107:0149:3
                    STEPIT;                                                        15274000 T  0107:0151:3
                    GO TO EXIT;                                                    15275000 T  0107:0152:0
                  END OF ROW DESIGNATOR PORTION  ;                                 15276000 T  0107:0152:1
IF ELCLASS=LITNO AND ELBAT[I].ADDRESS=0 AND TABLE(I+1)=RTBRKET                     15276010 T  0107:0152:1
    AND TALL.CLASS≥NAMEID THEN                                                     15276020 T  0107:0156:0
    BEGIN                                                                          15276030 T  0107:0158:1
    I←I+1;                                                                         15276040 T  0107:0159:2
    IF STEPI=ASSIGNOP THEN BEGIN                                                   15276050 T  0107:0160:0
LASS:    IF T1≠0 THEN EMITV(TALL.ADDRESS);                                         15276060 T  0107:0161:3
         STEPIT;  AEXP;  EMITD(48-T2,T1,T2);                                       15276070 T  0107:0164:1
         EMITN(TALL.ADDRESS);                                                      15276080 T  0107:0167:2
         EMITO(IF TALL.CLASS≠NAMEID THEN                                           15276090 T  0107:0168:1
              IF P1=FS THEN ISD ELSE ISN ELSE                                      15276100 T  0107:0170:0
              IF P1=FS THEN STD ELSE SND);                                         15276110 T  0107:0173:2
    STACKCT ← 0;                                                                   15276115 T  0107:0175:3
         GO TO EXIT      END                                                       15276120 T  0107:0176:1
    ELSE                                                                           15276130 T  0107:0177:2
    IF ELCLASS = PERIOD THEN BEGIN                                                 15276140 T  0107:0177:2
         IF DOTSYNTAX(T1,T2) THEN GO TO EXIT;                                      15276150 T  0107:0178:1
         IF STEPI = ASSIGNOP THEN IF P1=FS THEN GO TO LASS                         15276160 T  0107:0180:1
         ELSE BEGIN ERR(209); GO EXIT END;                                         15276170 T  0107:0182:1
                    END;                                                           15276180 T  0107:0184:1
    IF P1=FS THEN BEGIN ERR(210); GO EXIT END;                                     15276190 T  0107:0184:1
                                                                                   15276200 T  0107:0187:2
    EMITI(IF P1=FL THEN TALL ELSE TALL&REALID[2:41:7],T1,T2);                      15276210 T  0107:0187:2
                    GO TO EXIT;                                                    15276220 T  0107:0191:2
                    END;                                                           15276230 T  0107:0191:2
                    AEXP;                                                          15276240 T  0107:0191:3
                    STACKCT ← 1;                                                   15277000 T  0107:0191:3
                    J ← J + 1;                                                     15278000 T  0107:0192:0
                    IF  ELCLASS = COMMA THEN                                       15280000 T  0107:0193:2
                      BEGIN                                                        15287000 T  0107:0194:0
COMMENT  *****        MONITOR FUNCTION M4 GOES HERE ;                              15288000 T  0107:0195:2
    IF J = 1 THEN EMITV(TALL.ADDRESS) ELSE EMITO(COC);                             15289000 T  0107:0195:3
                                                                                   15290000 T  0107:0195:3
                      GO TO NEXT;                                                  15291000 T  0107:0199:2
                    END OF SUBSCRIPT COMMA HANDLER ;                               15292000 T  0107:0199:2
                  IF ELCLASS ≠ RTBRKET THEN BEGIN ERR(206);GO EXIT END;            15293000 T  0107:0199:3
COMMENT  206  VARIABLE- MISSING RIGHT BRACKET ON SUBSCRIPTED VARIABLE*;            15294000 T  0107:0199:3
                  GT1←IF TALL.CLASS≥NAMEID THEN 1 ELSE TALL.INCR;                  15295000 T  0107:0202:0
                  IF J≠GT1  THEN                                                   15295100 T  0107:0202:0
                                BEGIN ERR(208);GO TO EXIT END;                     15296000 T  0107:0206:0
COMMENT  208  VARIABLE- NUMBER OF SUBSCRIPTS DOES NOT MATCH WITH      *           15297000 T  0107:0206:1
                  ARRAY DECLARATION.                                   *;          15298000 T  0107:0208:1
                  IF STEPI = PERIOD THEN % PARTIAL WORD                %114-       15299000 T  0107:0208:1
                    IF DOTSYNTAX(T1,T2) THEN % ERROR                   %114-       15300000 P  0107:0208:1
                      GO TO EXIT                                       %114-       15300100 C  0107:0209:3
                    ELSE                                               %114-       15300200 C  0107:0211:2
                      STEPIT;                                          %114-       15300300 C  0107:0211:2
                  IF ELCLASS = ASSIGNOP THEN % ASSIGNMENT TO SUBSCRIPTED           15300400 C  0107:0211:2
                    BEGIN                    % VARIABLE                %114-       15300500 C  0107:0212:0
                      IF PARTIALWORD THEN                              %114-       15300600 C  0107:0213:2
                        IF P1 ≠ FS THEN                                %114-       15300700 C  0107:0213:3
                          ERREXIT(209); % PARTIALWORD NOT LEFT-MOST    %114-       15300800 C  0107:0214:0
                                                                                   15300900 C  0107:0215:3
```

```
XREFIT(TALL,REMEMBERSEQNO,ASSIGNREF);                          %114-      15301000 P  0107:0217:2
IF J = 1 THEN % SINGLE-DIMENSIONED                             %114-      15301100 C  0107:0219:3
  EMITN(TALL.ADDRESS)                                          %114-      15301200 C  0107:0220:1
ELSE                                                           %114-      15301300 C  0107:0222:0
  EMITO(CDC);                                                  %114-      15301400 C  0107:0222:0
IF TALL.CLASS ≥ LONGID THEN % EXPLICIT INDEX OP                %114-      15301500 C  0107:0223:3
  EMITO(INX);                    % REQUIRED                    %114-      15301600 C  0107:0224:1
IF P1 = FR THEN % CALLED FROM FOR STATEMENT                    %114-      15301700 C  0107:0226:0
  GO TO EXIT;                                                  %114-      15301800 C  0107:0226:1
IF TABLE(I+1) = ASTRISK THEN                                   %114-      15301900 C  0107:0227:2
  IF (DUPIT:=(TABLE(I+2) ≥ EQVOP AND TABLE(I+2)                %114-      15302000 P  0107:0229:2
     ≤ MULOP) OR TABLE(I+2) = AMPERSAND) THEN                  %114-      15302100 C  0107:0232:0
    STEPIT;                                                    %114-      15302200 C  0107:0235:3
IF PARTIALWORD OR DUPIT THEN % NEED VALUE ON STACK                         15302300 C  0107:0236:1
  BEGIN                                                        %114-      15302400 C  0107:0238:0
    EMITO(DUP);                                                %114-      15302500 C  0107:0238:1
    EMITO(LOD);                                                %114-      15302600 C  0107:0239:2
  END;                                                         %114-      15302700 C  0107:0240:0
IF PARTIALWORD AND DUPIT THEN                                  %114-      15302800 C  0107:0240:0
  BEGIN                                                        %114-      15302900 C  0107:0241:2
    EMITO(DUP);                                                %114-      15303000 P  0107:0241:3
    EMITI(0,T1,T2);                                            %114-      15303100 C  0107:0242:1
  END;                                                         %114-      15303200 C  0107:0243:3
STEPIT;                                                        %114-      15303300 C  0107:0243:3
IF DUPIT THEN                                                  %114-      15303400 C  0107:0244:0
  SIMPARITH                                                    %114-      15303500 C  0107:0244:1
ELSE                                                           %114-      15303600 C  0107:0245:2
  AEXP;                                                        %114-      15303700 C  0107:0245:3
  EMITD(48-T2,T1,T2);                                          %114-      15303800 C  0107:0246:1
  EMITO(XCH);                                                  %114-      15303900 C  0107:0248:0
  IF TALL.ADDRESS > 1023 THEN                                  %114-      15304000 P  0107:0249:2
    EMITO(PRTF);                                               %114-      15304100 C  0107:0250:0
  EMITO(IF TALL.CLASS MOD 2 = INTARRAYID MOD 2 THEN%114-      15304200 C  0107:0251:3
          IF P1 = FS THEN ISD ELSE ISN                         %114-      15304300 C  0107:0254:0
        ELSE IF P1 = FS THEN STD ELSE SND);                    %114-      15304400 C  0107:0256:1
  STACKCT := 0; % A & B ARE EMPTY                              %114-      15304500 C  0107:0259:3
  P1 := 0;                                                     %114-      15304600 C  0107:0260:1
 END OF ASSIGNMENT TO SUBSCRIPTED VARIABLE                     %114-      15304700 C  0107:0261:3
ELSE                                                           %114-      15304800 C  0107:0261:3
 BEGIN % HANDLING OF NO ASSIGNMENT CASE                        %114-      15304900 C  0107:0261:3
  SPCLMON := P1 = FP OR PARTIALWORD OR ELCLASS ≥                %114-      15305000 P  0107:0262:0
                AMPERSAND;                                     %114-      15305100 C  0107:0264:0
  IF J = 1 THEN % SINGLE DIMENSIONED                           %114-      15305200 C  0107:0265:2
    IF TALL.CLASS ≥ LONGID THEN % NAME ITEM                    %114-      15305300 C  0107:0266:0
      BEGIN                                                    %114-      15305400 C  0107:0267:3
        EMITN(TALL.ADDRESS);                                   %114-      15305500 C  0107:0268:0
        EMITO(INX);                                            %114-      15305600 C  0107:0269:3
        IF SPCLMON THEN                                        %114-      15305700 C  0107:0270:0
          EMITO(LOD);                                          %114-      15305800 C  0107:0270:1
      END                                                      %114-      15305900 C  0107:0271:3
    ELSE % REFERENCE TO SINGLE DIMENSIONED ARRAY               %114-      15306000 P  0107:0271:3
      IF SPCLMON THEN                                          %114-      15306100 C  0107:0271:3
        EMITV(TALL.ADDRESS)                                    %114-      15306200 C  0107:0272:1
      ELSE                                                     %114-      15306300 C  0107:0274:0
        EMITN(TALL.ADDRESS)                                    %114-      15306400 C  0107:0274:0
  ELSE % MULTI-DIMENSIONED CASE                                %114-      15306500 C  0107:0275:3
    EMITO(IF SPCLMON THEN COC ELSE CDC);                       %114-      15306600 C  0107:0276:0
```

```
                    IF P1 = FS THEN % EXPECTED AN ASSIGNMENT          %114-
                      ERREXIT(210);                                   %114-
                    STACKCT := 1; % BECAUSE REGISTERS ARE NON-EMPTY   %114-
                    IF PARTIALWORD THEN                               %114-
                      BEGIN                                           %114-
                        EMIT1(0,T1,T2);                               %114-
                        P1 := 0;                                      %114-
                      END;                                            %114-
                  END OF NO ASSIGNMENT CASE;                          %114-
                END OF SUBSCRIPTED VARIABLES
                ELSE
                BEGIN COMMENT LABELID;
                T1:=TAKE(T2:=GIT(TALL));
                PUT(L,T2);
                IF T1=0 THEN T1:=L;
                IF (T1←L-T1) DIV 4 > 127 THEN BEGIN T1←0;FLAG(50);END;
                EMIT(T1×4+3);
                STEPIT;
                END OF LABELID;
    EXIT  :   END  OF THE VARIABLE ROUTINE;

                                                                                        107 IS  302 LONG, NEXT SEG    3
COMMENT THIS SECTION GENERATES CODE FOR STREAM PROCEDURES;
COMMENT  DO  LABEL DECS UPON APPEARANCE OF LABEL  ;
PROCEDURE DECLARELABEL ;
                BEGIN
                KLASSF ← STLABID;
                VONF ← FORMALF ← FALSE;
                ADDRSF ← 0;
                MAKEUPACCUM;  E; PUTNBUMP(0);
                ELBAT[I] ← ACCUM[0]& LASTINFO[35:35:13];
                END;
          PROCEDURE STREAMSTMT ;
                BEGIN
                DEFINE LFTPAREN=LEFTPAREN#,LOC=[36:12]#,LASTGT=[24:12]#,

                                                                                    START OF SEGMENT *********  108
                    LOCFLD=36:36:12#,LGTFLD=24:24:12#;
                DEFINE LEVEL=LVL#,ADDOP=ADOP#;
          DEFINE
                JFW   =  39#,    COMMENT  7.5.5.1  JUMP FORWARD UNCONDITIONAL  ;
                 RCA  =  40#,  COMMENT    7.5.7.6  RECALL CONTROL ADDRESS      ;
                JRV   =  47#,   COMMENT   7.5.5.2  JUMP REVERSE UNCONDITIONAL  ;
                CRF   =  35#,   COMMENT    7.5.10.6 CALL REPEAT FIELD          ;
                 BNS  =  42#,   COMMENT   7.5.5.5  BEGIN LOOP                  ;
                NOP   =   1#,   COMMENT                                       ;
                ENS   =  41#,   COMMENT   7.5.5.6  END LOOP                   ;
                TAN   =  30#,   COMMENT   7.5.3.7  TEST FOR ALPHAMERIC        ;
                BIT   =  31#,   COMMENT   7.5.3.8  TEST BIT                   ;
                 JFC  =  37#,   COMMENT   7.5.5.3  JUMP FORWARD CONDITIONAL   ;
                SED   =  06#,   COMMENT   7.5.7.8  SET DESTINATION ADDRESS    ;
                RSA   =  43#,   COMMENT   7.5.7.4  RECALL SOURCE ADDRESS      ;
                TRP   =  60#,   COMMENT   7.5.2.2  TRANSFER PROGRAM CHARACTERS ;
                BSS   =   3#,   COMMENT   7.5.6.6  SKIP SOURCE BIT            ;
                BSD   =   2#,   COMMENT   7.5.6.5  SKIP DESTINATION BITS      ;
                SFC   =  34#,      COMMENT   7.5.10.1  SET COUNT              ;
                JNS   =  38#;    COMMENT  7.5.5.7  JUMP OUT LOOP              ;
          PROCEDURE ADJUST;;
             COMMENT FIXC EMITS  BASICLY FORWARD JUMPS, HOWEVER IN THE CASE
```

```
15306700 C  0107:0278:1
15306800 C  0107:0279:3
15306900 C  0107:0281:2
15307000 P  0107:0282:0
15307100 C  0107:0282:1
15307200 C  0107:0283:2
15307300 C  0107:0284:1
15307400 C  0107:0285:3
15307500 C  0107:0285:3
15376000 T  0107:0285:3
15376100 T  0107:0285:3
15376200 T  0107:0285:3
15376300 T  0107:0286:0
15376400 T  0107:0288:0
15376500 T  0107:0289:2
15376600 T  0107:0291:2
15376700 T  0107:0295:3
15376800 T  0107:0297:2
15376900 T  0107:0297:3
15377000 T  0107:0297:3
16000000 T  0003:0628:1
16000050 T  0003:0628:1
16000100 T  0003:0628:1
16000200 T  0003:0628:1
16000300 T  0003:0628:1
16000400 T  0003:0629:3
16000500 T  0003:0631:2
16000600 T  0003:0631:3
16000700 T  0003:0633:3
16000800 T  0003:0636:0
16001000 T  0003:0636:0
16002000 T  0003:0636:0
16003000 T  0003:0636:0
16004000 T  0108:0000:0
16005000 T  0108:0000:0
16006000 T  0108:0000:0
16007000 T  0108:0000:0
16008000 T  0108:0000:0
16009000 T  0108:0000:0
16010000 T  0108:0000:0
16011000 T  0108:0000:0
16012000 T  0108:0000:0
16013000 T  0108:0000:0
16014000 T  0108:0000:0
16015000 T  0108:0000:0
16016000 T  0108:0000:0
16017000 T  0108:0000:0
16018000 T  0108:0000:0
16019000 T  0108:0000:0
16020000 T  0108:0000:0
16021000 T  0108:0000:0
16022000 T  0108:0000:0
16023000 T  0108:0000:0
16023100 T  0108:0000:0
16024000 T  0108:0000:0
```

```
                    OF INSTRUCTIONS INTERPTED AS JUMPS BECAUSE OF A CRF ON
                    A VALUE = 0 AND THE JUMP ≥ 64 SYLLABLES A JFW 1 AND
                    A RCA L (L IS STACK ADDRESS OF A PSEUDO LABEL WHICH
                    MUST ALSO BE MANUFACTURED) IS EMITTED.  ;
PROCEDURE FIXC(S); VALUE S; REAL S;
        BEGIN
        REAL SAVL,D,F;

            IF D← (SAVL←L) - (L←S)-1 ≤ 63 THEN EMITC(D,GET(S))
                ELSE  FLAG(700);
            L←SAVL ;
        END  FIXC  ;

    COMMENT EMITJUMP IS CALLED BY GOTOS AND JUMPCHAIN.
            THIS ROUTINE WILL EMIT A JUMP IF THE DISTANCE IS ≤ 63
            SYLLABLES ,OTHERWISE, IT GETS A PRT CELL AND STUFFS THE
            STACK ADDRESS INTO THE LABEL ENTRY IN INFO AND EMITS AN
            RCA ON THIS STACK CELL. AT EXECUTION TIME ACTUAL PARAPART
            INSURES US THAT THIS CELL WILL CONATIN A LABEL DESCRIPTOR
            POINTING TO OUR LABEL IN QUESTION.  ;
PROCEDURE EMITJUMP( E); VALUE E; REAL E;
        BEGIN
        REAL T,D;

        REAL ADDR;
        IF ABS(
        D←(T←TAKE(GIT(E)),LOC)-L-1)≥64 THEN
            FLAG(700)
        ELSE EMITC(D,IF D <0 THEN JRV ELSE JFW);
        END  EMIT JUMP;

    COMMENT   WHEN JUMPCHAIN IS CALLED THERE IS A LINKEDLIST IN THE CODE
            ARRAY WHERE JFWS MUST BE PLACED. THE 1ST LINK IS POINTED
            TO BY THE LOC FIELD OF EACH LABEL ENTRY IN INFO, THE LAST
            LINK IS = 4096.  ;
PROCEDURE JUMPCHAIN( E); VALUE E;REAL E;
        BEGIN
        REAL SAVL ,LINK;

        SAVL ← L;
        L ←  TAKE(GIT(E)),LASTGT ;
        WHILE L≠ 4095 DO
            BEGIN
            LINK ← GET(L);
            EMITJUMP( E);
            L ← LINK
            END;
        L←SAVL;
        END  JUMPCHAIN  ;

    COMMENT NESTS COMPILES THE NEST STATEMENT.
            A VARIABLE NEST INDEX CAUSES THE CODE,
                CRF V, BNS 0 ,NOP,NOP, TO BE GENERATED INITIALLY.
            AT THE RIGHT PAREN THE BNS IS FIXED WITH THE LENGTH OF
            THE NEST (NUMBER OF SYLLABLES) IF THE LENGTH ≤63,OTHERWISE
            IT IS FIXED WITH A 1 AND THE NOPS REPLACED WITH JFW 1,
            RCA P.  THIS IS DONE BECAUSE THE VALUE OF V AT EXECUTION
```

```
                16025000 T  0108:0000:0
                16026000 T  0108:0000:0
                16027000 T  0108:0000:0
                16028000 T  0108:0000:0
                16029000 T  0108:0000:0
                16030000 T  0108:0000:0
                16031000 T  0108:0000:0
          START OF SEGMENT **********  109
                16032000 T  0109:0000:0
                16033000 T  0109:0004:1
                16034000 T  0109:0006:1
                16057000 T  0109:0007:2
           109 IS   10 LONG, NEXT SEG  108
                16058000 T  0108:0000:0
                16059000 T  0108:0000:0
                16060000 T  0108:0000:0
                16061000 T  0108:0000:0
                16062000 T  0108:0000:0
                16063000 T  0108:0000:0
                16064000 T  0108:0000:0
                16065000 T  0108:0000:0
                16066000 T  0108:0000:0
                16067000 T  0108:0000:0
          START OF SEGMENT **********  110
                16068000 T  0110:0000:0
                16069000 T  0110:0000:0
                16070000 T  0110:0000:0
                16071000 T  0110:0004:1
                16079000 T  0110:0005:2
                16080000 T  0110:0009:2
           110 IS   12 LONG, NEXT SEG  108
                16081000 T  0108:0000:0
                16082000 T  0108:0000:0
                16083000 T  0108:0000:0
                16084000 T  0108:0000:0
                16085000 T  0108:0000:0
                16086000 T  0108:0000:0
                16087000 T  0108:0000:0
          START OF SEGMENT **********  111
                16088000 T  0111:0000:0
                16089000 T  0111:0000:1
                16090000 T  0111:0003:2
                16091000 T  0111:0004:0
                16092000 T  0111:0004:0
                16093000 T  0111:0005:3
                16094000 T  0111:0006:0
                16095000 T  0111:0006:0
                16096000 T  0111:0009:2
                16097000 T  0111:0009:3
           111 IS   12 LONG, NEXT SEG  108
                16098000 T  0108:0000:0
                16099000 T  0108:0000:0
                16100000 T  0108:0000:0
                16101000 T  0108:0000:0
                16102000 T  0108:0000:0
                16103000 T  0108:0000:0
                16104000 T  0108:0000:0
```

```
                    MAY = 0 AND THIS CODE CAUSES A JUMP AROUND THE NEST.                    16105000 T  0108:0000:0
                    JUMPOUT INFO IS REMEMBERED IN A RECURSIVE CELL AND                      16106000 T  0108:0000:0
                    NEST LEVEL INCREASED BY ONE.                                            16107000 T  0108:0000:0
                    WHEN THE RIGHT PAREN IS REACHED,(IF THE STATEMENTS IN                   16108000 T  0108:0000:0
                    THE NEST COMPILED), JOINFO IS CHECKED FOR THE EXISTANCE                 16109000 T  0108:0000:0
                    OF JUMPOUT STATEMENTS IN THE NEST,IF SO,THE THE JUMPS                   16110000 T  0108:0000:0
                    ARE FIXED BY FAKING TOTOS INTO COMPILING THE REQUIRED                   16111000 T  0108:0000:0
                    JUMPS.                                                                  16112000 T  0108:0000:0
                    FINALLY THE BNS IS FIXED,IF REQUIRED,AND NEST LEVEL                     16113000 T  0108:0000:0
                    AND JOINFO RESTORED TO THEIR ORIGINAL VALUES.  ;                        16114000 T  0108:0000:0
PROCEDURE NESTS;                                                                            16115000 T  0108:0000:0
        BEGIN                                                                               16116000 T  0108:0000:0
        LABEL EXIT;                                                                         16117000 T  0108:0000:0
                                                                                  START OF SEGMENT **********  112
        REAL JOINT,BNSFIX;                                                                  16118000 T  0112:0000:0
        IF ELCLASS≠LITNO THEN                                                               16119000 T  0112:0000:0
             BEGIN                                                                          16120000 T  0112:0000:1
             EMITC(ELBAT[I].ADDRESS,CRF); BNSFIX← L;                                        16121000 T  0112:0001:2
             EMIT(BNS);                                                                     16122000 T  0112:0003:3
             END                                                                            16123000 T  0112:0004:1
        ELSE EMITC(ELBAT[I].ADDRESS,BNS);                                                   16124000 T  0112:0004:1
        IF STEPI ≠ LFTPAREN THEN BEGIN ERR(262); GO TO EXIT END;                            16125000 T  0112:0006:1
        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
        BEGIN                                                                               16157000 T  0108:0000:0
        REAL GT1;                                                                           16157100 T  0108:0000:0
                                                                                  START OF SEGMENT **********  113
```

```
                ADJUST;
                GT1 ← ELBAT[I];
                XMARK(LBLREF); % MARK LABEL OCCURENCE FOR XREF          %110-
                IF STEPI ≠ COLON THEN ERR(258)
                ELSE
                     BEGIN
                     IF TAKE(GT2←GIT(GT1)).LOC ≠ 0 THEN FLAG(259);
                     IF GT1>0 THEN
                          BEGIN
                          PUT(-(TAKE(GT1)&NESTLEVEL[11:43:5]),GT1);
                          PUT(-L,GT2)
                          END
                     ELSE
                          BEGIN
                          IF GT1.LEVEL≠NESTLEVEL THEN FLAG(257);
                          PUT((-L)&TAKE(GT2)[LGTFLD],GT2);
                          JUMPCHAIN(GT1);
                          END;
                     END
                ;  STEPIT;
                END  LABELS  ;
                                                                  113 IS   27 LONG, NEXT SEG  108
           COMMENT   IFS COMPILES IF STATEMENTS,
                     FIRST THE TEST IS COMPILED. NOTE THAT IN THE
                     CONSTRUCTS  "SC RELOP DC" AND  "SC RELOP STRING" THAT
                     THE SYLLABLE EMITTED IS FETCHED FROM ONE OF TWO FIELDS
                     IN THE ELBAT WORD FOR THE RELATIONAL OPERATOR, OTHERWISE
                     THE CODE IS EMITTED STRAIGHTAWAY.
                     A TEST IS MADE TO SEE WHETHER THE STATEMENT AFTER THE
                     "THEN" COULD POSSIBLY BE LONGER THAN 63 SYLLABLES,AND IF
                     SO, Z NOPS ARE EMITTED FOR FIXC IN CASE A RCA WILL HAVE
                     TO BE GENERATED.
                     THIS PROCEDURE DOES NO OPTIMAZATION IN THE CASES
                     IF THEN GO TO L,IF THEN STATEMENT ELSE GO TO L, OR
                     IF THEN GO TO L1 ELSE GO TO L2   ;
           PROCEDURE  IFS;   BEGIN
                DEFINE COMPARECODE=[42:6]#,TESTCODE=[36:6]#;

                LABEL  IFSB,IFTOG,IFSC,EXIT;
                SWITCH IFSW ← IFSB,IFTOG,IFSC;
                REAL ADDR,FIX1,FIX2  ;
                ADDR←1  ;
                GO TO IFSW[STEPI -SBV+1] ;
                IF ELCLASS=LOCLID THEN
                     BEGIN
                     EMITC(ELBAT[I],ADDRESS,CRF);
                     ADDR←0;
                     END
                ELSE
                     IF ELCLASS=LITNO THEN ADDR ← ELBAT[I],ADDRESS
                ELSE BEGIN ERR(250); GO TO EXIT END;
                IF STEPI ≠ SCV THEN BEGIN ERR(263);GO TO EXIT END;
           IFSC: IF STEPI ≠ RELOP THEN BEGIN ERR(264);GO TO EXIT END;
                IF STEPI = DCV THEN EMITC( ADDR,ELBAT[I-1].COMPARECODE)
                ELSE
                     IF ELCLASS = STRNGCON THEN
                          EMITC(ACCUM[1],[18:6],ELBAT[I-1],TESTCODE)
```

```
                                                                  16158000 T  0113:0000:0
                                                                  16159000 T  0113:0000:1
                                                                  16159100 C  0113:0001:3
                                                                  16160000 T  0113:0005:3
                                                                  16161000 T  0113:0007:3
                                                                  16162000 T  0113:0008:0
                                                                  16163000 T  0113:0008:1
                                                                  16164000 T  0113:0012:1
                                                                  16165000 T  0113:0013:2
                                                                  16166000 T  0113:0013:3
                                                                  16167000 T  0113:0016:1
                                                                  16168000 T  0113:0017:2
                                                                  16169000 T  0113:0017:3
                                                                  16170000 T  0113:0017:3
                                                                  16171000 T  0113:0018:0
                                                                  16172000 T  0113:0020:1
                                                                  16173000 T  0113:0023:3
                                                                  16174000 T  0113:0024:0
                                                                  16175000 T  0113:0024:0
                                                                  16176000 T  0113:0024:0
                                                                  16177000 T  0113:0024:1
                                                                  16178000 T  0108:0000:0
                                                                  16179000 T  0108:0000:0
                                                                  16180000 T  0108:0000:0
                                                                  16181000 T  0108:0000:0
                                                                  16182000 T  0108:0000:0
                                                                  16183000 T  0108:0000:0
                                                                  16184000 T  0108:0000:0
                                                                  16185000 T  0108:0000:0
                                                                  16186000 T  0108:0000:0
                                                                  16187000 T  0108:0000:0
                                                                  16188000 T  0108:0000:0
                                                                  16189000 T  0108:0000:0
                                                                  16190000 T  0108:0000:0
                                                                  16191000 T  0108:0000:0
                                                                  16192000 T  0108:0000:0
                                                          START OF SEGMENT **********  114
                                                                  16193000 T  0114:0000:0
                                                                  16194000 T  0114:0000:0
                                                                  16195000 T  0114:0004:1
                                                                  16196000 T  0114:0004:1
                                                                  16197000 T  0114:0005:3
                                                                  16198000 T  0114:0009:2
                                                                  16199000 T  0114:0009:3
                                                                  16200000 T  0114:0010:0
                                                                  16201000 T  0114:0012:0
                                                                  16202000 T  0114:0012:1
                                                                  16203000 T  0114:0012:1
                                                                  16204000 T  0114:0012:1
                                                                  16205000 T  0114:0014:1
                                                                  16206000 T  0114:0017:3
                                                                  16207000 T  0114:0020:1
                                                                  16208000 T  0114:0023:3
                                                                  16209000 T  0114:0026:1
                                                                  16210000 T  0114:0027:3
                                                                  16211000 T  0114:0028:1
```

```
        ELSE
        IF ELCLASS=LITNO THEN EMITC(C,ELBAT[I-1],TESTCODE) ELSE
        IF ELCLASS≤IDMAX AND Q="5ALPHA" THEN EMITC(17,TAN)
        ELSE BEGIN ERR(265);GO TO EXIT END;
        GO TO IFTOG  ;
 IFSB:  EMITC(1,BIT);
IFTOG:  IF STEPI ≠ THENV THEN BEGIN ERR(266); GO TO EXIT  END;
        FIX1 ← L;
        EMIT(JFC);
        IF STEPI≠ELSEV THEN%
        STREAMSTMT;
        IF ELCLASS= ELSEV THEN
             BEGIN
             FIX2 ← L;  EMIT(JFW);
             FIXC(FIX1);
             STEPIT;
             STREAMSTMT;
             FIXC(FIX2);
             END
        ELSE FIXC(FIX1);
 EXIT:END   IFS  ;

  COMMENT GOTOS HANDLES GO TO AND THE LAST PART OF JUMP OUT TO
             STATEMENTS.
             IF THE LABEL HAS BEEN ENCOUNTERED THEN EMITJUMP IS CALLED
             AN PRODUCES A JRV OR RCA IN THE CASE OF JUMPS≥64 SYLLABL
             ES, OTHERWISE, A LINK IS EMITTED POINTING ANY PREVIOUS
             GO TOS IN THE CASE OF FORWARD JUMPS.
             FINALLY, IF THE NEST LEVEL IS DEFINED THEN IT IS CHECKED
             AGAINST THE CURRENT LEVEL MINUS THE NUMBER OF LEVELS TO
             BE JUMPED OUT. OTHERWISE,NEST LEVEL IS DEFINED.  ;
PROCEDURE GOTOS;
        BEGIN
        LABEL EXIT;

        IF STEPI ≠TOV THEN I←I-1 ;
             IF STEPI ≠ STLABID THEN IF ELCLASS ≤ IDMAX THEN
                  DECLARELABEL ELSE BEGIN ERR(260); GO TO EXIT END;
        IF(GT2←TAKE(GIT(GT1←ELBAT[I]))).MON=1
        OR GT2.LOC≠0 THEN EMITJUMP(GT1)
        ELSE
          BEGIN   PUT(0&L[24:36:12],GIT(GT1));
             IF GT1>0 THEN
                  BEGIN
                  PUT(-(TAKE(GT1)&(NESTLEVEL-JUMPLEVEL)[11:43:5]),GT1);
                  EMITN(1023);
                  END
             ELSE
                  BEGIN
                  IF GT1.LEVEL ≠ NESTLEVEL-JUMPLEVEL THEN FLAG(257);
                  EMIT(GT2,LASTGT);
                  END;
          END;
        JUMPLEVEL←0 ;
 EXIT: END  GOTOS   ;

  COMMENT  RELEASES COMPILES THE STREAM RELEASE STATEMENT.
```

```
16212000 T  0114:0031:2
16212500 T  0114:0032:0
16213000 T  0114:0036:0
16214000 T  0114:0039:3
16215000 T  0114:0043:2
16216000 T  0114:0043:3
16217000 T  0114:0045:2
16218000 T  0114:0047:3
16219000 T  0114:0048:1
16220000 T  0114:0049:2
16229000 T  0114:0050:0
16230000 T  0114:0051:2
16231000 T  0114:0052:0
16232000 T  0114:0052:1
16233000 T  0114:0054:0
16234000 T  0114:0054:1
16235000 T  0114:0055:2
16236000 T  0114:0055:3
16237000 T  0114:0056:1
16238000 T  0114:0056:1
16239000 T  0114:0057:3
114 IS   61 LONG, NEXT SEG  108
16240000 T  0108:0000:0
16241000 T  0108:0000:0
16242000 T  0108:0000:0
16243000 T  0108:0000:0
16244000 T  0108:0000:0
16245000 T  0108:0000:0
16246000 T  0108:0000:0
16247000 T  0108:0000:0
16248000 T  0108:0000:0
16249000 T  0108:0000:0
16250000 T  0108:0000:0
16251000 T  0108:0000:0
START OF SEGMENT **********  115
16252000 T  0115:0000:0
16253000 T  0115:0002:1
16253100 T  0115:0005:2
16254000 T  0115:0007:3
16255000 T  0115:0010:1
16256000 T  0115:0013:3
16257000 T  0115:0014:0
16258000 T  0115:0017:2
16259000 T  0115:0017:3
16260000 T  0115:0018:0
16261000 T  0115:0021:2
16262000 T  0115:0022:0
16263000 T  0115:0022:0
16264000 T  0115:0022:0
16265000 T  0115:0022:1
16266000 T  0115:0025:3
16267000 T  0115:0027:2
16268000 T  0115:0027:2
16269000 T  0115:0027:2
16270000 T  0115:0027:3
115 IS   29 LONG, NEXT SEG  108
16271000 T  0108:0000:0
```

```
                THE CODE GENERATED IS :                                                 16272000 T  0108:0000:0
                     SED  FILE                                                          16273000 T  0108:0000:0
                     RSA  0.                                                            16274000 T  0108:0000:0
                AT EXECUTION TIME THIS CAUSES AN INVALID ADDRESS WHICH IS               16275000 T  0108:0000:0
                INTERPETED BY THE MCP TO MEAN RELEASE THE FILE POINTED TO               16276000 T  0108:0000:0
                BY THE DESTINATION ADDRESS.                                             16277000 T  0108:0000:0
                THE MONITOR BIT IS SET IN INFO FOR THE LOCAL VARIABLE SO                16278000 T  0108:0000:0
                THAT ACUTAL PARAPART MAY BE INFORMED LATER THAT A FILE                  16279000 T  0108:0000:0
                MUST BE PASSED FOR THIS FORMAL PARAMETER;                               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
      COMMENT   INDEXS  COMPILE STATEMENTS BEGINING WITH SI,DI,CI,TALLY                 16290000 T  0108:0000:0
                OR LOCALIDS .                                                           16291000 T  0108:0000:0
                THREE CASES PRESENT THEMSELVES,                                         16292000 T  0108:0000:0
                     LETING X BE EITHER OF  SI,DI,CI OR TALLY, THEY ARE:                16293000 T  0108:0000:0
                     CASE I    LOCLID ← X                                               16294000 T  0108:0000:0
                     CASE II   X ← X ...                                                16295000 T  0108:0000:0
                     CASE III  X ← EITHER LOC,LOCLID,SC OR DC,                          16296000 T  0108:0000:0
                THE VARIABLE "INDEX" IS COMPUTED,DEPENDING UPON WHICH                   16297000 T  0108:0000:0
                CASE EXISTS,SUCH THAT ARRAY ELEMENT "MACRO[INDEX]"CONTAINS              16298000 T  0108:0000:0
                THE CODE TO BE EMITTED.                                                 16299000 T  0108:0000:0
                     EACH ELEMENT OF MACRO HAS  1-3 SYLLABLES ORDERED FROM              16300000 T  0108:0000:0
                     RIGHT TO LEFT, UNUSED SYLLABLES MUST = 0.    EACH MACRO            16301000 T  0108:0000:0
                     MAY REQUIRE AT MOST ONE REPEAT PART,                               16302000 T  0108:0000:0
                     IN THIS PROCEDURE,INDEXS,THE VARIABLE "ADDR" CONTAINS THE          16303000 T  0108:0000:0
                     PROPER REPEAT PART BY THE TIME THE LABEL "GENERATE" IS             16304000 T  0108:0000:0
                     ENCOUNTERED. THE SYLLABLES ARE FETCHED FROM MACRO[TYPE]            16305000 T  0108:0000:0
                     ONE AT A TIME AND IF THE REPEAT PART ≠ 0 THEN"ADDR" IS             16306000 T  0108:0000:0
                     USED AS THE REPEAT PART,THUS BUILDING A SYLLABLE WITH              16307000 T  0108:0000:0
                     THE PROPER  ADDRESS AND OPERATOR .                                 16308000 T  0108:0000:0
                     NOTE: IF MACRO[TYPE] = 0 THEN THIS SIGNIFIES A SYNTAX              16309000 T  0108:0000:0
                          ERROR.                                ;                       16310000 T  0108:0000:0
           PROCEDURE INDEXS;                                                            16311000 T  0108:0000:0
                     BEGIN                                                              16312000 T  0108:0000:0
                     LABEL EXIT,GENERATE,L,L1;                                          16313000 T  0108:0000:0
                                                                          START OF SEGMENT **********  116
                     INTEGER TCLASS,INDEX,ADDR,J;                                       16314000 T  0116:0000:0
                     TCLASS ← ELCLASS ;                                                 16315000 T  0116:0000:0
                     IF STEPI ≠ ASSIGNOP THEN BEGIN ERR(251); GO TO EXIT END;           16316000 T  0116:0000:1
                     IF TCLASS = LOCLID THEN                                            16317000 T  0116:0003:3
                          BEGIN                                                         16318000 T  0116:0004:0
                          XMARK(ASSIGNREF);                                  %110-      16318500 C  0116:0004:1
                          IF SIV>STEPI OR ELCLASS>TALLYV THEN GO TO L;                  16319000 T  0116:0009:2
                          INDEX ← 32 + ELCLASS-SIV;                                     16320000 T  0116:0011:3
                          ADDR ← ELBAT[I-2].ADDRESS;                                    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
          BEGIN                                                                 16363000 T  0108:0000:0
          INTEGER ADDR,J,K,L,T;                                                 16364000 T  0108:0000:0
                                                                    START OF SEGMENT **********  117
          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
                    EMITC(ADDR,TRP);
                    IF STEPI≠STRNGCON THEN
                         BEGIN ERR(255);GO TO EXIT END;
                    IF ADDR MOD 2 ≠ 0 THEN
                         BEGIN
                         EMIT(ACCUM[1],[18:6]);  J ← 1;
                         END ;
                    FOR K ←J+2 STEP 2 UNTIL ADDR DO
                         BEGIN
                         FOR L ←6,7 DO
                         MOVECHARACTERS(1,ACCUM[1],2+(IF J←J+1>COUNT THEN J←1
                                   ELSE J),T,L  );
                         EMIT(T);
                    END  END
               ELSE
          L1:  ERR(250);
     EXIT:END  DSS  ;

     COMMENT SKIPS  COMPILES THE SKIP BIT STATEMENT,
                    IF THE REPEAT INDEX IS A LOCALID THEN A CRF IS EMITTED,
                    A  BSS OR BSD IS THEN EMITTED FOR SKIP SOURCE BITS (SB)
                    OR   SKIP DESTINATION BITS (DB) RESPECTIVELY ;
PROCEDURE SKIPS ;
          BEGIN
          REAL ADDR;

          IF STEPI = LOCLID THEN
               BEGIN
               EMITC(ELBAT[I].ADDRESS,CRF); ADDR←0; STEPIT;
               END
          ELSE IF ELCLASS = LITNO THEN
               BEGIN
               ADDR← ELBAT[I].ADDRESS; STEPIT
               END
          ELSE ADDR ← 1 ;
          IF ELCLASS =SBV THEN EMITC(ADDR,BSS)
          ELSE
          IF ELCLASS =DBV THEN  EMITC(ADDR,BSD)
          ELSE ERR(250);
          END  SKIPS  ;

     COMMENT JUMPS COMPILES JUMP OUT AND JUMP OUT TO STATEMENTS,
                    JUMP OUT TO STATEMENTS CAUSE JUMP LEVEL TO BE SET TO
                    THE NUMBER OF LEVELS SPECIFIED. THEN THIS NUMBER OF
                    JNS ARE EMITTED AND GOTOS IS CALLED TO COMPILE THE
                    JUMP INSTRUCTION.
                    SIMPLE JUMP OUTS ARE HANDLED BY EMITTING ONE JNS,ENTERING
                    A PSEUDO SYLABID IN INFO AND SETTING ELBAT[I] SUCH THAT
                    THE GOTOS PROCEDURE WILL PERFORM THE ACTION OF SETTING
                    UP THE LINKS FOR LATER FIX UPS. THE NEST STATEMENT CAUSES
                    THESE FIX UPS(IF EMITTING OF JUMP INSTRUCTIONS) BY CALLING
                    GO TOS WHEN THE RIGHT PAREN IS ENCOUNTERED.     ;
PROCEDURE JUMPS;
          BEGIN
          JUMPLEVEL←1;
               IF STEPI≠DECLARATORS THEN IF ACCUM[1]≠"30UT00" THEN
```

```
16382000 T  0117:0021:2
16383000 T  0117:0021:3
16384000 T  0117:0022:1
16384500 T  0117:0023:3
16385000 T  0117:0025:3
16386000 T  0117:0026:1
16387000 T  0117:0027:2
16388000 T  0117:0029:3
16389000 T  0117:0029:3
16390000 T  0117:0033:3
16391000 T  0117:0033:3
16392000 T  0117:0038:0
16393000 T  0117:0041:3
16394000 T  0117:0044:1
16395000 T  0117:0045:3
16396000 T  0117:0046:0
16397000 T  0117:0046:0
16398000 T  0117:0047:3
117 IS   52 LONG, NEXT SEG  108
16399000 T  0108:0000:0
16400000 T  0108:0000:0
16401000 T  0108:0000:0
16402000 T  0108:0000:0
16403000 T  0108:0000:0
16404000 T  0108:0000:0
16405000 T  0108:0000:0
START OF SEGMENT **********  118
16406000 T  0118:0000:0
16407000 T  0118:0001:2
16408000 T  0118:0001:3
16409000 T  0118:0004:1
16410000 T  0118:0004:1
16411000 T  0118:0005:3
16412000 T  0118:0006:0
16413000 T  0118:0007:3
16414000 T  0118:0008:0
16415000 T  0118:0009:3
16416000 T  0118:0011:2
16417000 T  0118:0011:3
16418000 T  0118:0014:0
16419000 T  0118:0015:3
118 IS   18 LONG, NEXT SEG  108
16420000 T  0108:0000:0
16421000 T  0108:0000:0
16422000 T  0108:0000:0
16423000 T  0108:0000:0
16424000 T  0108:0000:0
16425000 T  0108:0000:0
16426000 T  0108:0000:0
16427000 T  0108:0000:0
16428000 T  0108:0000:0
16429000 T  0108:0000:0
16430000 T  0108:0000:0
16431000 T  0108:0000:0
16432000 T  0108:0000:0
16433000 T  0108:0000:0
16434000 T  0108:0001:3
```

```
                                        FLAG(261);
     IF STEPI = LITNO THEN JUMPLEVEL← ELBAT[I].ADDRESS
     ELSE BEGIN
          IF ELCLASS≠ TOV AND ELCLASS≠ STLABID THEN
          BEGIN
COMMENT   SIMPLE JUMP OUT STATEMENT;
          IF JOINFO = 0 THEN
               BEGIN
               JOINFO ← NEXTINFO ;
               PUTNBUMP(STACKHEAD[0].LINK&(STLABID×2+1)
                    [2:40:8]&2[27:40:8 ]);
               PUTNBUMP(0&(JOINFO-LASTINFO    )[ 4:40:8]);
               PUTNBUMP (0);
               LASTINFO ← JOINFO;
               END;
          ELBAT[I←  I-1]←  TAKE(JOINFO)&JOINFO[35:35:13];
          END; I←I-1 ;
     END;
     FOR GT1← 1 STEP 1 UNTIL JUMPLEVEL DO
          EMIT(  JNS);
     GOTOS;
     END JUMPS;
COMMENT STREAMSTMT  ENVOKES THE APPROPRIATE PROCEDURE TO HANDLE
          THE VARIOUS AND SUNDRY STREAM PROCEDURE STATEMENTS.
          THE STATEMENTS ARE BROKEN DOWN AS FOLLOWS:
               IDENTIFIED BY          PROCEDURE ENVOKED
                 END                    GO TO FINI
                 SEMICOLON              GO TO FINI
                 )                      GO TO FINI
                 IF                     IFS
                 GO                     GOTOS
                 RELEASE                RELEASES
               BEGIN                    COMPOUNDTAIL
               SI,DI,CI,TALLY,LOCALID INDEXS
               DS                       DSS
               SKIP                     SKIPS
                 JUMP                   JUMPS
                 LABELID                LABELS
                 LITERAL NO.,LOCALID(   NESTS
          UPON EXITING,STREAMSTMT ASSURES THAT "I" POINTS TO
          THE SEMICOLON ,END OR ) IN SYNTACICALLY CORRECT PROGRAMS;
     LABEL  L,L1,L2,L3,L4,L5,L6,L7,L8,L9,L10,EXIT,FINI,START;
     SWITCH TYPE ← FINI,L,FINI,L3,L4,L5,L6,L7,L7,L7,L8,L9,L10;
START:    GO TO TYPE[ ELCLASS-ENDV+1];
          IF ELCLASS= RTPAREN THEN GO TO FINI ;
          IF ELCLASS= STLABID THEN GO TO L2 ;

     IF ELCLASS <IDMAX AND ELCLASS≠LOCLID THEN BEGIN
          DECLARELABEL; GO TO L2;  END;
     IF ELCLASS = LITNO OR ELCLASS = LOCLID AND TABLE(I+1)
          = LFTPAREN THEN GO TO L1;
     IF ELCLASS = LOCLID  THEN  GO TO L7;
L:        ERR( 250 ); GO TO FINI ;
L1:       NESTS;     GO TO EXIT;
L2:       LABELS;    GO TO START;
L3:       IFS;       GO TO FINI;
L4:       GOTOS;     GO TO EXIT;
```

```
16434100 T  0108:0004:0
16435000 T  0108:0005:3
16436000 T  0108:0007:2
16437000 T  0108:0010:0
16438000 T  0108:0011:3
16439000 T  0108:0012:0
16440000 T  0108:0012:0
16441000 T  0108:0013:2
16442000 T  0108:0013:3
16443000 T  0108:0014:0
16444000 T  0108:0016:1
16445000 T  0108:0018:1
16446000 T  0108:0021:2
16447000 T  0108:0021:3
16448000 T  0108:0022:1
16449000 T  0108:0022:1
16450000 T  0108:0026:0
16451000 T  0108:0027:3
16452000 T  0108:0027:3
16453000 T  0108:0029:2
16454000 T  0108:0032:0
16455000 T  0108:0032:1
16456000 T  0108:0032:1
16457000 T  0108:0032:1
16458000 T  0108:0032:1
16459000 T  0108:0032:1
16460000 T  0108:0032:1
16461000 T  0108:0032:1
16462000 T  0108:0032:1
16463000 T  0108:0032:1
16464000 T  0108:0032:1
16465000 T  0108:0032:1
16466000 T  0108:0032:1
16467000 T  0108:0032:1
16468000 T  0108:0032:1
16469000 T  0108:0032:1
16470000 T  0108:0032:1
16471000 T  0108:0032:1
16472000 T  0108:0032:1
16473000 T  0108:0032:1
16474000 T  0108:0032:1
16475000 T  0108:0032:1
16476000 T  0108:0032:1
16477000 T  0108:0043:2
16478000 T  0108:0047:2
16481000 T  0108:0048:0
16482000 T  0108:0049:3
16482100 T  0108:0049:3
16482200 T  0108:0051:3
16482300 T  0108:0052:1
16482400 T  0108:0055:2
16482500 T  0108:0057:2
16483000 T  0108:0058:1
16484000 T  0108:0060:0
16485000 T  0108:0062:0
16486000 T  0108:0063:2
16487000 T  0108:0064:0
```

```
L5:
        L6:     I←I+1 ; COMPOUNDTAIL; GO TO FINI;                                  16488000 T  0108:0065:2
        L7:     INDEXS;    GO TO EXIT;                                             16489000 T  0108:0065:2
        L8:     DSS;       GO TO EXIT;                                             16490000 T  0108:0067:2
        L9:     SKIPS;     GO TO EXIT;                                             16491000 T  0108:0069:2
        L10:    JUMPS;     GO TO EXIT;                                             16492000 T  0108:0070:0
      EXIT:     STEPIT;                                                            16493000 T  0108:0071:2
      FINI:     END   STREAMSTMT;                                                  16494000 T  0108:0072:0
                                                                                   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
                                                                                 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[INDFX],CF)#;                    %110-   17002010 C  0119:0000:0
      ARRAY TIMINGS[0:2,0:3];                                              %110-   17002012 C  0119:0000:0
      PROCEDURE SAVETIMES(I);                                              %110-   17002015 C  0119:0002:1
        VALUE I; INTEGER I;                                                %110-   17002020 C  0119:0002:1
      BEGIN                                                                %110-   17002025 C  0119:0002:1
        INTEGER J;                                                         %110-   17002030 C  0119:0002:1
                                                                                 START OF SEGMENT **********  120
        FOR J := 1 STEP 1 UNTIL 3 DO                                       %110-   17002035 C  0120:0000:0
          TIMINGS[I,J] := TIME(J);                                         %110-   17002040 C  0120:0001:2
      END;                                                                 %110-   17002045 C  0120:0005:3
                                                                          120 IS    8 LONG, NEXT SEG  119
      PROCEDURE UPDATETIMES(I);                                            %110-   17002050 C  0119:0002:1
        VALUE I; INTEGER I;                                                %110-   17002055 C  0119:0002:1
      BEGIN                                                                %110-   17002060 C  0119:0002:1
        INTEGER J;                                                         %110-   17002065 C  0119:0002:1
                                                                                 START OF SEGMENT **********  121
        FOR J := 1 STEP 1 UNTIL 3 DO                                       %110-   17002070 C  0121:0000:0
          TIMINGS[I,J] := TIME(J) - TIMINGS[I,J];                          %110-   17002075 C  0121:0001:2
      END;                                                                 %110-   17002080 C  0121:0007:2
                                                                          121 IS   10 LONG, NEXT SEG  119
      CLOSE(CARD,RELEASE);                                                 %108-   17002490 C  0119:0002:1
      CLOSE(TAPE,RELEASE);                                                 %108-   17002500 C  0119:0004:1
                          LOCK(NEWTAPE);                                   %108-   17002510 C  0119:0006:0
      WRITE(LINE[PAGE]);                                                   %108-   17002520 C  0119:0008:0
      SAVETIMES(0); % SAVE TIMES FOR START OF IDENTIFIER SORT,             %110-   17002525 C  0119:0012:0
    FOR XREFPT:=XREFPT STEP 1 UNTIL 29 DO XREFAY2[XREFPT]:=100000000;              17003000 C  0119:0012:1
    WRITE(DSK2,30,XREFAY2[*]);                                              %DFB   17004000 C  0119:0017:3
      TOTALNO := XLUN; % REMEMBER NUMBER OF IDENTIFIERS,                   %110-   17004500 C  0119:0021:3
        XREFPT←XLUN←0;                                                     %DFB   17004600 C  0119:0022:1
      FOR I := 0 STEP 1 UNTIL 8191 DO                                      %110-   17004700 C  0119:0023:3
        PUT(0,I);                                                          %110-   17004710 C  0119:0026:0
          BEGIN                                                            %DFB   17005000 C  0119:0029:2
          BOOLEAN PROCEDURE INPUT1(A);                                     %DFB   17006000 C  0119:0029:2
                                                                                 START OF SEGMENT **********  122
            ARRAY A[0];                                                    %DFB   17007000 C  0122:0000:0
            BEGIN                                                          %DFB   17008000 C  0122:0000:0
            LABEL L,EOF;                                                   %DFB   17009000 C  0122:0000:0
```

```
                                                                         START OF SEGMENT **********  123
          READ(DSK1,10,A[*])[EOF];                              %DFB         17010000 C  0123:0000:0
          GO TO L;                                              %DFB         17011000 C  0123:0005:2
          EOF: INPUT1:=TRUE;                                    %DFB         17012000 C  0123:0005:3
               REWIND(DSK1);                                    %DFB         17013000 C  0123:0006:1
          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
      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
     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
     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
           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
           ARRAY A[0];                                          %DFB         17042500 C  0122:0045:2
```

```
            HVS1(A);                                                                %DFB          17042600 C   0122:0045:2
      XL.UN:=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
            7000 ELSE 10000);                                                       %110-         17045100 C   0122:0070:1
END;                                                                                %DFB          17046000 C   0122:0072:1
                                                                                                 122 IS   79 LONG, NEXT SEG  119
BEGIN                                                                               %DFB          17047000 C   0119:0031:2
 ARRAY IDTYPE[0:(IDMAX+4)*4-1];                                                     %112-         17047100 C   0119:0031:2
                                                                                                START OF SEGMENT **********  124
 STREAM PROCEDURE SETUPHEADING(S,D,SEG,SEQNO,FWDTOG,LBLTOG,                         %110-         17047200 C   0124:0005:2
                               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");                            %110-         17049430 C   0124:0017:3
             DS := 10 LIT " PARAMETER");                                            %110-         17049440 C   0124:0020:0
   DS := 18 LIT "  --  DECLARED AT ";                                              %110-         17049500 C   0124:0022:0
   SI := LOC SEQNO;                                                                 %110-         17050400 C   0124:0024:1
   DS := 8 DEC;                                                                     %110-         17050500 C   0124:0024:1
   FWDTOG (DS := 17 LIT "  --  FORWARD AT ";                                       %110-         17050600 C   0124:0025:2
           SI := LOC FWDSEQNO;                                                      %110-         17050700 C   0124:0028:1
           DS := 8 DEC);                                                            %110-         17050800 C   0124:0028:1
   LBLTOG (DS := 16 LIT "  --  OCCURS AT ";                                        %110-         17050900 C   0124:0029:2
           SI := LOC FWDSEQNO;                                                      %110-         17051000 C   0124:0032:1
           DS := 8 DEC);                                                            %110-         17051100 C   0124:0032:1
 END OF SETUPHEADING;                                                               %110-         17051200 C   0124:0033:2
                                                                                    %110-         17051300 C   0124:0033:3
 STREAM PROCEDURE ADDASEQNO(SEQNO,N,STARS,D);                                       %110-         17051400 C   0124:0033:3
   VALUE SEQNO,N,STARS;                                                             %110-         17051500 C   0124:0033:3
 BEGIN                                                                              %110-         17051600 C   0124:0033:3
   DI := D;                                                                         %110-         17051700 C   0124:0034:0
   DI := DI + 8;                                                                    %110-         17051800 C   0124:0034:0
   N (DI := DI + 10);                                                               %110-         17051900 C   0124:0034:1
   STARS(DI := DI - 1; DS := LIT "*");                                              %110-         17052000 C   0124:0036:0
   SI := LOC SEQNO;                                                                 %110-         17052100 C   0124:0038:0
   DS := 8 DEC;                                                                     %110-         17052200 C   0124:0038:0
   DS := LIT " ";                                                                   %110-         17052300 C   0124:0038:1
   STARS (DI := DI - 1; DS := LIT "*");                                             %110-         17052400 C   0124:0039:2
 END;                                                                               %110-         17052500 C   0124:0041:2
 STREAM PROCEDURE BLANKET(D);                                                       %110-         17052600 C   0124:0041:2
 BEGIN                                                                              %110-         17052700 C   0124:0041:2
   DI := D;                                                                         %110-         17052800 C   0124:0042:0
   DS := 8 LIT " ";                                                                 %110-         17052900 C   0124:0042:0
   SI := D;                                                                         %110-         17053000 C   0124:0043:3
   DS := 16 WDS;                                                                    %110-         17053100 C   0124:0043:3
 END OF BLANKET;                                                                    %110-         17053200 C   0124:0044:0
```

```
PROCEDURE PRINTXREFSTATISTICS;
BEGIN
  SWITCH FORMAT STATS :=

    (///, "CROSS REFERENCE STATISTICS", /,
          "----- --------- ----------", /),
    ("PHASE ONE - SORT",I6," IDENTIFIERS"),
    ("PHASE TWO - SORT",I7," REFERENCES"),
    ("PHASE THREE - PRINT CROSS REFERENCE (",I7," LINES)"),
    (X5,I4,":",2I1," ELAPSED TIME (MIN:SEC)"),
    (X5,I4,":",2I1," PROCESSOR TIME"),
    (X5,I4,":",2I1," I/O TIME",/);

  INTEGER I,J,K;
  WRITE(LINE,STATS[0]);
  FOR I := 0 STEP 1 UNTIL 2 DO
    BEGIN
      WRITE(LINE,STATS[I+1],TIMINGS[I,0]);
      FOR J := 1 STEP 1 UNTIL 3 DO
        BEGIN
          K := (TIMINGS[I,J] + 30) DIV 60; % ROUND TO NEAREST SECON
          WRITE(LINE,STATS[J+3],K DIV 60,(K:=K MOD 60) DIV 10,
                                                K MOD 10);
        END;
    END;
END PRINTXREFSTATISTICS;

  DEFINE REFCOUNT = TIMINGS[1,0]#; % NUMBER OF REFERENCES SORTED.
  BOOLEAN FIRSTTIME; % TRUE ON FIRST CALL OF OUTPUT PROCEDURE.
  ARRAY PAY[0:17];
  REAL LASTADDRESS;
  BOOLEAN PROCEDURE INPUT2(A);
  ARRAY A[0];
    BEGIN
    LABEL L,EOF;

    DEFINE I = LASTADDRESS#;
    IF XREFPT:=XREFPT+1=30 THEN
      BEGIN
      READ(DSK2,30,XREFAY2[*])[EOF];
      XREFPT:=0;
      END;
  IF ( I  :=XREFAY2[XREFPT]).[21:27] GTR 99999999 THEN GO TO EOF;
  A[0] := I & NEWID[I.REFIDNOF] REFIDNOF;
  REFCOUNT := REFCOUNT + 1;
  GO TO L;
  EOF: INPUT2:=TRUE;
       BLANKET(PAY);
  XREFAY1[8] := XREFPT := LASTADDRESS := 0;
  FILL IDTYPE[*] WITH
    "UNKNOWN.                    ", % 0

    "STREAM LABEL.               ", % 1
    "STREAM VARIABLE.            ", % 2
    "DEFINE.                     ", % 3
    "LIST.                       ", % 4
    "FORMAT.                     ", % 5
```

```
%110-        17053300 C  0124:0044:0
%110-        17053400 C  0124:0044:0
%110-        17053500 C  0124:0044:0
      START OF SEGMENT **********  125
      START OF SEGMENT **********  126
%110-        17053600 C  0126:0000:0
%110-        17053700 C  0126:0000:0
%110-        17053800 C  0126:0000:0
%110-        17053900 C  0126:0000:0
%110-        17054000 C  0126:0000:0
%110-        17054100 C  0126:0000:0
%110-        17054200 C  0126:0000:0
%110-        17054300 C  0126:0000:0
    126 IS   87 LONG, NEXT SEG  125
%110-        17054400 C  0125:0000:0
%110-        17054500 C  0125:0000:0
%110-        17054600 C  0125:0004:0
%110-        17054700 C  0125:0005:2
%110-        17054800 C  0125:0005:2
%110-        17054900 C  0125:0015:2
%110-        17055000 C  0125:0016:0
             17055010 C  0125:0016:0
%110-        17055020 C  0125:0018:1
%110-        17055025 C  0125:0030:1
%110-        17055030 C  0125:0036:1
%110-        17055100 C  0125:0038:1
%110-        17055200 C  0125:0041:2
    125 IS   44 LONG, NEXT SEG  124
             17069300 C  0124:0044:0
             17069400 C  0124:0044:0
 %DFB        17069500 C  0124:0044:0
%110-        17069600 C  0124:0048:0
 %DFB        17070000 C  0124:0048:0
 %DFB        17071000 C  0124:0048:0
 %DFB        17072000 C  0124:0048:0
 %DFB        17073000 C  0124:0048:0
      START OF SEGMENT **********  127
%110-        17073100 C  0127:0000:0
 %DFB        17074000 C  0127:0000:0
 %DFB        17075000 C  0127:0001:3
 %DFB        17076000 C  0127:0002:0
 %DFB        17077000 C  0127:0007:3
 %DFB        17078000 C  0127:0008:0
             17079000 C  0127:0008:0
%110-        17080000 C  0127:0010:1
%110-        17080100 C  0127:0028:0
 %DFB        17081000 C  0127:0031:3
 %DFB        17082000 C  0127:0033:2
 %DFB        17083000 C  0127:0033:3
%110-        17084000 C  0127:0035:2
%110-        17084010 C  0127:0037:3
%110-        17084020 C  0127:0038:0
      START OF SEGMENT **********  128
%110-        17084030 C  0127:0039:2
%110-        17084040 C  0127:0039:2
%110-        17084050 C  0127:0039:2
%110-        17084060 C  0127:0039:2
%110-        17084070 C  0127:0039:2
```

```
"SWITCH FORMAT.                ", % 6     %110-      17084080 C  0127:0039:2
"REAL SUBROUTINE.              ", % 7     %110-      17084090 C  0127:0039:2
"SUBROUTINE.                   ", % 8     %110-      17084100 C  0127:0039:2
"SWITCH LABEL.                 ", % 9     %110-      17084110 C  0127:0039:2
"PROCEDURE.                    ", % 10    %110-      17084120 C  0127:0039:2
"INTRINSIC.                    ", % 11    %110-      17084130 C  0127:0039:2
"STREAM PROCEDURE.             ", % 12    %110-      17084140 C  0127:0039:2
"BOOLEAN STREAM PROCEDURE.     ", % 13    %110-      17084150 C  0127:0039:2
"REAL STREAM PROCEDURE.        ", % 14    %110-      17084160 C  0127:0039:2
"INTEGER STREAM PROCEDURE.     ", % 15    %110-      17084170 C  0127:0039:2
"INTEGER STREAM PROCEDURE.     ", % 16    %110-      17084180 C  0127:0039:2
"BOOLEAN PROCEDURE.            ", % 17    %110-      17084182 C  0127:0039:2
"REAL PROCEDURE.               ", % 18    %110-      17084184 C  0127:0039:2
"INTEGER PROCEDURE.            ", % 19    %110-      17084186 C  0127:0039:2
"INTEGER PROCEDURE.            ", % 20    %110-      17084188 C  0127:0039:2
"BOOLEAN.                      ", % 21    %110-      17084190 C  0127:0039:2
"REAL.                         ", % 22    %110-      17084200 C  0127:0039:2
"INTEGER.                      ", % 23    %110-      17084210 C  0127:0039:2
"INTEGER.                      ", % 24    %110-      17084220 C  0127:0039:2
"BOOLEAN ARRAY.                ", % 25    %110-      17084230 C  0127:0039:2
"REAL ARRAY.                   ", % 26    %110-      17084240 C  0127:0039:2
"INTEGER ARRAY.                ", % 27    %110-      17084250 C  0127:0039:2
"INTEGER ARRAY.                ", % 28    %110-      17084260 C  0127:0039:2
"                              ", % 29    %110-      17084270 C  0127:0039:2
"NAME.                         ", % 30    %110-      17084280 C  0127:0039:2
"INTEGER NAME.                 ", % 31    %110-      17084290 C  0127:0039:2
"LABEL.                        ", % 32    %112-      17084300 C  0127:0039:2
  "FIELD.                      "; % 33 (CLASS = 125)  %112-      17084400 C  0127:0039:2
                                                                  128 IS  136 LONG, NEXT SEG  127
L:                                                      %DFB       17085000 C  0127:0039:2
END;                                                    %DFB       17086000 C  0127:0040:0
                                                                  127 IS   45 LONG, NEXT SEG  124
PROCEDURE OUTPUT2(B,A);                                 %DFB       17087000 C  0124:0048:0
  VALUE B;                                              %DFB       17088000 C  0124:0048:0
  BOOLEAN B;                                            %DFB       17089000 C  0124:0048:0
  ARRAY A[0];                                           %DFB       17090000 C  0124:0048:0
  BEGIN      DEFINE PRINTER=LINE#;                      %DFB       17091000 C  0124:0048:0
                                                                  START OF SEGMENT **********  129
LABEL EOF2, SKIP;                                       %110-      17091100 C  0129:0000:0
OWN BOOLEAN B2, FWDTOG, LBLTOG, WAITINGFORFWDREF;       %110-      17091110 C  0129:0000:0
DEFINE MATCH(A,B) = REAL(BOOLEAN(A) EQV BOOLEAN(B)) =   %110-      17091115 C  0129:0000:0
                    REAL(NOT FALSE)#;                   %110-      17091116 C  0129:0000:0
REAL I;                                                 %110-      17091120 C  0129:0000:0
DEFINE LINECOUNT = TIMINGS[2,0]#; % NUMBER OF LINES PRINTED.       17091140 C  0129:0000:0
OWN REAL FWDSEQNO;                                      %110-      17091150 C  0129:0000:0
IF FIRSTTIME THEN % PRINT HEADINGS AND SAVE TIMINGS.    %110-      17091155 C  0129:0000:0
  BEGIN                                                 %110-      17091160 C  0129:0000:1
    FIRSTTIME := FALSE;                                 %110-      17091162 C  0129:0001:2
    TIME1 := TIME(1);                                   %110-      17091165 C  0129:0002:0
    DATIME;                                             %110-      17091170 C  0129:0003:2
    UPDATETIMES(1);                                     %110-      17091175 C  0129:0003:3
    SAVETIMES(2); % SAVE TIMES FOR START OF XREF PRINT. %110-      17091180 C  0129:0004:1
  END;                                                  %110-      17091200 C  0129:0005:2
IF NOT B2 THEN                                          %110-      17091210 C  0129:0005:2
  IF B THEN % END OF SORT - LIST OUT REST OF SEQ. NO.    %110-      17091300 C  0129:0006:0
    IF XREFPT # 0 THEN % WE GOT SOME TO LIST OUT        %110-      17091400 C  0129:0006:1
      BEGIN                                             %110-      17091500 C  0129:0008:0
```

```
      WRITE(LINE[DBL],15,PAY[*]);                                %110-      17091510 C  0129:0008:1
      LINECOUNT := LINECOUNT + 1;                                %110-      17091520 C  0129:0013:2
    END                                                          %110-      17091530 C  0129:0016:1
  ELSE % NOTHING TO LIST OUT                                     %110-      17091600 C  0129:0016:1
ELSE % NOT END OF SORT                                           %110-      17091700 C  0129:0016:1
  IF NOT MATCH(LASTADDRESS,A[0]) AND A[0],REFIDNOF ≠ 0 AND       %110-      17091800 C  0129:0017:2
     A[0],REFIDNOF ≥ XREFAY1[8],IDNOF THEN                       %110-      17091900 C  0129:0021:2
    IF A[0],TYPEREF = FORWARDREF THEN %                          %110-      17092000 C  0129:0023:3
      WAITINGFORFWDREF := TRUE                                   %110-      17092100 C  0129:0025:3
    ELSE                                                         %110-      17092200 C  0129:0026:1
      IF A[0],TYPEREF = LBLREF THEN %                            %110-      17092300 C  0129:0027:2
        BEGIN                                                    %110-      17092400 C  0129:0029:2
          LBLTOG := TRUE;                                        %110-      17092500 C  0129:0029:3
          FWDSEQNO := A[0],SEQNOF;                               %110-      17092600 C  0129:0030:1
        END                                                      %110-      17092700 C  0129:0032:1
      ELSE                                                       %110-      17092800 C  0129:0032:1
        IF A[0],TYPEREF = DECLREF THEN                           %110-      17092900 C  0129:0032:1
          IF WAITINGFORFWDREF THEN % THIS MUST BE IT             %110-      17093000 C  0129:0034:1
            BEGIN                                                %110-      17093100 C  0129:0035:3
              WAITINGFORFWDREF := FALSE;                         %110-      17093200 C  0129:0036:0
              FWDTOG := TRUE;                                    %110-      17093300 C  0129:0037:2
              FWDSEQNO := A[0],SEQNOF;                           %110-      17093400 C  0129:0038:0
            END                                                  %110-      17093500 C  0129:0039:3
          ELSE % ITS A NORMAL DECLARATION - NOT FORWARD          %110-      17093600 C  0129:0039:3
            BEGIN                                                %110-      17093700 C  0129:0039:3
              IF A[0],REFIDNOF > XREFAY1[8],IDNOF THEN           %110-      17093850 C  0129:0040:0
                DO                                               %110-      17093900 C  0129:0042:1
                  READ(DSK1,10,XREFAY1[*]) [EOF2]                %110-      17093950 C  0129:0043:2
                UNTIL                                            %110-      17094000 C  0129:0047:2
                  A[0],REFIDNOF ≤ XREFAY1[8],IDNOF;              %110-      17094050 C  0129:0048:0
    IF A[0], REFIDNOF < XREFAY1[8],IDNOF THEN                    %110-      17094100 C  0129:0051:2
      GO TO SKIP;                                                %110-      17094150 C  0129:0053:2
    IF XREFPT > 0 THEN % THERE IS STUFF TO PRINT                 %110-      17094200 C  0129:0053:3
     BEGIN                                                       %110-      17094240 C  0129:0054:1
      IF SINGLTOG THEN                                           %110-      17094250 C  0129:0055:2
        WRITE(LINE,15,PAY[*])                                    %110-      17094300 C  0129:0055:3
      ELSE                                                       %110-      17094350 C  0129:0059:2
        WRITE(LINE[DBL],15,PAY[*]);                              %110-      17094400 C  0129:0060:1
      LINECOUNT := LINECOUNT + 1;                                %110-      17094410 C  0129:0065:3
     END                                                         %110-      17094420 C  0129:0069:2
    ELSE                                                         %110-      17094450 C  0129:0069:2
      IF NOT SINGLTOG THEN                                       %110-      17094500 C  0129:0069:2
        WRITE(LINE);                                             %110-      17094550 C  0129:0070:1
    XREFPT := 0;                                                 %110-      17094600 C  0129:0075:2
    BLANKET(PAY[*]);                                             %110-      17094650 C  0129:0076:0
     SETUPHEADING(XREFAY1[*],PAY[*],XREFAY1[8],                  %110-      17094700 C  0129:0077:2
       SEGNOF,A[0],SEQNOF,FWDTOG,LBLTOG,                         %110-      17094800 C  0129:0079:2
      FWDSEQNO,IDTYPE[(IF (I :=                                  %110-      17094900 C  0129:0081:3
      XREFAY1[9],CLASS) > IDMAX THEN IF I =                      %112-      17095000 C  0129:0082:0
      FIELDID THEN 33 ELSE 0 ELSE I) × 4],                       %112-      17095100 C  0129:0085:2
      REAL(I ≥ BOOID AND XREFAY1[9],[9:2] = 1),                  %112-      17095300 C  0129:0088:1
      REAL((I ≥ BOOID OR I = LOCLID) AND BOOLEAN                 %112-      17095310 C  0129:0091:2
      (XREFAY1[9],[9:1])), XREFAY1[9],[10:1]);                   %112-      17095320 C  0129:0092:1
    FWDTOG := LBLTOG := FALSE;                                   %110-      17095400 C  0129:0095:2
    WRITE(LINE,15,PAY[*]);                                       %110-      17095500 C  0129:0097:2
    LINECOUNT := LINECOUNT + 1;                                  %110-      17095510 C  0129:0101:3
    BLANKET(PAY[*]);                                             %110-      17095550 C  0129:0105:2
END                                                              %110-      17095600 C  0129:0106:0
```

```
                ELSE % IT MUST BE A NORMAL REFERENCE                    %110-
                  IF A[0].SEQNOF ≠ LASTADDRESS.SEQNOF THEN              %110-
                    BEGIN                                               %110-
                      ADDASEQNO(A[0].SEQNOF,XREFPT,A[0].[5:1],
                                PAY[*]);                                %110-
                      IF (XREFPT := XREFPT + 1) = 11 THEN %FULL
                        BEGIN                                           %110-
                          WRITE(LINE,15,PAY[*]);                        %110-
                          LINECOUNT := LINECOUNT + 1;                   %110-
                          XREFPT := 0;                                  %110-
                          BLANKET(PAY[*]);                              %110-
                        END                                             %110-
                    END
                  ELSE % REFERENCE TO SAME SEQ, NO, SKIP IT             %110-
    ELSE % THIS IS A REFERENCE TO THE SAME SEQ, NO. - SKIP
 ELSE % HIT END OF IDENTIFIER FILE - JUST SKIP OVER REFERENCES
   EOF2: B2 := TRUE; %  SO SORT CAN GO TO NORMAL EOJ                    %110-
 IF NOT B THEN SKIP: LASTADDRESS := A[0];                               %110-
END OF OUTPUT2;                                                         %110-

                                                                                129 IS   135 LONG, NEXT SEG  124
PROCEDURE HV2(A);                                                        %DFB
   ARRAY A[0];                                                           %DFB
   A[0] := 3"777777777777777"; % BIGGEST FLOATING PT, NO.                %110-
BOOLEAN PROCEDURE COMP2(A,B);                                            %DFB
   ARRAY A,B[0];                                                         %DFB
      COMP2 := IF A[0].REFIDNOF < B[0].REFIDNOF THEN % DIF IDS
                 TRUE                                                    %110-
               ELSE                                                      %110-
                 IF A[0].REFIDNOF = B[0].REFIDNOF THEN                   %110-
                   IF A[0].[1:4] LSS B[0].[1:4] THEN                     %110-
                     TRUE                                                %110-
                   ELSE                                                  %110-
                     IF A[0].[1:4] = B[0].[1:4] THEN                     %110-
                       IF A[0].SEQNOF < B[0].SEQNOF THEN                 %110-
                         TRUE                                            %110-
                       ELSE                                              %110-
                         IF A[0].SEQNOF = B[0].SEQNOF THEN               %110-
                           BOOLEAN(A[0].[5:1])                           %110-
                         ELSE                                            %110-
                           FALSE                                         %110-
                     ELSE                                                %110-
                       FALSE                                             %110-
                 ELSE                                                    %110-
                   FALSE;                                                %110-
     SAVETIMES(1); % SAVE TIMES FOR START OF REFERENCES SORT             %110-
     FIRSTTIME := TRUE; % LET OUTPUT PROCEDURE KNOW ABOUT FIRST CAL
     XREFPT:=29;       REWIND(DSK2);                                     %DFB
     SORT(OUTPUT2,INPUT2,0,HV2,COMP2,1,6000);                            %110-
    UPDATETIMES(2); % UPDATE TIMES FOR PRINTING CROSS REFERENCE          %110-
    PRINTXREFSTATISTICS;                                                 %110-
END;                                                                     %DFB

                                                                                124 IS   108 LONG, NEXT SEG  119
END;                                                                     %DFB
                                                                                119 IS    36 LONG, NEXT SEG    3
                END MAIN BLOCK;                                          %108-
                                                                                  3 IS   651 LONG, NEXT SEG    2
END.                                                                     %DFB
```

```
                                                                        17095700 C  0129:0106:0
                                                                        17095750 C  0129:0106:0
                                                                        17095800 C  0129:0109:2
                                                                        17095900 C  0129:0109:3
                                                                        17096000 C  0129:0112:0
                                                                        17096100 C  0129:0113:2
                                                                        17096200 C  0129:0114:1
                                                                        17096300 C  0129:0115:2
                                                                        17096350 C  0129:0119:3
                                                                        17096400 C  0129:0123:2
                                                                        17096450 C  0129:0124:0
                                                                        17096500 C  0129:0125:2
                                                                        17096550 C  0129:0125:2
                                                                        17096575 C  0129:0125:2
                                                                        17096600 C  0129:0125:2
                                                                        17096700 C  0129:0125:3
                                                                        17096800 C  0129:0126:0
                                                                        17096850 C  0129:0128:0
                                                                        17096900 C  0129:0130:0
                                                                        17112000 C  0124:0048:0
                                                                        17113000 C  0124:0048:0
                                                                        17114000 C  0124:0048:0
                                                                        17115000 C  0124:0052:0
                                                                        17116000 C  0124:0052:0
                                                                        17117000 C  0124:0052:0
                                                                        17117100 C  0124:0054:0
                                                                        17117200 C  0124:0055:2
                                                                        17117300 C  0124:0055:3
                                                                        17117400 C  0124:0057:3
                                                                        17117500 C  0124:0060:1
                                                                        17117600 C  0124:0061:2
                                                                        17117700 C  0124:0061:3
                                                                        17117702 C  0124:0064:0
                                                                        17117704 C  0124:0066:1
                                                                        17117706 C  0124:0067:3
                                                                        17117708 C  0124:0068:0
                                                                        17117710 C  0124:0070:0
                                                                        17117712 C  0124:0071:3
                                                                        17117714 C  0124:0072:0
                                                                        17117720 C  0124:0072:1
                                                                        17117730 C  0124:0073:2
                                                                        17117800 C  0124:0073:2
                                                                        17117900 C  0124:0073:3
                                                                        17117910 C  0124:0076:0
                                                                        17117920 C  0124:0077:3
                                                                        17118000 C  0124:0078:1
                                                                        17119000 C  0124:0081:2
                                                                        17119100 C  0124:0102:1
                                                                        17119200 C  0124:0103:2
                                                                        17120000 C  0124:0103:3
                                                                        17121000 C  0119:0032:0
                                                                        17121500 C  0003:0647:2
                                                                        17122000 C  0002:0038:0
```

```
                                                                    2 IS    42 LONG, NEXT SEG     1
                                                                    1 IS     2 LONG, NEXT SEG     0
                                                                  143 IS    69 LONG, NEXT SEG     0

NUMBER OF ERRORS DETECTED =   0.  COMPILATION TIME =  380 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.


 LABEL,  000000000LINE   001772312 COMPILE ESPOL/DISK ALGOL LIBRARY                    ALGOL  /ESPOL
```

LABEL 000000000LINE 001772317 EXECUTE PATCH/MERGE

PATCH /MERGE

```
? EXECUTE PATCH/MERGE

            PACKET     21
            INPUT      26 CARDS FROM CRA
            TIME     1303
            DATE    77231 FRIDAY, 08/19/77



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

            #NO MESSAGES TODAY

         ? EXECUTE PATCH/MERGE
             ? FILE CARD= PMCARD
         ? DATA PMCARD
                  5:PATCH/MERGE= 1 BOJ 1303 02/06/75
                  CDA IN PMCARD:PATCH/MERGE= 1
                  PBD0022 OUT 011 LINE:PATCH/MERGE= 1
                  DKA OUT SER MASTERF U000000:PATCH/MERGE= 1
                  DKA IN SER PATCH FSPOL:PATCH/MERGE= 1
                  DKA OUT SER PATCHDF U000000:PATCH/MERGE= 1
                  DKA REL PATCH ESPOL:PATCH/MERGE= 1
                  CDA REL PMCARD:PATCH/MERGE= 1
         ? END
                  DKA REL PATCHDF U000000:PATCH/MERGE= 1
                  PBD0022 REL 011 LINE 1809:PATCH/MERGE= 1
                  DKA REL MASTERF U000000:PATCH/MERGE= 1
                  PATCH/MERGE= 1 EOJ 1304
                  FOR PATCH/MERGE= 1: PROCESS= 53 SECS, IO= 37 SECS, OLAY= 1
                  PKT#0021 REMOVED
```

BURROUGHS B-5700 PATCH/MERGE PROGRAM MARK XVI.0.00 FRIDAY, 08/19/77, 1:03 PM.

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

```
$@CARD CONFLICTS LIST1 MERGE ZIP
$, 18 PATCHES FOR ESPOL

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

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

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

      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[*1)[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 RES06320000 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 ELCLASS = 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(XREFINFO1,XREFINFO2)=                                            01001750 P  108
          XINFO(XREFINFO1,(XREFINFO2)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
            ST:=SQ;                                                             %DFB13301610 P  108
            DS:=8 DEC;                                                          %DFB13301640 P  108
            ST:=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
        FNDOFITALL;                                                             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 FOF2: 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

$ VOINT 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 IDENTIFIER01007470 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 REFERENCE 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); % FLBAT WORD                                 02001820 P  110
  WRITE(DSK1,10,XREFAY1[*]);                                              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[FLBAT[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
```

```
                    XREFITCTALL,REMEMBERSEQNO,ASSIGNREF); % ASSIGNMENT TO15301100 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
                          XREFITCTALL,REMEMBERSEQNO,ASSIGNREF); % PARTIAL15342300 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
              SI := A;                                                   17033400 P  110
              DI := 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 & NEWID[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
       BLANKET(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) x 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
       BLANKET(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
ELSE % 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
                                ELSE                                             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 FSPOL CONTAINS 3 CARDS.                          00000000 P  111

    IF MERGETOG THEN % INDICATE NAME OF SOURCE FILE.                    01835600 P  111
      WRITECLINE,<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 FSPOL 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
0CT1310157730000002, "3SCS00",                              %652             09128600 P  112
0CT0500000250000000, "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 = LFTRKET 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 STEP1 = 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

$ VOID 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
$ VOID 15127001                                                                 15085150 P  114
              IF TALL.CLASS ≤ INTID THEN % SIMPLE VAR OR FORMAL NAME            15085200 P  114
```

```
BEGIN                                                          15085400 P  114
  IF STEPI ≠ 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 STEPT ≠ LEFTBRKET 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
            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
                EMITI(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 STEPI = 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) ≥ EQVOP 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
                  EMITI(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
        LINK,                   % HOLDS RELATIVE ADDRESS OF FIRST OF
                                % BRANCHES THAT MUST BE FIXED UP TO
                                % BRANCH TO THE RESUME POINT.
        ADR,                    % HOLDS RELATIVE ADDRESS OF THE BRANCH
                                % AROUND THE CASE STATEMENTS TO THE
                                % BRANCH TABLE. THIS BRANCH GETS FIXED
                                % UP WHEN WE FIND WHERE THE BRANCH TABLE
                                % GOES.
        N,                      % COUNT OF NUMBER OF CASE STATEMENT
                                % ENCOUNTERED.
        ENDOFIT,                % ADDRESS OF RESUME POINT
        J,                      % TEMPORARY
        K;                      % TEMPORARY
                                %
      BOOLEAN
        GOTOG;                  % TRUE IF WE ARE COMPILING A SIMPLE
                                % GO TO.
      LABEL XIT;
%
%
      STEPIT;    % STEP OVER "CASE"
      AEXP;      % GENERATE CODE FOR CASE INDEX
      IF STEPI ≠ BEGINV THEN % NOTICE WE JUST JUMPED OVER "OF"
        BEGIN
          ERR(70); GO TO XIT;
        END;
      EMIT(0);  % GENERATE DUMMY BRANCH TO BRANCH TABLE. WILL FIX
      EMITO(BFW);  %     IT UP LATER.
      ADR := L;
      WHILE STEPI ≠ ENDV DO
        BEGIN   % PROCESSING CASE STATEMENTS
          ERRORTOG := TRUE;
          IF ELCLASS = SEMICOLON THEN % NULL STATEMENT, NO CODE NEEDED
            N := N + 1  % THIS LEAVES A ZERO IN CASEADDRESS[N]
          ELSE
            BEGIN
              CASEADDRESS[N] := L; % REMEMBER BEGINNING ADDRESS OF
              N := N + 1;          % THIS CASE.
              IF (GOTOG := SIMPGO) THEN
                ELBAT[I:=I-1] := ELCLASS := GOV; % REMEMBER IF SIMPLE
              STMT; % PROCESS THE STATEMENT
              IF ELCLASS = SEMICOLON THEN
                IF NOT GOTOG THEN % GENERATE DUMMY BRANCH TO RESUME
                  BEGIN
                    EMIT(LINK);
                    EMITO(BFW);
                    LINK := L;
                  END
                ELSE % SIMPLE GO TO, NO CODE TO GENERATE
              ELSE
                IF ELCLASS ≠ ENDV THEN
                  ERR(71);
            END;
        END OF WHILE LOOP;
      ENDTOG := TRUE;      % SKIP OVER COMMENT AFTER END
      DO STOPDEFINE := TRUE UNTIL
```

```
07646350 P  115
07646355 P  115
07646360 P  115
07646365 P  115
07646370 P  115
07646375 P  115
07646380 P  115
07646385 P  115
07646390 P  115
07646395 P  115
07646400 P  115
07646405 P  115
07646410 P  115
07646415 P  115
07646420 P  115
07646425 P  115
07646430 P  115
07646435 P  115
07646436 P  115
07646440 P  115
07646445 P  115
07646450 P  115
07646455 P  115
07646456 P  115
07646457 P  115
07646458 P  115
07646459 P  115
07646460 P  115
07646465 P  115
07646470 P  115
07646475 P  115
07646480 P  115
07646485 P  115
07646490 P  115
07646495 P  115
07646500 P  115
07646505 P  115
07646510 P  115
07646515 P  115
07646520 P  115
07646525 P  115
07646530 P  115
07646535 P  115
07646540 P  115
07646545 P  115
07646550 P  115
07646555 P  115
07646560 P  115
07646565 P  115
07646570 P  115
07646575 P  115
07646580 P  115
07646585 P  115
07646590 P  115
07646595 P  115
07646601 P  115
07646602 P  115
```

```
      STEP; ≤ ENDV AND FLCLASS ≥ 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
    EMIT0(DUP);           % GENERATE CODE TO MULTIPLY INDEX               07646610 P  115
    EMIT0(ADD);           % BY TWO AND BRANCH INTO BRANCH TABLE           07646615 P  115
    EMIT0(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,1);                                                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
    OCT0610000000000000, "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 XVI.0.116                                   00001030 P  117
%                         MAY  9, 1977                                    00001040 P  117
"XVI.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

$: 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 ESPOL 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(LINEPAGE])#;                         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

     0&T0500000250000002, "5FIELD",                                %654      09128700 P  115 CONFLICTED WITH:
     0&T0500000250000000, "5FIELD",                                %654      09128700 P  112    DISCARDED

     0&T0500000250000000, "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
        SI:=SQ;                                                             %DFB13301610 P  108    VOIDED
        DS:=8 DEC;                                                          %DFB13301640 P  108    VOIDED
        SI:=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 TO            15301100 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); % PARTIAL            15342300 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  108    DISCARDED


                SORT(OUTPUT1,INPUT1,0,HV1,COMP1,10,TOTALNO    );     %DFB17045000 P  110 CONFLICTED WITH:
                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
                S1:=A;                                               %DFB17067000 P  108    VOIDED
```

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

        ARRAY IDTYPE[0:(IDMAX+4)*4-1];                                                    17047100 P  112 CONFLICTED WITH:
        ARRAY IDTYPE[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]:= I&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[*]);                                  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: B?:=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]:=549755813887;                                                           %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 =   53 SECONDS.
    I/O TIME =   34 SECONDS.


     LABEL  000000000LINE   001772317 EXECUTE PATCH/MERGE                                                    PATCH  /MERGE
```

```
? COMPILE ESPOL/DISK ALGOL LIBRARY

          PACKET    23
          INPUT   1144 CARDS FROM ZIP
          TIME    1304
          DATE   77231 FRIDAY, 08/19/77



       *** 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= 1 BOJ 1305 05/12/77
                CDB IN CARD DB:ALGOL/ESPOL= 1
                DKA IN SER SYMBOL ESPOL:ALGOL/ESPOL= 1
                PBD0024 OUT 011 LINE:ALGOL/ESPOL= 1
                DKA OUT RDM ESPOL DISK:ALGOL/ESPOL= 1
               #DUP LIBRARY ESPOL/DISK:ALGOL ESPOL= 1
                RM OK DS
               #DUP LIBRARY ESPOL/DISK:ALGOL ESPOL= 1
               +OPERATOR KEYED IN: 1RM
                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
                PBD0024 REL 011 LINE 7763:ALGOL/ESPOL= 1
                ALGOL/ESPOL= 1 EOJ 1311
                FOR ALGOL/ESPOL= 1: PROCESS= 258 SECS, IO= 117 SECS, OLAY= 4
                PKT#0023 REMOVED
```