LABEL 000000000PRINTER00175100CC EX OBJECT/READ;FILE SOURCEFILE=SYMBOL/FIND;END←

OBJECT /READ

SYMBOL/FIND

```
BEGIN   REAL COMMON;                                                    00010000
%************************** FIND/CANDE 3-70 ****************************00010100
  COMMENT: * TITLE: B5500/B5700 MARK XIV SYSTEM RELEASE            *    00010110
           * FILE ID: SYMBOL/FIND  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 48232        *    00010116
           *                                                       *    00010117
           * COPYRIGHT (C) 1971, 1972 BURROUGHS CORPORATION        *    00010118
           *  AA320206   AA332366   AA386657                       *;   00010119
DEFINE LEFTARROW="←"#, MAXRANGES = 25#;                                 00010200
ARRAY GRPTKNS,FILINX[0:63],FNDINX[0:170],LBOUND,UBOUND[0:MAXRANGES];    00010300
SAVE ARRAY ESP[0:30],  OUTBUF[0:11],  WORK[0:30];                       00010400
BOOLEAN FOUND, BRAAK, TEXT, SEQCHK, INFIL, OUTFIL, SITE, HAVEFILE, SFLG;00010500
FILE IN  OLD 12 (2,0,0);                                                00010600
FILE OUT LPA[illegible](2,15);                                          00010700
FILE OUT DSK2 DISK SERIAL [20:600] (2,10,300,SAVE 30);                  00010800
INTEGER OLDSIZE, TOKENCOUNT, TOSCAN;                                    00010900
REAL  ADJ,  COLS,  COLSLEFT,  COUNT,  ENDWORD,  FLG,  GRPMRK,  GRPCOUNT,00011000
   HRANGE,  I,  EOR,  LINSIZ,  LREC,  LRANGE,  MAX,  NAM1,  NAM2,  NAM3,00011100
   ESPPTR,  NF,  NRANGES,  NR,  NUM1,  NUM2,  PREVNF,  PERLINE,  REC,   00011200
   RC,  SCANNED,  SEQ,  STRTWORD,  TKNS,  TOKENLENGTH,  TOKENPTR,       00011300
   TOTAL,  USER,  WRKADRS,  WORKPTR;                                    00011400
LABEL ENDILOOP,  EXIT,  FINDLIT,  HAVEOLD,  NEXTRANGE,                  00011500
   RANDOMRD,  SERIALRD,  FINISH,  TESTREAD,  TOKENLOOP;                 00011600
%***********************************************************************00011700
PROCEDURE TWXOUT(A,N,T); VALUE N,T; REAL A,N,T;                         00011800
   BEGIN COMMUNICATE(-11); BRAAK:=BOOLEAN(T); END;                      00011900
%***********************************************************************00012000
REAL STREAM PROCEDURE ADDRESS(A);                                       00012100
   BEGIN SI:=A; ADDRESS:=SI; END;                                       00012200
%***********************************************************************00012300
REAL STREAM PROCEDURE DECCONV(OCTV); VALUE OCTV;                        00012400
   BEGIN SI:=LOC OCTV; DI:=LOC DECCONV; DS:=8DEC; END;                  00012500
%***********************************************************************00012600
STREAM PROCEDURE SHIFTSEQ(WORK);                                        00012700
%***********************************************************************00012800
   BEGIN                                                                00012900
   DI:=WORK; DI:=DI+8; SI:=WORK; DS:=WDS; DI:=WORK; DS:=LIT LEFTARROW;  00013000
   END STREAM PROCEDURE SHIFTSEQ;                                       00013100
%***********************************************************************00013200
STREAM PROCEDURE BACKSHIFT(WORK);                                       00013300
   BEGIN SI:=WORK; SI:=SI+8; DI:=WORK; DS:=WDS; END;                    00013400
%***********************************************************************00013500
STREAM PROCEDURE MOVEWORDS(N,A,B);  VALUE N;                            00013600
   BEGIN SI:=A; DI:=B; DS:=N WDS; END;                                  00013700
%***********************************************************************00013800
REAL STREAM PROCEDURE OCTCONV(DECV);                                    00013900
   BEGIN SI:=DECV; DI:=LOC OCTCONV; DS:= 8 OCT; END;                    00014000
%***********************************************************************00014100
PROCEDURE DISKWAIT(IO,AREA,WDS,ADDR);                                   00014200
VALUE IO,WDS,ADDR; REAL IO,WDS,ADDR; ARRAY AREA[0]; COMMUNICATE(-8);    00014300
%***********************************************************************00014400
STREAM PROCEDURE MOVETYPE(F1,F2);                                       00014500
%***********************************************************************00014600
   BEGIN LOCAL SV;                                                      00014700
   SI:=F2;  3(SI:=SI-8); DI:=LOC F2; DS:=WDS;                           00014800
   SI:=F2; 14(SI:=SI+8); DI:=LOC F2; DS:=WDS;                           00014900
```

```
    SI:=F2;  4(SI:=SI+8);  SI:=SI+6; SV:=SI;                                  00015000
    SI:=F1;  3(SI:=SI-8); DI:=LOC F1; DS:=WDS;                                00015100
    SI:=F1; 14(SI:=SI+8); DI:=LOC F1; DS:=WDS;                                00015200
    SI:=F1;  4(SI:=SI+8);  SI:=SI+6; DI:=SV; DS:=CHR;                         00015300
    END STREAM PROCEDURE MOVETYPE;                                            00015400
%*************************************************************************** 00015500
STREAM PROCEDURE NUMFOUND(D,NM); VALUE NM;                                    00015600
%*************************************************************************** 00015700
    BEGIN                                                                     00015800
    DI:=D; DS:=26LIT"NUMBER OF STRINGS FOUND = ";                             00015900
    SI:=LOC NM; DS:=8 DEC; DI:=DI-8; DS:=7 FILL;                              00016000
    END STREAM PROCEDURE NUMFOUND;                                            00016100
%*************************************************************************** 00016200
REAL STREAM PROCEDURE HDR(F,N); VALUE N;                                      00016300
%*************************************************************************** 00016400
    BEGIN                                                                     00016500
    SI:=F;  3(SI:=SI-8); DI:=LOC F; DS:=WDS;                                  00016600
    SI:=F; 14(SI:=SI+8); DI:=LOC F; DS:=WDS;                                  00016700
    SI:=F;  N(SI:=SI+8); DI:=LOC HDR; DS:=WDS;                                00016800
    END STREAM PROCEDURE HDR;                                                 00016900
%*************************************************************************** 00017000
STREAM PROCEDURE  ERRMSG (BUF, ERRMSG);   VALUE ERRMSG;                       00017100
%*************************************************************************** 00017200
    BEGIN                                                                     00017300
    DI:=BUF; DS:=4LIT"ERR:"; SI:=LOC ERRMSG; SI:=SI+1; DS:=7CHR;              00017400
    DS:=2LIT" ";                                                              00017500
    END ERRMSG;                                                               00017600
%*************************************************************************** 00017700
STREAM PROCEDURE REKNUMBER(BUF,LREC); VALUE LREC;                             00017800
%*************************************************************************** 00017900
    BEGIN                                                                     00018000
    SI:=LOC LREC; DI:=BUF; DS:=8DEC; DI:=BUF; DS:=7FILL;                      00018100
    SI:=BUF; DI:=BUF; 8(IF SC=" " THEN SI:=SI+1 ELSE DS:=CHR);                00018200
    DS:=8LIT" ";                                                              00018300
    END STREAM PROCEDURE REKNUMBER;                                           00018400
%*************************************************************************** 00018500
REAL PROCEDURE CHEKFILE(NAM1,NAM2,ARAY); VALUE NAM1,NAM2;                     00018600
%*************************************************************************** 00018700
REAL NAM1,NAM2; ARRAY ARAY[0];                                                00018800
    BEGIN FILE IN DSK 12 (2,10,30); REAL V;                                   00018900
    STREAM PROCEDURE MESSAGE(ARAY,NAM1,NAM2,N); VALUE N;                      00019000
        BEGIN LABEL LB1,LB2,LB3,FIN;                                          00019100
        DI:=ARAY; DS:=5LIT"FILE "; SI:=NAM1;                                  00019200
        SI:=SI+1; 7(IF SC=ALPHA THEN DS:=CHR ELSE SI:=SI+1); DS:=LIT"/";      00019300
        SI:=SI+1; 7(IF SC=ALPHA THEN DS:=CHR ELSE JUMP OUT); DS:=LIT" ";      00019400
        CI:=CI+N; GO LB1; GO LB2; GO LB3;                                     00019500
LB1:    DS:=14LIT"IS NOT ON DISK"; GO TO FIN;                                 00019600
LB2:    DS:= 9LIT"IS LOCKED"; GO TO FIN;                                      00019700
LB3:    DS:=26LIT"IS ALREADY IN YOUR LIBRARY";                                00019800
FIN:    DS:=LIT LEFTARROW;                                                    00019900
        END STREAM PROCEDURE MESSAGE;                                         00020000
    FILL DSK WITH NAM1,NAM2; SEARCH(DSK,ARAY[*]);                             00020100
    CHEKFILE:=V:=ARAY[0]; IF V:=V+1 GTR 2 THEN V:=2;                          00020200
    MESSAGE(ARAY,NAM1,NAM2,V);                                                00020300
    END PROCEDURE CHEKFILE;                                                   00020400
%*************************************************************************** 00020500
REAL STREAM PROCEDURE ATOKEN(IPTR, IPTR, EOR); VALUE EOR;                     00020600
%*************************************************************************** 00020700
    BEGIN LOCAL STRT,FIN; LABEL LOOP,NEXTADRS,EXIT;                           00020800
    SI:=IPTR; SI:=SI+5; SI:=SC;                                               00020900
```

```
LOOP: IF SC=" " THEN BEGIN SI:=SI+1; GO TO LOOP; END;                          00021000
   STRT:=SI; % SAVE STARTING ADDRESS                                           00021100
   IF SC GEQ "0" THEN IF SC LEQ "9" THEN;                                      00021200
   IF TOGGLE THEN % NUMERIC TOKEN                                              00021300
      BEGIN                                                                    00021400
      SI:=SI+1; TALLY := 1;                                                    00021500
      62(IF SC GEQ "0" THEN IF SC LEQ "9" THEN;                                00021600
      IF TOGGLE THEN BEGIN SI:=SI+1; TALLY:=TALLY+1; END ELSE JUMP OUT);       00021700
      GO TO NEXTADRS;                                                          00021800
      END; % IF NUMERIC TOKEN                                                  00021900
   IF SC = ALPHA THEN IF SC LEQ "9" THEN % ALPHANUMERIC TOKEN                  00022000
      BEGIN                                                                    00022100
      SI:=SI+1; TALLY:=1;                                                      00022200
      62(IF SC = ALPHA THEN IF SC LEQ "9" THEN;                                00022300
      IF TOGGLE THEN BEGIN SI:=SI+1; TALLY:=TALLY+1; END ELSE JUMP OUT);       00022400
      GO TO NEXTADRS;                                                          00022500
      END; % IF ALPHANUMERIC TOKEN                                             00022600
   IF SC NEQ LEFTARROW THEN % SPECIAL CHARACTER                                00022700
      BEGIN TALLY:=1; SI:=SI+1; GO TO NEXTADRS; END;                           00022800
   SI:=LOC STRT; DI:=LOC EOR;                                                  00022900
   IF 8 SC=DC THEN GO TO EXIT; % END OF RECORD                                 00023000
   SI:=STRT; SI:=SI+1; TALLY:=1;                                               00023100
NEXTADRS:                                                                      00023200
   FIN:=SI; SI:=LOC FIN; DI:=IPTR; DS:=WDS; % ADDRESS OF NEXT CHR.             00023300
   SI:=LOC STRT; DI:=TPTR; DS:=WDS; % ADDRESS OF TOKEN                         00023400
EXIT:                                                                          00023500
   ATOKEN := TALLY; % TOKEN SIZE. ( ZERO IF EOR )                              00023600
   END STREAM PROCEDURE ATOKEN;                                                00023700
%**************************************************************************00023800
STREAM PROCEDURE OUTFORMAT(BUF,WORK,SEQ);                                      00023900
%**************************************************************************00024000
   BEGIN                                                                       00024100
   SI:=SEQ; DI:=BUF; DS:=8CHR; DI:=BUF; DS:=7FILL;                             00024200
   SI:=BUF;DI:=BUF; 8(IF SC=" " THEN SI:=SI+1 ELSE DS:=CHR); DS:=LIT" ";       00024300
   SI:=WORK; 2(DS:=36 CHR); DS:=9LIT" ";                                       00024400
   END STREAM PROCEDURE OUTFORMAT;                                             00024500
%**************************************************************************00024600
REAL STREAM PROCEDURE LITSCAN(TS,WP,FINX,F); VALUE TS;                         00024700
%**************************************************************************00024800
   BEGIN LOCAL FLOK,OS;                                                        00024900
   DI:=F; DS:=8LIT"0"; % RESET F TO FALSE                                      00025000
   SI:=FINX; SI:=SI+4; DI:=LOC OS; DI:=DI+7; DS:=CHR; % OLD SIZE               00025100
   SI:=SC; FLOK:=SI; % ADDRESS OF OLD STRING                                   00025200
   SI:=WP; SI:=SI+5; SI:=SC; % ADDRESS OF WORK AREA                            00025300
   TS(LITSCAN:=SI; DI:=FLOK;                                                   00025400
   IF OS SC = DC THEN JUMP OUT;                                                00025500
   SI:=LITSCAN; SI:=SI+1; TALLY:=TALLY+1);                                     00025600
   IF TOGGLE THEN % FOUND A MATCH                                              00025700
      BEGIN DI:=F; DI:=DI+7; DS:=LIT"1"; END;                                  00025800
   LITSCAN:=SI; SI:=LOC LITSCAN; DI:=WP; DS:=WDS;                              00025900
   LITSCAN:=TALLY;                                                             00026000
   END STREAM PROCEDURE LITSCAN;                                               00026100
%**************************************************************************00026200
REAL STREAM PROCEDURE SETNUM(BUF,SEQ,LREC,SFLG,STAR); VALUE SFLG,STAR;         00026300
%**************************************************************************00026400
   BEGIN LOCAL SV; LABEL LBL;                                                  00026500
   DI:=BUF; DS:=LIT" "; TALLY:=1;                                              00026600
   STAR(DS:=LIT"*"; TALLY:=2);                                                 00026700
   SV:=DI;                                                                     00026800
   SFLG(SI:=SEQ; DI:=LOC STAR; DS:=8CHR; JUMP OUT TO LBL);                     00026900
```

```
      SI:=LREC; DI:=LOC STAR; DS:=8DEC;                                    00027000
   LBL:                                                                    00027100
      DI:=LOC STAR; DS:=7FILL; SI:=LOC STAR;                               00027200
      DI:=SV;                                                              00027300
      8(IF SC=" " THEN SI:=SI+1 ELSE                                       00027400
         BEGIN DS:=CHR; TALLY:=TALLY+1; END);                              00027500
      SETNUM := TALLY;                                                     00027600
      END STREAM PROCEDURE SETNUM;                                         00027700
   %*******************************************************************   00027800
   BOOLEAN STREAM PROCEDURE TOKENSMATCH(N,NDX1,NDX2);   VALUE N;           00027900
   %*******************************************************************   00028000
      BEGIN                                                                00028100
      LABEL   EXIT;                                                        00028200
      LOCAL   S,D,R;                                                       00028300
      SI:=NDX1; SI:=SI+4; DI:=NDX2; DI:=DI+4;                              00028400
      N(S:=SI;  D:=DI;                                                     00028500
      IF SC=DC THEN    % TOKEN LENGTHS ARE EQUAL.                         00028600
         BEGIN                                                             00028700
         SI:=SI-1; DI:=LOC R; DI:=DI+7; DS:=CHR;                           00028800
         SI:=SC; DI:=D; DI:=DI+1; DI:=DC;                                  00028900
         IF R SC NEQ DC THEN JUMP OUT TO EXIT;   % TOKENS NOT EQUAL.       00029000
         END  ELSE  JUMP OUT TO EXIT;                                      00029100
      SI:=S;  SI:=SI+8;  DI:=D;  DI:=DI+8;);                               00029200
      TALLY:=1;                                                            00029300
   EXIT:                                                                   00029400
      TOKENSMATCH:=TALLY;                                                  00029500
      END OF TOKENSMATCH;                                                  00029600
   %*******************************************************************   00029700
   PROCEDURE PRINT;                                                        00029800
   %*******************************************************************   00029900
      BEGIN                                                                00030000
      FORMAT FMT(X81,I8);                                                  00030100
      IF OUTFIL THEN % MOVE RECORDS TO OUTPUT FILE                         00030200
         BEGIN                                                             00030300
         IF SFLG THEN BACKSHIFT(WORK[9]); % MOVE SEQUENCE NUMBER BACK      00030400
         WRITE(DSK2,10,WORK[*]);                                           00030500
         END                                                               00030600
      ELSE IF SITE THEN % LIST RECORDS ON LINE PRINTER                     00030700
         BEGIN                                                             00030800
         IF SFLG THEN BACKSHIFT(WORK[9]) ELSE WRITE(LPA[NO],FMT,LREC);     00030900
         WRITE(LPA[NO],10,WORK[*]);                                        00031000
         END                                                               00031100
      ELSE IF NOT BRAAK THEN % OUTPUT TO REMOTE                            00031200
         BEGIN                                                             00031300
         IF TEXT THEN                                                      00031400
            BEGIN                                                          00031500
            IF NOT SFLG THEN                                               00031600
               BEGIN                                                       00031700
               REKNUMBER(OUTBUF,LREC); TWXOUT(OUTBUF[0],8,1);              00031800
               TWXOUT(WORK[0],80,1);                                       00031900
               END                                                         00032000
            ELSE                                                           00032100
               BEGIN                                                       00032200
               OUTFORMAT(OUTBUF,WORK[0],WORK[10]);                         00032300
               TWXOUT(OUTBUF[0],81,1);                                     00032400
               END;                                                        00032500
            END % IF TEXT                                                  00032600
         ELSE                                                              00032700
            BEGIN                                                          00032800
            COUNT:=SETNUM(OUTBUF,WORK[10],LREC,SFLG,(PERLINE GTR 1));      00032900
```

```
            TWXOUT(OUTBUF[0],COUNT,3"100000");                                  00033000
            IF LINSIZ := LINSIZ + COUNT GTR 60 THEN                            00033100
               BEGIN                                                           00033200
               TWXOUT(OUTBUF[0],0,1); LINSIZ := 0;                             00033300
               END;                                                            00033400
            END; % IF NOT TEXT;                                                00033500
         END; % IF NOT BRAAK;                                                  00033600
      END PROCEDURE PRINT;                                                     00033700
%*************************************************************************00033800
REAL STREAM PROCEDURE TSIZE(WORK);                                             00033900
%*************************************************************************00034000
   BEGIN                                                                       00034100
   SI:=WORK; SI:=SI+4; DI:=LOC TSIZE; DI:=DI+4; DS:=4CHR;                      00034200
   END STREAM PROCEDURE TSIZE;                                                 00034300
%*************************************************************************00034400
REAL STREAM PROCEDURE UNPAK(ADRS,FINX,NF,TKN,NUM1,NUM2,FLG);                   00034500
%*************************************************************************00034600
COMMENT ROUTINE TO TRANSLATE CODED STRINGS IN INPUT SAVE ARRAY                 00034700
INTO USEABLE FORM FOR "FIND" PROGRAM.                                          00034800
FORMAT OF INPUT ARRAY IS:                                                      00034900
WD[0],[02:01] = 1 IF INPUT FILE NAME SPECIFIED                                 00035000
WD[0],[03:01] = 1 IF OUTPUTFILE NAME SPECIFIED                                 00035100
WD[0],[04:01] = 1 IF "SITE" ( LINE PRINTER ) SPECIFIED                         00035200
WD[0],[17:01] = 1 IF "PRINT TEXT" SPECIFIED                                    00035300
WD[0],[24:24] = TOTAL SIZE ( IN CHARACTERS ) OF INPUT ARRAY (OCTAL)            00035400
FORMAT OF INPUT ARRAY, STARTING AT WORD 1:                                     00035500
EACH "STRING" IS PRECEEDED BY A FOUR CHARACTER CODE -                          00035600
CHARACTER 1 = 2 IF A STRING                                                    00035700
CHARACTER 1 = 3 IF A SINGLE SEQUENCE NUMBER                                    00035800
CHARACTER 1 = 4 IF A LOWER BOUND FOR A SEQUENCE RANGE                          00035900
CHARACTER 1 = 7 IF A FILE NAME                                                 00036000
CHARACTER 2 = 1 IF "FIRST" IS SPECIFIED                                        00036100
CHARACTER 3 = 1 IF "LITERAL" IS SPECIFIED                                      00036200
CHARACTER 4 = SIZE ( IN CHARACTERS ) OF THE STRING (OCTAL)                     00036300
CHARACTER 5 = FIRST CHARACTER OF THE STRING                                    00036400
FORMAT OF OUTPUT ARRAY WORDS:                                                  00036500
WD.[17.01] = 1 IF "FIRST" SPECIFIED                                            00036600
WD.[23:01] = 1 IF "LITERAL" SPECIFIED                                          00036700
WD.[24:06] = SIZE OF STRING OR TOKEN                                           00036800
WD.[30:18] = ADDRESS OF STRING OR TOKEN                                        00036900
THE ROUTINE -                                                                  00037000
BREAKS THE "EXISTING" STRINGS INTO TOKENS ( UNLESS "LITERAL" IS                00037100
SPECIFIED ) AND PLACES THE SIZE AND LOCATION (ABSOLUTE ADDRESS                 00037200
OF THE TOKEN IN THE INPUT ARRAY) INTO ARRAY "FINX".                            00037300
THE NUMBER OF TOKENS INTO WHICH THE STRING IS DIVIDED IS                       00037400
PASSED BACK TO PARAMETER "TKN", AND THE WORD INDEX (PARAMETER                  00037500
"NF" ) IS INCREMENTED BY 1 FOR EACH TOKEN PLACED IN "FINX".                    00037600
WHEN A SEQUENCE NUMBER IS ENCOUNTERED-                                         00037700
FLG = 1 FOR A SINGLE SEQUENCE NUMBER,                                          00037800
FLG = 2 FOR A SEQUENCE "RANGE"                                                 00037900
NUM2 = OCTAL VALUE OF SEQUENCE NUMBER ( OR UPPER BOUND FOR RANGE )             00038000
NUM1 = OCTAL VALUE OF LOWER BOUND FOR RANGE ( WHERE APPLICABLE ),              00038100
WHEN A FILE NAME IS ENCOUNTERED,                                               00038200
FLG=3 FOR A SINGLE NAME                                                        00038300
FLG=4 FOR A DOUBLE NAME ( A/B )                                                00038400
NUM1 = FIRST NAME                                                              00038500
NUM2 = SECOND NAME ( IF GIVEN )                                                00038600
THE POSITION OF THE INPUT STRING TO BE EXAMINED IS SPECIFIED                   00038700
BY THE ADDRESS IN PARAMETER "ADRS".  THIS PARAMETER IS                         00038800
RE-ASSIGNED THE ADDRESS OF THE NEXT INPUT STRING WHEN                          00038900
```

```
      THE PROCEDURE IS EXITED.                                                  00039000
      THE SIZE OF THE CURRENT STRING IS RETURNED TO "UNPAK"                     00039100
      TO ALLOW COMPARISON BETWEEN THE TOTAL SIZE OF THE INPUT ARRAY             00039200
      AND THE SIZE OF THE STRINGS ALREADY PROCESSED.                            00039300
      END OF COMMENT;                                                           00039400
         BEGIN LOCAL N,T1,T2,SV1,SV2,SV3,DV,TK;                                 00039500
         LABEL CYCLE,XFER,NUMBR,FILL,EXIT;                                      00039600
         DI:=FLG; DS:=8LIT"0"; % RESET THE SEQUENCE NUMBER FLAG                 00039700
         SI:=ADRS; SI:=SI+5; SI:=SC; SV1:=SI; % STARTING ADDRESS                00039800
         SI:=SI+3; DI:=LOC T1; DI:=DI+7; DS:=CHR; % SIZE OF THIS STRING         00039900
         TALLY:=T1; UNPAK:=TALLY; % SIZE TO "UNPAK" ALSO                        00040000
         SI:=SV1; IF SC GTR "6" THEN GO TO FILL; % FILE NAME                    00040100
         IF SC GTR "2" THEN GO TO NUMBR; % SEQUENCE NUMBER FIELD                00040200
         IF SC LSS "3" THEN % A STRING                                          00040300
            BEGIN SI:=NF; DI:=LOC N; DS:=WDS, % STRING INDEX WORD               00040400
            DI:=FINX; N(DI:=DI+8); DV:=DI; % SKIP PREV. WDS, SAVE ADDRES        00040500
            SI:=SV1; SI:=SI+2; IF SC="1" THEN %  "LITERAL" CODE                 00040600
               BEGIN SI:=SI-1; DI:=DI+2; DS:=3 CHR; % "FIRST","LITERAL",SIZE    00040700
               SV2:=SI; SI:=LOC SV2; SI:=SI+5; DS:=3 CHR; % ADDRESS OF STRING   00040800
               TALLY:=N; TALLY:=TALLY+1; N:=TALLY; % INCREMENT INDEX            00040900
               SI:=LOC N; DI:=NF; DS:=WDS; % TRANSFER TO EXTERNAL VARIABLE      00041000
               SI:=SV2; SI:=SI+T1; SV3:=SI; % ADDRESS OF NEXT FIELD            00041100
               TALLY:=1; TK:=TALLY; % "LITERAL" STRING = 1 TOKEN                00041200
               GO TO EXIT;                                                      00041300
               END; % IF "LITERAL"                                              00041400
            % BREAK UP INTO "TOKENS" IF NOT "LITERAL"                          00041500
            SI:=SI+2; SV2:=SI; % POINT TO ACTUAL STRING                         00041600
      CYCLE:                                                                    00041700
            SI:=SV2; TALLY:=T1; % SKIP OVER BLANKS NEXT                         00041800
            T1(IF SC NEQ " " THEN JUMP OUT; SI:=SI+1; TALLY:=TALLY+63);         00041900
            T1:=TALLY; SV2:=SI;                                                 00042000
            SI:=LOC T1; SI:=SI+7; IF SC="0" THEN GO TO EXIT; % END STRING       00042100
            TALLY:=TK; TALLY:=TALLY+1; TK:=TALLY; % TOKEN COUNT, THIS STRING    00042200
            SI:=SV2; IF SC=ALPHA THEN IF SC LSS "0" THEN % STARTS WITH LETTER   00042300
               BEGIN TALLY:=0;                                                  00042400
               T1(IF SC LEQ "9" THEN ELSE JUMP OUT;                             00042500
               IF SC = ALPHA THEN ELSE JUMP OUT;                                00042600
               SI:=SI+1; TALLY:=TALLY+1);                                       00042700
               T2:=TALLY; TALLY:=0; SV3:=SI;                                    00042800
               IF TOGGLE THEN T1:=TALLY ELSE % ALL CHARACTERS USED IF TOGGLE    00042900
                  BEGIN TALLY:=T1; T2(TALLY:=TALLY+63); T1:=TALLY; END;         00043000
      XFER:    SI:=SV1; SI:=SI+1; DI:=DV; DI:=DI+2;                            00043100
               DS:=2CHR; % "FIRST" AND "LITERAL" CODES                          00043200
               SI:=LOC T2; SI:=SI+7; DS:=CHR; % TOKEN SIZE                      00043300
               SI:=LOC SV2; SI:=SI+5; DS:=3 CHR; % ADDRESS OF TOKEN             00043400
               SI:=SV3; SV2:=SI; DV:=DI; % SAVE ADDRESSES                       00043500
               TALLY:=N; TALLY:=TALLY+1; N:=TALLY; % INCREMENT INDEX            00043600
               SI:=LOC N; DI:=NF; DS:=WDS; % TRANSFER TO EXTERNAL VARIABLE      00043700
               GO TO CYCLE;                                                     00043800
               END % IF A LETTER STRING                                         00043900
            ELSE IF SC LEQ "9" THEN % A DIGIT STRING                            00044000
               BEGIN TALLY:=0;                                                  00044100
               T1(IF SC LSS "0" THEN JUMP OUT;                                  00044200
               IF SC GTR "9" THEN JUMP OUT;                                     00044300
               SI:=SI+1; TALLY:=TALLY+1);                                       00044400
               T2:=TALLY; TALLY:=0; SV3:=SI;                                    00044500
               IF TOGGLE THEN % ALL CHARACTERS NOT USED                         00044600
                  BEGIN TALLY:=T1; T2(TALLY:=TALLY+63); T1:=TALLY; END          00044700
               ELSE T1:=TALLY;                                                  00044800
               GO TO XFER; % TRANSFER TOKENS                                    00044900
```

```
          END IF A DIGIT STRING;                                                    00045000
        % A "SPECIAL" CHARACTER                                                     00045100
        TALLY:=T1; TALLY:=TALLY+63; T1:=TALLY; % ONE CHARACTER TOKEN               00045200
        TALLY:=1; T2:=TALLY; SI:=SI+1; SV3:=SI;                                    00045300
        GO TO XFER;                                                                00045400
        END; % IF AN "EXISTING" STRING                                             00045500
NUMBR: % A SEQUENCE NUMBER                                                         00045600
   DI:=FLG; DS:=8LIT"00000001"; % SEQUENCE NUMBER FLAG                             00045700
   IF SC="4" THEN % SEQUENCE "RANGE" SPECIFIED                                     00045800
      BEGIN DI:=FLG; DI:=DI+7; DS:=LIT"2"; % CODE FOR "RANGE"                      00045900
      SI:=SI+3; DI:=LOC N; DI:=DI+7; DS:=CHR; % SIZE OF FIELD                      00046000
      DI:=NUM1; DS:=N OCT;                                                         00046100
      SI:=SI+3; DI:=LOC N; DI:=DI+7; DS:=CHR; % SIZE OF NEXT FIELD                 00046200
      TALLY:=UNPAK; TALLY:=TALLY+4; TALLY:=TALLY+N; UNPAK:=TALLY;                  00046300
      DI:=NUM2; DS:=N OCT; SV3:=SI; GO TO EXIT;                                    00046400
      END; % IF "RANGE" SPECIFIED                                                  00046500
   SI:=SI+3; DI:=LOC N; DI:=DI+7; DS:=CHR; % SIZE OF FIELD                         00046600
   DI:=NUM2; DS:=N OCT; SV3:=SI; GO TO EXIT;                                       00046700
FILE: % A FILE NAME                                                                00046800
   SI:=SI+4; % POINT TO START OF STRING                                            00046900
   DI:=FLG; DS:=8LIT"00000003"; % CODE FOR SINGLE NAME                             00047000
   DI:=NUM1; DS:=8LIT"0       "; DI:=DI-7;                                        00047100
   TALLY:=0; T2:=TALLY;                                                            00047200
   T1(IF SC=ALPHA THEN                                                             00047300
      BEGIN                                                                        00047400
      SV3:=SI; SI:=LOC T2; SI:=SI+7;                                               00047500
      IF SC LSS "7" THEN;                                                          00047600
      SI:=SV3; IF TOGGLE THEN DS:=CHR ELSE SI:=SI+1;                               00047700
      TALLY:=T2; TALLY:=TALLY+1; T2:=TALLY;                                        00047800
      END                                                                          00047900
   ELSE IF SC="/" THEN                                                             00048000
      BEGIN                                                                        00048100
      DI:=FLG; DI:=DI+7; DS:=LIT"4";                                               00048200
      DI:=NUM2; DS:=8LIT"0       "; DI:=DI-7;                                      00048300
      SI:=SI+1; TALLY:=0; T2:=TALLY;                                               00048400
      END                                                                          00048500
   ELSE SI:=SI+1);                                                                 00048600
   SV3:=SI; GO TO EXIT;                                                            00048700
EXIT:                                                                              00048800
   SI:=LOC SV3; DI:=ADRS; DS:=WDS; % ADDRESS OF NEXT FIELD                         00048900
   SI:=LOC TK; DI:=TKN; DS:=WDS; % TRANSFER TOKEN COUNT                            00049000
   END STREAM PROCEDURE UNPAK;                                                     00049100
%*****************************************************************************   00049200
ESP[0]:=0; DISKWAIT(1,ESP,30,COMMON);                                              00049300
COMMON := ESP[0];                                                                  00049400
NAM1   := ESP[5]; NAM2 := ESP[6]; % WORK FILE NAME                                 00049500
SFLG   := ESP[7] NEQ 8; % NOT TYPE DATA FILE                                       00049600
USER   := TIME(-1); % USERCODE                                                     00049700
DISKWAIT(1,ESP,30,COMMON); COMMON := 0;                                            00049800
INFIL      := BOOLEAN(ESP[0].[2:1]); % "FIND FILE"                                 00049900
OUTFIL     := BOOLEAN(ESP[0].[3:1]); % "PRINT FILE"                                00050000
SITE       := BOOLEAN(ESP[0].[4:1]); % "PRINT SITE"                                 00050100
TEXT       := BOOLEAN(ESP[0].[17:1]); % TEXT OR SEQUENCE NUMBERS                    00050200
TOTAL      := TSIZE(ESP); % TOTAL NO. OF CHARACTERS IN INPUT RECORD                 00050300
ESPPTR := ADDRESS(ESP[1]); % STARTING ADDRESS OF STRINGS                            00050400
GRPCOUNT := NR := NF := PREVNF := NRANGES := 0; % INITIALIZE                        00050500
WHILE TOTAL GTR 0 DO % DECODE THE INPUT RECORD                                     00050600
   BEGIN TOTAL := TOTAL - 4 - %  DECREMENT TOTAL                                   00050700
   UNPAK(ESPPTR,FNDINX,NF,TKNS,NUM1,NUM2,FLG);                                     00050800
   IF NF GTR PREVNF THEN % A STRING FOUND                                          00050900
```

Data Documents, Inc.

```
      BEGIN IF NF GTR 100 THEN % TOO MANY TOKENS                          00051000
         BEGIN ERRMSG(OUTBUF,"TOOLONG");                                  00051100
         TWXOUT(OUTBUF[0],13,0); COMMON.[2:1]:=1; GO TO EXIT;             00051200
         END IF TOO MANY TOKENS;                                          00051300
      GRPTKNS[GRPCOUNT := GRPCOUNT + 1] := TKNS; % TOKENS/GROUP           00051400
      PREVNF := NF;                                                       00051500
      END IF A STRING;                                                    00051600
   IF FLG GTR 0 AND FLG LSS 3 THEN % REFERENCE NUMBER(S)                  00051700
      BEGIN NRANGES := NRANGES + 1;                                       00051800
      IF NRANGES GTR MAXRANGES THEN % TOO MANY REFERENCE NUMBERS          00051900
         BEGIN ERRMSG(OUTBUF,"TOOMANY");                                  00052000
         TWXOUT(OUTBUF[0],13,0); COMMON.[2:1]:=1; GO TO EXIT;             00052100
         END IF TOO MANY REFERENCE NUMBERS;                               00052200
      LBOUND[NRANGES]:=UBOUND[NRANGES]:=NUM2;                             00052300
      IF FLG GTR 1 THEN LBOUND[NRANGES]:=NUM1; % SEQ. RANGE               00052400
      END; % IF REFERENCE NUMBER(S).                                      00052500
   IF FLG GTR 2 THEN % FILE NAME(S) GIVEN                                 00052600
      BEGIN                                                               00052700
      IF INFIL AND NOT HAVEFILE THEN                                      00052800
         BEGIN                                                            00052900
         HAVEFILE:=TRUE; NAM1:=NUM1;                                      00053000
         NAM2:=IF FLG=4 THEN NUM2 ELSE USER;                              00053100
         END                                                              00053200
      ELSE IF OUTFIL THEN NAM3:=NUM1;                                     00053300
      END; % IF FLG GTR 2                                                 00053400
   END OF WHILE STATEMENT;                                                00053500
%.....................................................................00053600
FILL OLD WITH NAM1,NAM2;                                                 00053700
IF INFIL THEN IF CHEKFILE(NAM1,NAM2,OUTBUF) LEQ 0 THEN % MISSING OR LOCK 00053800
   BEGIN                                                                 00053900
   TWXOUT(OUTBUF[0],60,1); COMMON.[2:1]:=1; GO TO EXIT;                  00054000
   END;                                                                  00054100
IF OUTFIL THEN IF CHEKFILE(NAM3,USER,OUTBUF) GEQ 0 THEN                  00054200
   BEGIN                                                                 00054300
   TWXOUT(OUTBUF[0],60,1); COMMON.[2:1] := 1; GO TO EXIT;                00054400
   END                                                                   00054500
ELSE FILL DSK2 WITH NAM3,USER;                                           00054600
READ SEEK(OLD[0]); MAX:=HDR(OLD,7); % EOF POINTER                        00054700
IF INFIL THEN SFLG := HDR(OLD,4).[36:6] NEQ 8; % FILE TYPE               00054800
WORK[0]:=0; WRKADRS := ADDRESS(WORK);                                    00054900
GRPMRK  := REAL(NOT FALSE); % LEFT ARROW                                 00055000
EOR     := ADDRESS(WORK[10-REAL(SFLG)]); % END OF RECORD ADDRESS         00055100
LREC := REC := RC := 0;                                                  00055200
IF SEQCHK := NRANGES GTR 0 THEN                                          00055300
   BEGIN LREC := 1; SEQ := -1;                                           00055400
NEXTRANGE:                                                               00055500
   LREC := LREC - 1;                                                     00055600
   LRANGE := LBOUND[RC:=RC+1]; HRANGE := UBOUND[RC];                     00055700
   IF NOT SFLG THEN % RECORD LOCATION SPECIFIED                          00055800
      BEGIN                                                              00055900
      READ SEEK(OLD[LREC:=(LRANGE-1)*REAL(LRANGE NEQ 0)]);               00056000
      GO TO SERIALRD; % SKIP BINARY SEARCH                               00056100
      END;                                                               00056200
   IF SEQ=LRANGE THEN                                                    00056300
      BEGIN LREC:=LREC+1; GO TO HAVEOLD; END;                            00056400
   IF (MAX-LREC) LSS 30 THEN GO TO SERIALRD; % SKIP BINARY SEARCH        00056500
   REC:=LREC+(MAX-LREC) DIV 2; % START IN THE CENTER OF RANGE            00056600
   GO TO RANDOMRD;                                                       00056700
      DO BEGIN                                                           00056800
      IF LRANGE LSS SEQ THEN % MOVE BACK, NOT LSS 0                      00056900
```

```
      REC:=(REC:=REC-ADJ)×REAL(REC.[1:1]=0) ELSE % MOVE UP                  00057000
      IF REC:=(LREC:=REC)+ADJ GTR MAX THEN REC:=MAX; % NOT MORE THAN MAX    00057100
RANDOMRD:                                                                   00057200
      READ(OLD[REC],10,WORK[*]);                                            00057300
      SEQ:=OCTCONV(WORK[9]);                                                00057400
      IF SEQ=LRANGE THEN % MATCHING RECORD                                  00057500
         BEGIN LREC:=REC; READ(OLD); GO TO HAVEOLD; END;                    00057600
      ADJ:=(REC-LREC) DIV 2;                                                00057700
      END UNTIL ADJ LEQ 1;                                                  00057800
   READ SEEK(OLD[LREC]); % NOT FOUND IN BINARY SEARCH, GO BACK              00057900
   END; % IF RANGE SPECIFIED                                                00058000
SERIALRD:                                                                   00058100
READ(OLD,10,WORK[*])[FINISH];                                               00058200
LREC:=LREC+1; % POINTS TO THE NEXT RECORD                                   00058300
IF SEQCHK THEN % CHECK SEQUENCE NUMBERS                                     00058400
   BEGIN                                                                    00058500
   SEQ:=IF SFLG THEN OCTCONV(WORK[9]) ELSE LREC;                            00058600
   IF SEQ LSS LRANGE THEN GO TO SERIALRD; % TOO LOW                         00058700
   IF SEQ GTR HRANGE THEN % TOO HIGH                                        00058800
   IF RC LSS NRANGES THEN GO NEXTRANGE ELSE GO FINISH;                      00058900
   END;                                                                     00059000
HAVEOLD:                                                                    00059100
IF SFLG THEN SHIFTSEQ(WORK[9]) ELSE WORK[10]:=GRPMRK;                       00059200
ENDWORD := -1; PERLINE := 0;                                                00059300
FOR I:=1 STEP 1 UNTIL GRPCOUNT DO % CHECK ALL TOKEN GROUPS                  00059400
   BEGIN                                                                    00059500
   WORKPTR := WRKADRS; % START OF RECORD                                    00059600
   STRTWORD := ENDWORD + 1; % LOCATION IN FNDINX FOR GROUP                  00059700
   ENDWORD := ENDWORD + GRPTKNS[I];                                         00059800
   TOKENCOUNT := COLS := 0; % RESET                                         00059900
   IF BOOLEAN(FNDINX[STRTWORD].[23:1]) THEN GO FINDLIT; % LITERAL           00060000
TOKENLOOP:                                                                  00060100
   IF TOKENLENGTH := ATOKEN(WORKPTR,TOKENPTR,EOR) = 0 THEN                  00060200
   GO TO ENDILOOP; % END OF RECORD                                          00060300
   FILINX[TOKENCOUNT]:=TOKENPTR & TOKENLENGTH[24:42:6]; % LOCATION          00060400
   TOKENCOUNT := TOKENCOUNT + 1;                                            00060500
   IF TOKENCOUNT LSS GRPTKNS[I] THEN GO TO TOKENLOOP;                       00060600
   IF TOKENSMATCH(TOKENCOUNT,FNDINX[STRTWORD],FILINX) THEN                  00060700
      BEGIN % FOUND A MATCH                                                 00060800
      COMMON := COMMON + 1; % NUMBER FOUND                                  00060900
      PERLINE := PERLINE + 1; % NUMBER IN THIS RECORD                       00061000
      TOKENCOUNT := 0; % RESET                                              00061100
      IF BOOLEAN(FNDINX[STRTWORD].[17:1]) THEN  % FIRST                     00061200
         BEGIN PRINT; GO TO FINISH; END;                                    00061300
      GO TO TOKENLOOP; % LOOK AT REMAINDER OF RECORD                        00061400
      END IF TOKENS MATCH;                                                  00061500
   TOKENCOUNT := TOKENCOUNT -1;                                             00061600
   IF TOKENCOUNT GTR 0 THEN % SHIFT LEFT                                    00061700
   MOVEWORDS(TOKENCOUNT,FILINX[1],FILINX[0]);                               00061800
   GO TO TOKENLOOP; % CONTINUE TO END OF RECORD                             00061900
FINDLIT:                                                                    00062000
   COLSLEFT := 80 - 8 × REAL(SFLG) - COLS; % CHARACTERS LEFT TO SCAN        00062100
   OLDSIZE := FNDINX[STRTWORD].[24:6]; % LITERAL STRING SIZE                00062200
   IF COLSLEFT LSS OLDSIZE THEN GO TO ENDILOOP; % TOO SMALL                 00062300
   TOSCAN := COLSLEFT - OLDSIZE +1;                                         00062400
   IF TOSCAN GTR 63 THEN TOSCAN := 63; % 63 IN ONE BREATH                   00062500
   SCANNED := LITSCAN(TOSCAN,WORKPTR,FNDINX[STRTWORD],FOUND);               00062600
   COLS := COLS + SCANNED; % UP TO STRING, IF FOUND                         00062700
   IF FOUND THEN                                                            00062800
      BEGIN                                                                 00062900
```

```
        COLS := COLS + OLDSIZE; % ADD SIZE OF STRING                    00063000
        COMMON := COMMON +1; % TOTAL NO. FOUND                          00063100
        PERLINE := PERLINE + 1; % NO. FOR THIS RECORD                   00063200
        IF PERLINE GTR 79 THEN GO TO ENDILOOP;                          00063300
        IF BOOLEAN(FNDINX[STRTWORD].[17:1]) THEN  % FIRST               00063400
           BEGIN PRINT; GO TO FINISH; END;                              00063500
        END IF FOUND;                                                   00063600
     GO TO FINDLIT;                                                     00063700
ENDILOOP:                                                               00063800
     END I LOOP;                                                        00063900
IF PERLINE GTR 0 THEN PRINT;                                            00064000
GO TO SERIALRD;                                                         00064100
FINISH:                                                                 00064200
IF OUTFIL AND COMMON GTR 0 THEN MOVETYPE(OLD,DSK2);                     00064300
CLOSE(OLD);                                                             00064400
IF OUTFIL AND COMMON GTR 0 THEN LOCK(DSK2,*);                           00064500
TWXOUT(OUTBUF[0],0,1);                                                  00064600
NUMFOUND(OUTBUF,COMMON); TWXOUT(OUTBUF[0],34,1);                        00064700
EXIT:                                                                   00064800
END OF PROGRAM.                                                         00064900
END;END.          LAST CARD ON ORDING TAPE                              99999999
```

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

OBJECT /READ