LABEL 000000000PRINTER00175100CC EX OBJECT/READ;FILE SOURCEFILE=SYMBOL/USER;END↵

OBJECT /READ

SYMBOL/USER

```
**********************************************************************00010000
%       USER/CANDE  SOURCE FILE  12-70.                                %001-00010100
  COMMENT: * TITLE: B5500/B5700 MARK XIV SYSTEM RELEASE                *    00010110
           * FILE ID: SYMBOL/USER  TAPE ID: SYMBOL2/FILE000            *    00010111
           * THIS MATERIAL IS PROPRIETARY TO BURROUGHS CORPORATION      *    00010112
           * AND IS NOT TO BE REPRODUCED, USED, OR DISCLOSED           *    00010113
           * EXCEPT IN ACCORDANCE WITH PROGRAM LICENSE OR UPON         *    00010114
           * WRITTEN AUTHORIZATION OF THE PATENT DIVISION OF           *    00010115
           * BURROUGHS CORPORATION, DETROIT, MICHIGAN 46232            *    00010116
           *                                                           *    00010117
           * COPYRIGHT (C) 1971, 1972 BURROUGHS CORPORATION            *    00010118
           *  AA320206   AA386657                                      *;   00010119
%**********************************************************************00010200
COMMENT                                                                 %001-00010300
THIS PROGRAM CREATES THE FILE "USERS/CANDE" WHICH IS USED BY            %001-00010400
CANDE/TSHARER TO OBTAIN THE SECURITY AND ACCOUNTING INFORMATION         %001-00010500
ASSOCIATED WITH EACH USER.                                              %001-00010600
THE FILE IS COMPRISED OF THREE SECTIONS:                                %001-00010700
    A) SEGMENT ZERO, WHICH CONTAINS THE RECORD COUNT AND DATE OF        %001-00010800
       LAST ALTERATION OF THE FILE.  THE NEXT AVAILABLE SEGMENT         %001-00010900
       POSITION IS KEPT IN WORD 1 OF SEGMENT ZERO, AND THE DATE OF      %001-00011000
       LAST ALTERATION IS KEPT IN WORD 2 OF SEGMENT ZERO.               %001-00011100
    B) THE INITIAL SCRAMBLE TABLE SECTION, LOCATED IN SEGMENTS          %001-00011200
       1 THROUGH 29 OF THE FILE.  EACH SCRAMBLE SEGMENT CONSISTS        %001-00011300
       OF 15 TWO WORD ENTRIES, THE FIRST WORD CONTAINING THE USERCODE      00011400
       AND THE [33:15] FIELD OF THE SECOND WORD CONTAINING THE          %001-00011500
       RELATIVE DISK ADDRESS OF THE SECURITY SEGMENT ASSOCIATED         %001-00011600
       WITH THAT USERCODE. THE [18:15] FIELD OF WORD 1 OF EACH          %001-00011700
       SCRAMBLE SEGMENT IS A LINK TO THE NEXT SCRAMBLE SEGMENT.         %001-00011800
       WHEN AN ATTEMPT IS MADE TO ADD A NEW USERCODE TO THE SCRAMBLE       00011900
       BLOCK SECTION, AND THAT BLOCK IS ALREADY FULL, THE USER CANDE       00012000
       PROGRAM WILL CREATE A NEW SCRAMBLE BLOCK IN THE USERS/CANDE      %001-00012100
       FILE, AND WILL PLACE THE LOCATION OF THE LINKED BLOCK IN THE     %001-00012200
       [18:15] FIELD OF WORD 1 OF THE CURRENTLY ACCESSED BLOCK.         %001-00012300
       THE [18:15] FIELD OF WORD 29 OF EACH LINKED SCRAMBLE BLOCK IS       00012400
       USED AS A BACK LINK TO THE PREVIOUS SCRAMBLE BLOCK FOR           %001-00012500
       PURPOSES OF FILE MAINTENANCE.                                    %001-00012600
    C) THE BODY OF THE USERS/CANDE FILE, LOCATED IN SEGMENTS 30 AND     %001-00012700
       BEYOND.  THIS SECTION OF THE FILE IS COMPRISED OF BOTH LINKED       00012800
       SCRAMBLE BLOCKS AND INDIVIDUAL USERS SECURITY RECORDS.           %001-00012900
       WORD 29 OF EACH SECURITY RECORD CONTAINS A BACK LINK TO THE      %001-00013000
       SCRAMBLE BLOCK WHICH POINTS TO THAT RECORD. THE [33:15] FIELD       00013100
       OF WORD 29 POINTS TO THE RELATIVE DISK ADDRESS OF THE SCRAMBLE      00013200
       BLOCK, AND THE [18:15] FIELD POINTS TO THE WORD LOCATION OF      %001-00013300
       THE USERCODE WITHIN THE SCRAMBLE BLOCK.                          %001-00013400
       THE SIGN BIT OF WORD 29 OF EACH OF THESE SEGMENTS IS USED TO     %001-00013500
       INDICATE WHETHER THE SEGMENT IS A LINKED SCRAMBLE BLOCK, OR      %001-00013600
       A USERS SECURITY RECORD.  THE SIGN BIT IS ON WHEN THE SEGMENT       00013700
       IS A SCRAMBLE BLOCK, AND IT IS OFF WHEN THE SEGMENT IS A         %001-00013800
       USERS SECURITY RECORD.                                           %001-00013900
THE PROCEDURES "READUSER" AND "REMOVEUSER" ARE RESPONSIBLE FOR          %001-00014000
CREATING AND / OR ALTERING THE LINKS ASSOCIATED WITH THE VARIOUS        %001-00014100
SECTIONS OF THE FILE.                                                   %001-00014200
END COMMENT;                                                            %001-00014300
BEGIN REAL COMMON;                                                      %001-00014400
FILE USERS DISK RANDOM "USERS""CANDE" (1,30);                           %001-00014500
FILE CARD   (2,10);                                                     %001-00014600
FILE LINE   (2,15);  "PRINTER"                                          %001-00014700
FILE DSK DISK SERIAL [20:30] (2,30,300); % SCRATCH FILE FOR SORT        %001-00014800
ARRAY TABLE,USER,OPTIONS[0:30];                                         %001-00014900
```

```
ARRAY CMPS[0:11],VRBS[0:60];                                                    %001-00015000
SAVE ARRAY A[0:16],B[0:5];                                                      %001-00015100
INTEGER TABLELOC,USRINDX,APTR,AQ;                                               %001-00015200
REAL USERCODE,USERLOC,WORD;                                                     %001-00015300
REAL I,K,NEXTOPEN,COUNT,TCOUNT;                                                 %001-00015400
BOOLEAN FIRSTCARD,USERPRESENT,OPTIONSBUILT,DOLLARCARD,FOUND,ALTERED;              00015500
LABEL NEXT,EOF,COMPILERS,VERBS;                                                 %001-00015600
SWITCH FORMAT FM :=                                                             %001-00015700
   ("ERR:BAD $ CARD.",                                                          %001-00015800
   ("ERR:$ NEW MUST BE FIRST CARD.",                                            %001-00015900
   ("ERR:QUOTES REQD AROUND PARAMETERS.",                                       %001-00016000
   ("ERR:UNKNOWN OPTION IN THIS CARD.",                                         %001-00016100
   ("ERR: UNRECOGNIZABLE COMPILER NAME",                                        %001-00016200
   ("ERR: UNRECOGNIZABLE VERB NAME ");                                          %001-00016300
DEFINE      LEFTARROW      = "←"#;                                              %001-00016400
DEFINE      NUMOFCOMPILERS = 11#;                                               %001-00016500
DEFINE      NUMOFVERBS     = 49#;                                                 00016600
DEFINE      USERWORD       = USER[1]#;                                          %001-00016700
DEFINE      PASSWORD       = USER[2]#;                                          %001-00016800
DEFINE      CHARGEWORD     = USER[3]#;                                          %001-00016900
DEFINE      USERNAME       = USER[4]#;  % AND 5.                                %001-00017000
DEFINE      PHONEWORD      = USER[6]#;                                          %001-00017100
DEFINE      TIMEWORD       = USER[7]#;                                          %001-00017200
DEFINE      COMPILERWORD   = USER[8]#;                                          %001-00017300
DEFINE      VERBWORD       = USER[9]#; % AND USER[10]                           %001-00017400
DEFINE      OPTIONBITS     = USER[11]#; % SET/RESET OPTIONS                     %001-00017500
DEFINE      MONITORNAME    = USER[12]#; % MONITOR FILE NAME                    %001-00017600
DEFINE      LINKWORD       = USER[29]#;                                         %001-00017700
DEFINE      MATCH(MATCH1,MATCH2)=                                               %001-00017800
   IF REAL(BOOLEAN(MATCH1) EQV BOOLEAN(MATCH2))=REAL(NOT FALSE) THEN#;          00017900
%*********************************************************************************00018000
STREAM PROCEDURE MOVE(N,A,B); VALUE N;                                          %001-00018100
   BEGIN SI:=A; DI:=B; DS:= N WDS; END;                                         %001-00018200
%*********************************************************************************00018300
PROCEDURE PRINT;                                                                %001-00018400
%*********************************************************************************00018500
   BEGIN LABEL MAP,EXIT;                                                        %001-00018600
   FORMAT NUMOFUSERS(I6," USERS IN USERS/CANDE FILE"//),                        %001-00018700
   TABLEMAPP(X48,"***SCRAMBLE TABLE MAP***",X3,O/),                             %001-00018800
   USERSRECORDS(X53,"***USERS RECORDS***"/);                                    %001-00018900
   %*****************************************************************************00019000
   STREAM PROCEDURE TABLEMAP(TABLE,A);                                          %001-00019100
   %*****************************************************************************00019200
      BEGIN                                                                     %001-00019300
      SI:=TABLE; DI:=A;                                                         %001-00019400
      15(DS:=8LIT" "; SI:=SI+1; IF SC NEQ "0" THEN                              %001-00019500
         BEGIN DI:=DI-7; DS:=7CHR; END ELSE SI:=SI+7;                           %001-00019600
      SI:=SI+8);                                                                %001-00019700
      END STREAM PROCEDURE TABLEMAP;                                            %001-00019800
   %*****************************************************************************00019900
   STREAM PROCEDURE REFORM1(USER,A,CMPS);                                       %001-00020000
   %*****************************************************************************00020100
      BEGIN LOCAL SV1,SV2,Z,Y,X;                                                %001-00020200
      DI:=A; DS:=8LIT" "; SI:=A; DS:=14WDS; % BLANK "A"                         %001-00020300
      SI:=USER; SI:=SI+9; DI:=A; DS:=7CHR; DS:=LIT" "; % USERCODE (1)            00020400
      SI:=SI+1; DS:=7 CHR; DS:=LIT" "; % PASSWORD (2)                           %001-00020500
      SI:=SI+1; DS:=7 CHR; DS:=LIT" "; % CHARGEWORD (3)                         %001-00020600
      SI:=SI+1; DS:=15CHR; DS:=LIT" "; % USERNAME (4) (5)                       %001-00020700
      SI:=SI+1; DS:=7 CHR; DS:=LIT" "; % PHONEWORD (6)                         %001-00020800
      SI:=SI+4; 24(IF SB THEN DS:=LIT"1" ELSE DS:=LIT"0"; SKIP SB);             %001-00020900
```

```
      DS:=LIT" "; % TIMEWORD (7)                                          %001-00021000
      SKIP SB; SV1:=DI; DI:=LOC X; % LANGUAGEWORD (8)                     %001-00021100
      NUMOFCOMPILERS(IF SB THEN DS:=LIT"1" ELSE DS:=LIT"0"; SKIP SB);      00021200
      DI:=SV1; SI:=CMPS; SI:=SI+9; SV2:=SI; SI:=LOC X;                    %001-00021300
      NUMOFCOMPILERS(SV1:=SI;                                             %001-00021400
      IF SC="1" THEN % RESTRICTED COMPIILER                               %001-00021500
         BEGIN                                                            %001-00021600
         SI:=SV2;DS:=CHR;SI:=SI+4;DS:=2CHR;DS:=LIT" ";SI:=SI+1;SV2:=SI;   00021700
         END                                                              %001-00021800
      ELSE                                                                %001-00021900
         BEGIN                                                            %001-00022000
         SI:=SV2; SI:=SI+8; SV2:=SI;                                      %001-00022100
         END;                                                             %001-00022200
      SI:=SV1; SI:=SI+1));                                                %001-00022300
      END STREAM PROCEDURE REFORM1;                                       %001-00022400
    %*****************************************************************00022500
    BOOLEAN STREAM PROCEDURE REFORM2(VERBWORD,A,VRBS,SKP); VALUE SKP;      00022600
    %*****************************************************************00022700
      BEGIN LOCAL SV1,SV2,Z,Y,X;                                          %001-00022800
      DI:=A; DS:=8LIT" "; SI:=A; DS:=14WDS; % BLANK "A"                   %001-00022900
      SI:=VERBWORD; SKIP SB; SKIP SKP SB; DI:=LOC X;                      %001-00023000
      20(IF SB THEN DS:=LIT"1" ELSE DS:=LIT"0"; SKIP SB); % DECODE        %001-00023100
      SI:=VRBS; SI:=SI+9; SV2:=SI; SI:=LOC X; DI:=A;                      %001-00023200
      20(SV1:=SI;                                                         %001-00023300
      IF SC="1" THEN % RESTRICTED VERB                                    %001-00023400
         BEGIN                                                            %001-00023500
         SI:=SV2; DS:=5CHR; SI:=SI+3; DS:=LIT" "; SV2:=SI; TALLY:=1;       00023600
         END                                                              %001-00023700
      ELSE                                                                %001-00023800
         BEGIN                                                            %001-00023900
         SI:=SV2; SI:=SI+8; SV2:=SI;                                      %001-00024000
         END;                                                             %001-00024100
      SI:=SV1; SI:=SI+1);                                                 %001-00024200
      REFORM2:=TALLY;                                                     %001-00024300
      END STREAM PROCEDURE REFORM2;                                       %001-00024400
    %*****************************************************************00024500
    BOOLEAN PROCEDURE INPUT(A); ARRAY A[0]; % SORT INPUT ROUTINE          %001-00024600
    %*****************************************************************00024700
    IF NOT (INPUT:=(COUNT:=COUNT+1) GTR TCOUNT) THEN                      %001-00024800
      BEGIN                                                               %001-00024900
      IF (USRINDX:=USRINDX+2) GTR 28 THEN % GET ANOTHER SEGMENT           %001-00025000
         BEGIN                                                            %001-00025100
         READ(DSK,30,USER[*]);                                            %001-00025200
         USRINDX := 0;                                                    %001-00025300
         END;                                                             %001-00025400
      MOVE(2,USER[USRINDX],A[0]); % TRANSFER 2 WDS FROM SEGMENT           %001-00025500
      END                                                                 %001-00025600
    ELSE                                                                  %001-00025700
      BEGIN                                                               %001-00025800
      REWIND(DSK); USRINDX:=-2;                                           %001-00025900
      END PROCEDURE;                                                      %001-00026000
    %*****************************************************************00026100
    PROCEDURE OUTPT(B,A); VALUE B; BOOLEAN B; ARRAY A[0]; % SORT OUTPUT    00026200
    %*****************************************************************00026300
    IF B THEN % LAST RECORD                                               %001-00026400
      BEGIN                                                               %001-00026500
      WRITE(DSK,30,USER[*]); % WRITE REMAINDER OF SEGMENT                 %001-00026600
      REWIND(DSK);                                                        %001-00026700
      END                                                                 %001-00026800
    ELSE                                                                  %001-00026900
```

```
        BEGIN                                                          %001-00027000
        IF (USRINDX:=USRINDX+2) GTR 28 THEN % WRITE ANOTHER SEGMENT    %001-00027100
           BEGIN                                                       %001-00027200
           WRITE(DSK,30,USER[*]);                                      %001-00027300
           USRINDX := 0;                                               %001-00027400
           END;                                                        %001-00027500
        MOVE(2,A[0],USER[USRINDX]); % TRANSFER 2 WDS TO SEGMENT        %001-00027600
        END PROCEDURE;                                                 %001-00027700
      %*************************************************************************00027800
      BOOLEAN STREAM PROCEDURE CMP(A,B); % SORT COMPARISON             %001-00027900
      %*************************************************************************00028000
        BEGIN                                                          %001-00028100
        SI:=A; SI:=SI+1; DI:=B; DI:=DI+1;                              %001-00028200
        IF 7SC LEQ D0 THEN TALLY:=1;                                   %001-00028300
        CMP:=TALLY;                                                    %001-00028400
        END STREAM PROCEDURE CMP;                                      %001-00028500
      %*************************************************************************00028600
      BOOLEAN PROCEDURE COMP(A,B); ARRAY A,B[0]; COMP:=CMP(A,B);       %001-00028700
      %*************************************************************************00028800
      PROCEDURE HIVALUE(A); ARRAY A[0]; A[0]:=3"141414141414";              00028900
      %*************************************************************************00029000
      TCOUNT := 0; USRINDX := -2; I:=TIME(5);                          %001-00029100
      I := "0/00/00" & I[1:13:11] & I[18:24:12] & I[36:36:12];         %001-00029110
      WRITE(LINE,TABLEMAPP,I); WRITE(LINE[DBL]);                       %001-00029200
      FOR TABLELOC := 1 STEP 1 UNTIL 29 DO % GET SCRAMBLE SEGMENTS     %001-00029300
        BEGIN                                                          %001-00029400
        READ(USERS[TABLELOC],30,TABLE[*]);                             %001-00029500
MAP:    TABLEMAP(TABLE,A); % PRINT SCRAMBLE TABLE CONTENTS             %001-00029600
        WRITE(LINE,15,A[*]);                                           %001-00029700
        FOR I:=0 STEP 2 UNTIL 28 DO IF TABLE[I] NEQ 0 THEN % VALID USER   00029800
           BEGIN                                                       %001-00029900
           TCOUNT := TCOUNT + 1; % COUNT THIS USERCODE                 %001-00030000
           IF (USRINDX:=USRINDX+2) GTR 28 THEN % WRITE OUT THIS SEGMENT   00030100
              BEGIN                                                    %001-00030200
              WRITE(DSK,30,USER[*]);                                   %001-00030300
              USRINDX := 0;                                            %001-00030400
              END;                                                     %001-00030500
           MOVE(2,TABLE[I],USER[USRINDX]); % TRANSFER TO USER ARRAY    %001-00030600
           END I LOOP;                                                 %001-00030700
        IF (I:=TABLE[1].[18:15]) GTR 0 THEN % LINK TO NEXT SEGMENT     %001-00030800
           BEGIN                                                       %001-00030900
           READ(USERS[I],30,TABLE[*]); % READ LINKED SEGMENT           %001-00031000
           GO TO MAP;                                                  %001-00031100
           END; % IF LINKED SEGMENT                                    %001-00031200
        WRITE(LINE); % SPACE BETWEEN LINKED SEGMENTS                   %001-00031300
        END TABLELOC LOOP;                                             %001-00031400
      WRITE(DSK,30,USER[*]); % WRITE REMAINDER OF SEGMENT              %001-00031500
      WRITE(LINE,NUMOFUSERS,TCOUNT);                                   %001-00031600
      REWIND(DSK); COUNT := 0; USRINDX := -2; % SET UP FOR SORT        %001-00031700
      IF TCOUNT=0 THEN GO TO EXIT; % EMPTY FILE                        %001-00031800
      SORT(OUTPT,INPUT,0,HIVALUE,COMP,2,4000);                         %001-00031900
      COUNT := 0;                                                      %001-00032000
      WRITE(LINE[PAGE]); WRITE(LINE[DBL]); WRITE(LINE,USERSRECORDS);   %001-00032100
      WHILE COUNT LSS TCOUNT DO                                        %001-00032200
        BEGIN                                                          %001-00032300
        READ(DSK,30,USER[*]); % READ NEXT SEGMENT                      %001-00032400
        FOR I:=0 STEP 2 WHILE I LSS 29 AND COUNT LSS TCOUNT DO         %001-00032500
           BEGIN                                                       %001-00032600
           READ(USERS[USER[I+1],[33:15]],30,TABLE[*]); % GET USERS SGMNT   00032700
           REFORM1(TABLE,A,CMPS); WRITE(LINE,15,A[*]);                 %001-00032800
```

```
            IF REFORM2(TABLE[9],A,VRBSL[0], 0) THEN WRITE(LINE,15,A[*]);    00032900
            IF REFORM2(TABLE[9],A,VRBSL[20],20) THEN WRITE(LINE,15,A[*]);   00033000
            IF REFORM2(TABLE[9],A,VRBSL[40],40) THEN WRITE(LINE,15,A[*]);   00033100
            WRITE(LINE[DBL]);                                               %001-00033200
            COUNT := COUNT + 1;                                             %001-00033300
            END I LOOP;                                                     %001-00033400
         END COUNT LOOP;                                                    %001-00033500
      EXIT:                                                                 %001-00033600
         CLOSE(DSK); WRITE(LINE[PAGE]);                                     %001-00033700
         END PROCEDURE PRINT;                                               %001-00033800
      %*****************************************************************00033900
      STREAM PROCEDURE BLANK(A,N);   VALUE N;                               %001-00034000
      %*****************************************************************00034100
         BEGIN                                                              %001-00034200
         DI:=A;  DS:=8 LIT "    ";  SI:=A;  DS:=N WDS;                      %001-00034300
         DI:=A; DS:=LIT"0"; % AVOID FLAG BIT ERRORS                         %001-00034400
         END  STREAM BLANK;                                                 %001-00034500
      %*****************************************************************00034600
      STREAM PROCEDURE   CLEAR(A,N);   VALUE N;                             %001-00034700
      %*****************************************************************00034800
         BEGIN                                                              %001-00034900
         DI:=A;  DS:=8 LIT "0000";  SI:=A;  DS:=N WDS;                      %001-00035000
         END  CLEAR;                                                        %001-00035100
      %*****************************************************************00035200
      STREAM PROCEDURE BUILDTIMEWORD(W,CARD);                               %001-00035300
      %*****************************************************************00035400
         BEGIN                                                              %001-00035500
         DI:=W;  DS:=4 LIT "0000";  SI:=CARD;                               %001-00035600
         40(IF SC NEQ """ THEN ELSE JUMP OUT; SI:=SI+1);                    %001-00035700
         SI := SI+1; % SKIP " CHAR.                                         %001-00035800
         24(IF SC = "1" THEN DS:=SET ELSE DS:=RESET; SI:=SI+1);             %001-00035900
         END  BUILD TIME WORD;                                              %001-00036000
      %*****************************************************************00036100
      BOOLEAN STREAM PROCEDURE DOLLAR(A);                                   %001-00036200
         BEGIN                                                              %001-00036300
         SI:=A;  IF SC = "$" THEN TALLY:=1; DOLLAR:=TALLY;                  %001-00036400
         END  DOLLAR;                                                       %001-00036500
      %*****************************************************************00036600
      REAL STREAM PROCEDURE ADDRESS(A);                                     %001-00036700
         BEGIN SI:=A; ADDRESS:=SI; END;                                     %001-00036800
      %*****************************************************************00036900
      BOOLEAN STREAM PROCEDURE GETWORD(APTR,WORD);                          %001-00037000
      %*****************************************************************00037100
      % SCANS CARD IMAGE AT ADDRESS "APTR" FOR ALPHA STRING                 %001-00037200
      % RETURNS A VALUE OF "TRUE" TO GETWORD AND END OF RECORD              %001-00037300
         BEGIN LOCAL SV; LABEL SCN,EXIT;                                    %001-00037400
         SI:=APTR; SI:=SI+5; SI:=SC; % STARTING ADDRESS                     %001-00037500
         2(40(                                                              %001-00037600
         IF SC=ALPHA THEN JUMP OUT 2 TO SCN;                                %001-00037700
         IF SC=LEFTARROW THEN % END OF RECORD MARKER                        %001-00037800
            BEGIN                                                           %001-00037900
            TALLY:=1; GETWORD := TALLY; JUMP OUT 2 TO EXIT;                 %001-00038000
            END;                                                            %001-00038100
         SI:=SI+1));                                                        %001-00038200
      SCN:                                                                  %001-00038300
         DI:=WORD; DS:=LIT"0"; DS:=CHR;                                     %001-00038400
         6(IF SC=ALPHA THEN DS:=CHR ELSE DS:=LIT" ");                       %001-00038500
         63(IF SC=ALPHA THEN SI:=SI+1 ELSE JUMP OUT);                       %001-00038600
      EXIT:                                                                 %001-00038700
         SV:=SI; SI:=LOC SV; DI:=APTR; DS:=WDS;                             %001-00038800
```

```
    END STREAM PROCEDURE GETWORD;                                          %001-00038900
%**************************************************************************00039000
BOOLEAN STREAM PROCEDURE UNQUOTE(N,APTR,B); VALUE N,APTR;                  %001-00039100
%**************************************************************************00039200
% SCANS CARD IMAGE AT ADDRESS "APTR" FOR QUOTATION MARK                    %001-00039300
% RETURNS A VALUE OF "TRUE" TO UNQUOTE IF QUOTE IS NOT PRESENT             %001-00039400
% TRANSFERS UP TO "N" CHARACTERS FOLLOWING QUOTE TO ARRAY "B"              %001-00039500
    BEGIN LABEL EXIT;                                                      %001-00039600
    SI:=APTR; DI:=B; DS:=LIT"0";                                           %001-00039700
    2(40(                                                                  %001-00039800
    IF SC=LEFTARROW THEN                                                   %001-00039900
        BEGIN                                                              %001-00040000
        TALLY:=1; JUMP OUT 2 TO EXIT;                                      %001-00040100
        END;                                                               %001-00040200
    IF SC="""" THEN % PICK UP THE CHARACTERS AFTER THE QUOTE               %001-00040300
        BEGIN                                                              %001-00040400
        SI:=SI+1; % SKIP OVER THE FIRST QUOTE                              %001-00040500
        N(IF SC="""" THEN DS:=LIT" " ELSE DS:=CHR);                        %001-00040600
        JUMP OUT 2 TO EXIT;                                                %001-00040700
        END;                                                               %001-00040800
    SI:=SI+1));                                                            %001-00040900
EXIT:                                                                      %001-00041000
    UNQUOTE := TALLY;                                                      %001-00041100
    END STREAM PROCEDURE UNQUOTE;                                          %001-00041200
%**************************************************************************00041300
PROCEDURE ERR(N);  VALUE N;  REAL N;                                       %001-00041400
%**************************************************************************00041500
    BEGIN                                                                  %001-00041600
    IF N=0 THEN % BAD DOLLAR CARD                                          %001-00041700
        BEGIN                                                              %001-00041800
        OPTIONSBUILT:=USERPRESENT:=FALSE;  CLEAR(USER,29);                 %001-00041900
        END  BAD DOLLAR CARD;                                              %001-00042000
    WRITE(LINE,FM[N]);                                                     %001-00042100
    WRITE(LINE[DBL],10,A[*]);                                              %001-00042200
    END  ERROR PROCEDURE;                                                  %001-00042300
%**************************************************************************00042400
PROCEDURE WRITEUSER;                                                       %001-00042500
%**************************************************************************00042600
    BEGIN                                                                  %001-00042700
    USERWORD:=TABLE[USRINDX]:=USERCODE;                                    %001-00042800
    TABLE[USRINDX+1],[33:15]:=USERLOC;                                     %001-00042900
    WRITE(USERS[TABLELOC],30,TABLE[*]);                                    %001-00043000
    WRITE(USERS[USERLOC],30,USER [*]);                                     %001-00043100
    END  WRITE USER;                                                       %001-00043200
%**************************************************************************00043300
PROCEDURE BUILDFILE;                                                       %001-00043400
%**************************************************************************00043500
    BEGIN                                                                  %001-00043600
    FILE NEW DISK SERIAL [20:600] (2,30,300,SAVE 999);                     %001-00043700
    USER[1]:=NEXTOPEN:=30;                                                 %001-00043800
    FILL NEW WITH "USERS  ","CANDE  ";                                     %001-00043900
        DO BEGIN                                                           %001-00044000
        WRITE(NEW,30,USER[*]); CLEAR(USER,29);                             %001-00044100
        END  UNTIL USRINDX:=USRINDX+1 GTR 29;                              %001-00044200
    LOCK(NEW,SAVE);                                                        %001-00044300
    END  BUILDING NEW FILE;                                                %001-00044400
%**************************************************************************00044500
PROCEDURE SETOPTIONS;                                                      %001-00044600
%**************************************************************************00044700
    BEGIN                                                                  %001-00044800
```

```
      FOR I:=2,3,6,7,8 DO % PASSWORD,CHARGE,PHONE,TIME,COMPILERS      %001-00044900
         BEGIN                                                        %001-00045000
         MATCH(OPTIONS[I],0) ELSE USER[I]:=OPTIONS[I];                %001-00045100
         END  FOR;                                                    %001-00045200
      MATCH(OPTIONS[4], 0) ELSE MOVE(2,OPTIONS[4] ,USER[4] ); %NAME   %001-00045300
      MATCH(OPTIONS[9], 0) ELSE MOVE(1,OPTIONS[9], USER[9] ); %VRBWRD %001-00045400
      MATCH(OPTIONS[10],0) ELSE MOVE(1,OPTIONS[10],USER[10]); %VRBWRD %001-00045500
      END  SET OPTIONS;                                               %001-00045600
%*****************************************************************************00045700
STREAM PROCEDURE SETBIT(BITNUM,WD); VALUE BITNUM;                     %001-00045800
%*****************************************************************************00045900
      BEGIN LOCAL N;                                                  %001-00046000
      SI:=LOC BITNUM; SI:=SI+6; DI:=LOC N; DI:=DI+7; DS:=CHR;         %001-00046100
      DI:=WD; N(2(SKIP 32 DB)); SKIP BITNUM DB; DS:=SET;              %001-00046200
      END STREAM PROCEDURE SETBIT;                                    %001-00046300
%*****************************************************************************00046400
PROCEDURE READUSER;                                                   %001-00046500
%*****************************************************************************00046600
      BEGIN LABEL READIN,OK;                                          %001-00046700
      IF NEXTOPEN = 0 THEN % FILE NOT OPEN                            %001-00046710
         BEGIN                                                        %001-00046720
         READ(USERS[0],5,B[*]);                                       %001-00046730
         NEXTOPEN := B[1];                                            %001-00046740
         END;                                                         %001-00046750
      IF UNQUOTE(7,APTR,USERCODE) THEN                                %001-00046800
         BEGIN                                                        %001-00046900
         ERR(0); ERR(2); % BAD DOLLAR CARD, QUOTES REQUIRED           %001-00047000
         END                                                          %001-00047100
      ELSE                                                            %001-00047200
         BEGIN % GOT A USERCODE.                                      %001-00047300
         WORD := USERCODE;  USRINDX:=-2;                              %001-00047400
         TABLELOC:=((WORD.[6:21]+WORD.[27:21])MOD 29)+1; % SCRAMBLE   %001-00047500
READIN:                                                               %001-00047600
         READ(USERS[TABLELOC],30,TABLE[*]);                           %001-00047700
            DO BEGIN                                                  %001-00047800
            USRINDX:=USRINDX+2;                                       %001-00047900
            MATCH(TABLE[USRINDX],WORD) USERPRESENT:=TRUE; % ALREADY PRESENT00048000
            END  UNTIL USERPRESENT OR USRINDX GTR 26;                 %001-00048100
         IF NOT USERPRESENT THEN % NOT IN THIS BLOCK                  %001-00048200
         IF TABLE[1].[18:15] NEQ 0 THEN % LINK TO NEXT SEGMENT        %001-00048300
            BEGIN                                                     %001-00048400
            TABLELOC := TABLE[1].[18:15]; % LOCATION OF LINKED SEGMENT%001-00048500
            USRINDX := -2; GO TO READIN;                              %001-00048600
            END;                                                      %001-00048700
         IF USERPRESENT THEN USERLOC:=TABLE[USRINDX+1].[33:15] ELSE   %001-00048800
            BEGIN % MUST BE NEW USER.                                 %001-00048900
            USRINDX := -1;  I:=0;  FOUND:=FALSE;                      %001-00049500
            DO IF FOUND:=TABLE[I]=0 THEN USRINDX:=I                   %001-00049600
            UNTIL FOUND OR I:=I+2 GTR 28;                             %001-00049700
            IF FOUND THEN                                             %001-00049800
               BEGIN                                                  %001-00049900
OK;            NEXTOPEN:=(USERLOC:=NEXTOPEN)+1;                       %001-00050000
               BLANK(USERNAME,1);  BLANK(PASSWORD,0);                 %001-00050100
               USER[29]:=TABLELOC & USRINDX[18:33:15]; % BACK LINK    %001-00050200
               END                                                    %001-00050300
            ELSE                                                      %001-00050400
               BEGIN % NO ROOM IN THIS BLOCK                          %001-00050500
               TABLE[1].[18:15]:=NEXTOPEN; % LINK TO NEXT AVAILABLE SLOT  00050600
               WRITE(USERS[TABLELOC],30,TABLE[*]); % REPLACE SCRAMBLE BLOCK00050700
               CLEAR(TABLE,29); TABLE[29]:=-0&TABLELOC[18:33:15]; % BK LNK 00050800
```

```
                TABLELOC := NEXTOPEN; USRINDX := 0;                                %001-00050900
                NEXTOPEN := NEXTOPEN + 1; % INCREMENT FILE SIZE                    %001-00051000
                WRITE(USERS[TABLELOC],30,TABLE[*]); %WRITE IN CASE OF ERROR 00051100
                FOUND := TRUE; GO TO OK;                                           %001-00051200
                END;                                                               %001-00051300
             END  NEW USER;                                                        %001-00051400
          END  IF WE HAD A USER CODE;                                              %001-00051500
       IF USERPRESENT THEN READ(USERS[USERLOC],30,USER[*]);                        %001-00051600
       USERPRESENT:=(FOUND OR USERPRESENT);                                        %001-00051700
       IF USERLOC LSS 30 THEN ERR(4);                                              %001-00051800
       ALTERED := ALTERED OR USERPRESENT;                                          %001-00051900
       END  READ USER;                                                             %001-00052000
    %*************************************************************************** 00052100
    PROCEDURE REMOVEUSER;                                                          %001-00052200
    %*************************************************************************** 00052300
       BEGIN                                                                       %001-00052400
       % ROUTINE FOR REMOVING USERCODE FROM USERS/CANDE FILE                      %001-00052500
       % EXPECTS CARD IMAGE OF THE FORM: $ REMOVE "USERCODE"                      %001-00052600
       REAL USRCODE,LASTUSERINX,LASTABLELOC;                                       %001-00052700
       LABEL FOUNDIT,EXIT,READIN;                                                  %001-00052800
       FORMAT                                                                      %001-00052900
       NOTINFILE("REMOVE IGNORED, USERCODE NOT IN FILE"),                          %001-00053000
       INVALIDLOC("REMOVE IGNORED, INVALID RECORD ADDRESS"),                       %001-00053100
       INVALIDLINK("REMOVE IGNORED, FILE NOT PROPERLY LINKED");                    %001-00053200
       IF UNQUOTE(7,APTR,USRCODE) THEN                                             %001-00053300
          BEGIN                                                                    %001-00053400
          ERR(2); GO TO EXIT; % QUOTES REQUIRED AROUND PARAMETERS                  %001-00053500
          END;                                                                     %001-00053600
       TABLELOC:=((USRCODE.[6:21]+USRCODE.[27:21]) MOD 29) +1;                     %001-00053700
    READIN:                                                                        %001-00053800
       READ(USERS[TABLELOC],30,TABLE[*]); % GET INDEX SEGMENT                      %001-00053900
       FOR USRINDX:=0 STEP 2 UNTIL 28 DO                                           %001-00054000
       MATCH(TABLE[USRINDX],USRCODE) GO TO FOUNDIT;                                %001-00054100
       IF (TABLELOC:=TABLE[1].[18:15]) NEQ 0 THEN GO READIN; % LINKED              %001-00054200
       WRITE(LINE,NOTINFILE); WRITE(LINE[DBL],10,A[*]); GO EXIT;                   %001-00054300
    FOUNDIT:                                                                       %001-00054400
       USERLOC := TABLE[USRINDX+1].[33:15]; % USERS RECORD LOCATION                %001-00054500
       IF USERLOC LSS 30 THEN                                                      %001-00054600
          BEGIN                                                                    %001-00054700
          WRITE(LINE,INVALIDLOC); WRITE(LINE[DBL],10,A[*]); GO EXIT;               %001-00054800
          END;                                                                     %001-00054900
       IF NEXTOPEN=0 THEN % FIRST CARD, GET FILE LENGTH                            %001-00055000
          BEGIN                                                                    %001-00055100
          READ(USERS[0],5,B[*]); NEXTOPEN:=B[1];                                   %001-00055200
          END;                                                                     %001-00055300
       IF USERLOC = NEXTOPEN-1 THEN % THIS IS THE LAST RECORD                      %001-00055400
          BEGIN                                                                    %001-00055500
          TABLE[USRINDX] := TABLE[USRINDX+1] := 0;                                 %001-00055600
          WRITE(USERS[TABLELOC],30,TABLE[*]);                                      %001-00055700
          NEXTOPEN := NEXTOPEN -1; % DECREMENT RECORD COUNT                        %001-00055800
          GO TO EXIT;                                                              %001-00055900
          END;                                                                     %001-00056000
       READ(USERS[NEXTOPEN-1],30,USER[*]); % GET LAST RECORD IN FILE               %001-00056100
       IF BOOLEAN(USER[29].[1:1]) THEN % LAST RECORD IS A SCRAMBLE BLOCK            00056200
          BEGIN                                                                    %001-00056300
          IF (I:=USER[29].[18:15]) = 0 THEN % INVALID LINK                         %001-00056400
             BEGIN                                                                 %001-00056500
             WRITE(LINE,INVALIDLINK); WRITE(LINE[DBL],10,A[*]); GO EXIT;            00056600
             END;                                                                  %001-00056700
          TABLE[USRINDX]:=0; TABLE[USRINDX+1].[33:15]:=0; % REMOVE USERCODE       00056800
```

```
      WRITE(USERS[TABLELOC],30,TABLE[*]); % REPLACE SCRAMBLE BLOCK %001-00056900
      IF TABLELOC=NEXTOPEN-1 THEN % USER IS IN LAST BLOCK         %001-00057000
         BEGIN                                                   %001-00057100
         USER[USRINDX]:=0; USER[USRINDX+1].[33:15]:=0;           %001-00057200
         END;                                                    %001-00057300
      K := -2;                                                   %001-00057400
      DO K:=K+2 UNTIL USER[K] NEQ 0 OR K GTR 28;                 %001-00057500
      IF K GTR 28 THEN % EMPTY BLOCK                             %001-00057600
         BEGIN                                                   %001-00057700
         USERLOC := 0; NEXTOPEN := NEXTOPEN -1;                  %001-00057800
         END                                                     %001-00057900
      ELSE                                                       %001-00058000
      WRITE(USERS[USERLOC],30,USER[*]); % MOVE LAST RECORD UP    %001-00058100
      NEXTOPEN := NEXTOPEN - 1; % DECREMENT FILE SIZE            %001-00058200
      READ(USERS[I],30,TABLE[*]); % GET PREVIOUS SCRAMBLE BLOCK  %001-00058300
      TABLE[1].[18:15]:=USERLOC; % NEW LOCATION FOR LINKED BLOCK  %001-00058400
      WRITE(USERS[I],30,TABLE[*]); % REPLACE SCRAMBLE BLOCK      %001-00058500
      END % IF LAST RECORD IS A SCRAMBLE BLOCK                   %001-00058600
   ELSE                                                          %001-00058700
      BEGIN % NOT A SCRAMBLE BLOCK                               %001-00058800
      LASTUSERINX := USER[29].[18:15]; % LOCATION IN SCRAMBLE BLOCK%001-00058900
      LASTABLELOC := USER[29].[33:15]; % SCRAMBLE BLOCK LOCATION  %001-00059000
      IF LASTABLELOC = 0 OR LASTUSERINX GTR 28 THEN % INVALID LINK %001-00059100
         BEGIN                                                   %001-00059200
         WRITE(LINE,INVALIDLINK); WRITE(LINE[DBL],30,A[*]); GO EXIT;    00059300
         END;                                                    %001-00059400
      TABLE[USRINDX]:=0; TABLE[USRINDX+1].[33:15]:=0; % REMOVE USER%001-00059500
      IF LASTABLELOC NEQ TABLELOC THEN % DIFFERENT SCRAMBLE BLOCK %001-00059600
         BEGIN                                                   %001-00059700
         WRITE(USERS[TABLELOC],30,TABLE[*]); % WRITE CURRENT BLOCK %001-00059800
         READ(USERS[LASTABLELOC],30,TABLE[*]); % BLOCK FOR LST RECRD    00059900
         END;                                                    %001-00060000
      TABLE[LASTUSERINX+1].[33:15]:=USERLOC; % NEW LOC. FOR LST. REC.    00060100
      WRITE(USERS[LASTABLELOC],30,TABLE[*]); % REPLACE SCRAMBLE BLOCK    00060200
      WRITE(USERS[USERLOC],30,USER[*]); % MOVE UP LAST RECORD      %001-00060300
      NEXTOPEN := NEXTOPEN -1; % DECREMENT FILE SIZE              %001-00060400
      END; % IF NOT SCRAMBLE BLOCK                               %001-00060500
   ALTERED := TRUE;                                              %001-00060600
EXIT:                                                            %001-00060700
   END PROCEDURE REMOVEUSER;                                     %001-00060800
%************************************************************************00060900
PROCEDURE LINKUP;                                                %001-00061000
%************************************************************************00061100
   BEGIN LABEL LINK;                                             %001-00061200
   %ROUTINE TO ESTABLISH LINKS FROM USERS/CANDE RECORDS TO INDEX TABLE  00061300
   %ROUTINE EXPECTS TO FIND CARD IMAGE OF THE FORM: $ LINK        %001-00061400
   TABLELOC := 0;                                                %001-00061500
      DO BEGIN                                                   %001-00061600
      READ(USERS[TABLELOC:=TABLELOC+1],30,TABLE[*]);             %001-00061700
LINK: USRINDX := -2;                                             %001-00061800
      DO BEGIN                                                   %001-00061900
         IF TABLE[USRINDX:=USRINDX+2] NEQ 0 THEN                 %001-00062000
         IF USERLOC := TABLE[USRINDX+1].[33:15] GEQ 30 THEN      %001-00062100
            BEGIN                                                %001-00062200
            READ(USERS[USERLOC],30,USER[*]);                     %001-00062300
            USER[29] := TABLELOC & USRINDX[18:33:15];            %001-00062400
            WRITE(USERS[USERLOC],30,USER[*]);                    %001-00062500
            END;                                                 %001-00062600
      END UNTIL USRINDX = 28;                                    %001-00062700
   IF (I:=TABLE[1].[18:15]) NEQ 0 THEN % LINKED SCRAMBLE BLOCK   %001-00062800
```

```
            BEGIN                                                                %001-00062900
            READ(USERS[I],30,TABLE[*]);                                          %001-00063000
            GO LINK;                                                             %001-00063100
            END;                                                                 %001-00063200
         END UNTIL TABLELOC = 29;                                                %001-00063300
      END PROCEDURE LINKUP;                                                      %001-00063400
%************************************************************************      00063500
FILL CMPS[*] WITH 0,                                                           %001-00063600
"BASIC  ","ALGOL  ","COBOL  ","FORTRAN","TSPOL  ","XALGOL ",                   %001-00063700
"*******","*******","*******","*******","COBOL68";                             %001-00063710
FILL VRBS[*] WITH 0,                                                           %001-00063800
"LIST   ","RUN    ","SAVE   ","SEQ    ","EXECUTE","DO     ",                   %001-00063900
"MAKE   ","LOAD   ","COMPILE","REMOVE ","CHANGE ","RENAME ",                   %001-00064000
"FILES  ","WHATS  ","CREATE ","HELLO  ","TYPE   ","BYE    ",                   %001-00064100
"COPY   ","MERGE  ","RESEQ  ","DELETE ","APPEND ","ADD    ",                   %001-00064200
"TO     ","SS     ","GUARD  ","LOCK   ","UNLOCK ","TAPE   ",                   %001-00064300
"PRINT  ","PUNCH  ","CHARGE ","SCHEDUL","STATUS ","STOP   ",                   %001-00064400
"TIME   ","UPDATE ","REPLACE","FIND   ","CC     ","RMERGE ",                   %001-00064500
"DISPLAY","SET    ","RESET  ","MONITOR","SSFILE ","EQUATE ",                      00064600
"PUBLIC ";                                                                        00064610
A[0]:=0; A0:=ADDRESS(A[0]); FIRSTCARD:=TRUE; CLEAR(USER,29);                   %001-00064700
NEXT:                                                                          %001-00064800
READ(CARD,10,A[*])[EOF]; A[10]:=REAL(NOT FALSE); % END RECORD MARK             %001-00064900
APTR := A0;                                                                    %001-00065000
IF GETWORD(APTR,WORD) THEN GO TO NEXT; % BLANK CARD                            %001-00065100
IF DOLLARCARD:=DOLLAR(A) THEN                                                  %001-00065200
   BEGIN % DOLLAR CARD READ.                                                   %001-00065300
   IF USERPRESENT THEN WRITEUSER                                               %001-00065400
   ELSE                                                                        %001-00065500
   IF OPTIONSBUILT THEN MOVE(30,USER,OPTIONS);                                 %001-00065600
   OPTIONSBUILT:=USERPRESENT:=FALSE;  CLEAR(USER,29);                          %001-00065700
   USERCODE := "       ";                                                      %001-00065800
   MATCH(WORD,"USER   ")                                                       %001-00065900
      BEGIN                                                                    %001-00066000
      READUSER; SETOPTIONS;                                                    %001-00066100
      END                                                                      %001-00066200
   ELSE                                                                        %001-00066300
   MATCH(WORD,"OPTIONS") OPTIONSBUILT := TRUE                                  %001-00066400
   ELSE                                                                        %001-00066500
   MATCH(WORD,"NEW    ") IF FIRSTCARD THEN BUILDFILE ELSE ERR(1)               %001-00066600
   ELSE                                                                        %001-00066700
   MATCH(WORD,"PRINT  ") PRINT                                                 %001-00066800
   ELSE                                                                        %001-00066900
   MATCH(WORD,"REMOVE ") REMOVEUSER                                            %001-00067000
   ELSE                                                                        %001-00067100
   MATCH(WORD,"LINK   ") LINKUP                                                %001-00067200
   ELSE                                                                        %001-00067300
   ERR(0); % BAD $ CARD.                                                       %001-00067400
   FIRSTCARD := FALSE;                                                         %001-00067500
   END  DOLLAR CARD                                                            %001-00067600
ELSE                                                                           %001-00067700
   BEGIN % NOT A DOLLAR CARD.                                                  %001-00067800
   MATCH(WORD,"USE    ") % OPTIONAL WORD                                       %001-00067900
      BEGIN                                                                    %001-00068000
      IF GETWORD(APTR,WORD) THEN GO TO NEXT;                                   %001-00068100
      END;                                                                     %001-00068200
   MATCH(WORD,"PASSWOR") IF UNQUOTE(7,APTR,PASSWORD) THEN ERR(2) % "    00068300
   ELSE ELSE                                                                   %001-00068400
   MATCH(WORD,"NAME   ") IF UNQUOTE(15,APTR,USERNAME) THEN ERR(2) % "    00068500
   ELSE ELSE                                                                   %001-00068600
```

```
      MATCH(WORD,"TIME    ") BUILDTIMEWORD(TIMEWORD,A)                          %001-00068700
      ELSE                                                                     %001-00068800
      MATCH(WORD,"CHARGE ") IF UNQUOTE(/,APTR,CHARGEWORD) THEN ERR(2) % "  00068900
      ELSE ELSE                                                                %001-00069000
      MATCH(WORD,"REQUEST") CHARGEWORD := 0                                    %001-00069100
      ELSE                                                                     %001-00069200
      MATCH(WORD,"NO      ") CHARGEWORD := "-1      "                          %001-00069300
      ELSE                                                                     %001-00069400
      MATCH(WORD,"LANGUAG")                                                    %001-00069500
         BEGIN                                                                 %001-00069600
   COMPILERS:                                                                  %001-00069700
         IF GETWORD(APTR,WORD) THEN GO TO NEXT; % END OF RECORD                %001-00069800
         MATCH(WORD,"NONE    ") COMPILERWORD := 0 ELSE                         %001-00069900
            BEGIN                                                              %001-00070000
            FOR I:=1 STEP 1 UNTIL NUMOFCOMPILERS DO                            %001-00070100
            MATCH(WORD,CMPS[I])                                                %001-00070200
               BEGIN                                                           %001-00070300
               SETBIT(I,COMPILERWORD);                                         %001-00070400
               GO TO COMPILERS;                                                %001-00070500
               END;                                                            %001-00070600
            ERR(4); GO TO NEXT; % NOT COMPILER NAME                            %001-00070700
            END; % IF A COMPILER NAME                                          %001-00070800
         END % IF "LANGUAGE"                                                   %001-00070900
      ELSE                                                                     %001-00071000
      MATCH(WORD,"PHONE  ") IF UNQUOTE(/,APTR,PHONEWORD) THEN ERR(2) % "   00071100
      ELSE ELSE                                                                %001-00071200
      MATCH(WORD,"VERBS  ")                                                    %001-00071300
         BEGIN                                                                 %001-00071400
   VERBS:                                                                      %001-00071500
         IF GETWORD(APTR,WORD) THEN GO TO NEXT;                                %001-00071600
         MATCH(WORD,"NONE    ") CLEAR(VERBWORD,1) ELSE                         %001-00071700
            BEGIN                                                              %001-00071800
            FOR I:=1 STEP 1 UNTIL NUMOFVERBS DO                                %001-00071900
            MATCH(WORD,VRBS[I])                                                %001-00072000
               BEGIN                                                           %001-00072100
               SETBIT(I,VERBWORD);                                             %001-00072200
               GO TO VERBS;                                                    %001-00072300
               END;                                                            %001-00072400
            ERR(5); % NOT VERB NAME GO TO NEXT;                                %001-00072500
            END; % IF A VERB NAME                                              %001-00072600
         END % IF VERBS                                                        %001-00072700
      ELSE                                                                     %001-00072800
      ERR(3); % UNKNOWN OPTION                                                 %001-00072900
      END  NOT DOLLAR CARD;                                                    %001-00073000
   GO TO NEXT;                                                                 %001-00073100
   EOF:                                                                        %001-00073200
   IF USERPRESENT THEN WRITEUSER; % WRITE REMAINING SEGMENTS                   %001-00073300
   IF ALTERED THEN % UPDATE RECORD COUNT AND ALTERATION DATE                   %001-00073400
      BEGIN                                                                    %001-00073500
      READ(USERS[0],29,USER[*]); % GET SEGMENT ZERO                            %001-00073600
      USER[1] := NEXTOPEN;                                                     %001-00073700
      USER[2]:=TIME(5);                                                        %001-00073800
      WRITE(USERS[0],30,USER[*]);                                              %001-00073900
      CLOSE(USERS,SAVE);                                                       %001-00074000
      END; % IF ALTERED                                                        %001-00074100
   END  USER.                                                                  %001-00074200
   END;END.          LAST CARD ON OCRDING TAPE                                    99999999
```

LABEL 000000000PRINTER0017510OCC EX OBJECT/READ;FILE SOURCEFILE=SYMBOL/USER;END←

OBJECT /READ