XBASIC

on Heriot-Watt Univ. distribution tape
April 1977

```
 $ SET SINGLE BEND
 BEGIN
COMMENT::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
:::::                                                             :::::
:::::       XBASIC -- A CONVERSATIONAL BASIC INTERPRETER          :::::
:::::            MK XV 1.04: 1 DECEMBER 1975                       :::::
:::::                                                             :::::
:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::

      XBASIC IS A BASIC INTERPRETER DEVELOPED FOR THE BURROUGHS RANGE
 OF COMPUTERS BY STAFF AT PAISLEY COLLEGE OF TECHNOLOGY.  ITS PURPOSE
 IS TO PROVIDE FAST RESPONSE TO SIMPLE BASIC PROGRAMS UNDER TIME-
 SHARING.  ACTUAL RUNNING OF PROGRAMS IS MUCH SLOWER FOR XBASIC, BUT
 EXPERIENCE SHOWS THAT MOST CLASS TIME IS SPENT ON INPUT AND COMPILE.

      XBASIC IS AN IMPLEMENTATION OF STANDARD BASIC, AND DIFFERS
 FROM BURROUGHS BASIC IN CERTAIN MATTERS OF DETAIL.  THE COMMANDS
 ALLOWED IN XBASIC FORM A SUBSET OF THE COMMANDS AVAILABLE UNDER CANDE.
 FILES ARE EDITABLE UNDER CANDE.  TRACE AND UNTRACE STATEMENTS
 CAN BE MADE AVAILABLE - SEE LINES 83300-83400 AND 107300-108700.
 XBASIC OUTPUT MAY BE DIVERTED FROM REMOTE TO PRINTER ("SEND").
 VIDEO TERMINALS ARE SUPPORTED FOR EASY CORRECTION OF SOURCE PROGRAM.
 PROGRAMS ARE MONITORED FOR EXCESS LOOPING.

       TO USE XBASIC FROM THE BATCH TERMINAL, THE FOLLOWING
 CARDS SHOULD BE SUPPLIED:
           ? EXECUTE OXBASIC/UTILITY
           ? COMMON=2
           ? DATA CRD
               (INSERT DECK HERE:  USE TERMINAL FORMAT)
           ? END

:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::;
COMMENT
   THE BLOCK STRUCTURE FOR THE PROGRAM IS AS FOLLOWS:

             -------------          /------x
             1 SOURCEIN: 1         /  DOES  x
   START-->--1 & EXECUTE 1--RUN->----<  OBJECT  >--YES-->---x
       1     1  COMMANDS 1         x EXIST? /               1
       A     -------------          x------/                V
       1          1                    1                    1
       1          1 (YES)              V (NO)               1
       1        /------x               1                    1
       1       /  ANY   x         -------------             1
       1      <  SYNTAX  >------<----1  COMPILE: 1          1
       1       x ERRORS?/         1           1             1
       1        x------/          -------------             1
       1          1                                         1
       1          V (NO)                                    1
       1          1                                         1
       1       -------------                                1
       x---<---1 EXECUTE:  1----------------<---------------/
               -------------
```

```
:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::;      00005600
                                                                        00005700
INTEGER IU;     % (COMMON)  INPUT UNIT=0 VDU,1 TTY,2 CRD                00005800
                                                                        00005900
INTEGER ARRAY  PROG[0:200,2:11],       % SOURCE STRINGS                 00006000
               SSEQ[0:200],            % SEQUENCE NUMBERS               00006100
               STYP,                   % STATEMENT TYPES                00006200
               SPOB[1:200],            % POINTERS TO OBJECTPROGRAM      00006300
               SUB[1:26],              % INFO ON USER FUNCTIONS         00006400
               ARR[1:26,0:2],          %         ARRAYS                 00006500
               STRAR[1:26,0:1],        %         STRING ARRAYS          00006600
               IOB[1:240],IOF[1:10],   % I/O PSEUDOBUFFERS              00006700
               IOBE[1:14],             %  "         "                   00006800
               KEY[1:38],              % COMPILE KEYWORDS               00006900
               FNM[1:3,1:2];           % FILENAMES FOR EXECUTE          00007000
                                                                        00007100
REAL ARRAY     OBJ[0:999],             % PSEUDO-OBJECTPROGRAM           00007200
               FORX[1:10,1:4],         % INFO ON FOR STATEMENTS         00007300
               CONST[1:1000],          % SOURCE PROGRAM CONSTANTS       00007400
               ANSA[0:9],              % FOR DISK SEARCH                00007500
               IO[0:3];                % FILENAMES                      00007600
                                                                        00007700
INTEGER        MS,             % NUMBER OF LINES OF PROGRAM             00007800
               ACS,            % FIRST EXECUTABLE STATEMENT             00007900
               CS,             % CURRENT STATEMENT NUMBER               00008000
               CHA,            % CURRENT SOURCE CHARACTER               00008100
               CP,             % POSITION OF CHA IS SOURCE              00008200
               LP,             % SEE LOOK                               00008300
               CO,             % POSITION IN OBJ                        00008400
               OU,             % OUTPUT UNIT (O REMOTE, 1 PRINTER)      00008500
               MSTO,           % CURRENT TOP OF ARRAYS                  00008600
               MSTR,           % CURRENT TOP OF STRING ARRAYS           00008700
               NCON,           % NUMBER OF CONSTANTS                    00008800
               DELIM,          % SEE NCH                                00008900
               TIM,            % MAX EXECUTION TIME (2 MIN USUALLY)     00009000
               LL,             % CURRENT LINE NUMBER IN INPUT PAGE      00009100
               AREASIZE,       % FOR EXECUTE OUTPUT FILE IF ANY         00009200
               BEG,            % LIST COMMAND INITIAL LINE              00009300
               EN,             %                FINAL LINE              00009400
               NDEP,           % ARITH STACK COUNTER                    00009500
               ADDR,           % CURRENT VARIABLE ADDRESS               00009600
               FORE,FORC,      % HELP COMPILE FOR NESTS                 00009700
               NF,             % NUMBER OF EXECUTE FILES                00009800
               A,B,C,D,K;      % HASH                                   00009900
                                                                        00010000
REAL           R,S,T;          % HASH                                   00010100
                                                                        00010200
BOOLEAN        OBJECT,         % "THERE IS AN OBJECTPROGRAM"            00010300
               STRIN,          % "EXPRESSION IS A STRING"               00010400
               HDDR,           % "PRINTER HEADER PRINTED"               00010500
               INFILTOG,       % "PROGRAM NEEDS INPUT FILE"             00010600
               OUTFILTOG,      % "PROGRAM NEEDS OUTPUT FILE"            00010700
               FIRSTOFF,       % "HELLO"                                00010800
               DANGER,         % "NEW MATERIAL IN WORKFILE"             00010900
               VAR,            % "EXPRESSION IS A VARIABLE"             00011000
               SY,             % "PROGRAM CONTAINS SYNTAX ERRORS"       00011100
               EQOK,           % "EXPRESSION MAY CONTAIN ="             00011200
```

```
                    AA;             % "SUCCESSIVE EXPNS TO PRINT"          00011300
                                                                          00011400
          POINTER   PINB,           % START OF LINE     IN IOB[*]          00011500
                    PIOB,           % CURRENT CHARACTER    "               00011600
                    PIBE,           % LAST CHARACTER IN IOBE[1]            00011700
                    POB,            % FIRST CHARCTER IN OBJ[*]             00011800
                    APR,BPR,CPR;    % HASH                                 00011900
                                                                          00012000
          FORMAT WHT  ("ERR- ",A6," IS NOT A COMMAND IN XBASIC"),         00012100
                 STP  (/"END ",A6),                                       00012200
                 SPC  (/),                                                00012300
                 WRN  ("MORE THAN 100 GOTOS. IS YOUR PROGRAM ALL RIGHT?"), 00012400
                 REP  (A3),                                               00012500
                 SYER ("SYNTAX ERRORS: CLEAR SCREEN AND TRANSMIT A SPACE"), 00012600
                 SYR  ("SYNTAX ERRORS:"),                                 00012700
                 WT   ("WAIT-"),                                          00012800
                 MESS ("EXECUTING"),                                      00012900
                 INTR ("ILLEGAL NUMBER"),                                 00013000
                 LNGPRG("PROGRAM TOO LONG AT LINE ",I6),                  00013100
                 INVIT ("VDU ASSUMED - ELSE SAY TTY"),                    00013200
                 DVO  ("OUTPUT IS BEING DIVERTED TO PRINTER"),            00013300
                 BK   ("EXECUTION STOPPED - EXCESS TIME."/                00013400
                       "FOR LONG PROGRAMS USE MAIN SYSTEM"),              00013500
                 SNUM (X72,I8),                                           00013600
                 F1   ("USE RENAME XXXXXX OR SAVE XXXXXX COMMAND"),       00013700
                 F2   ("ERR?  THIS WILL DELETE THE WORKFILE"),            00013800
                 F3   ("OK- ",I3," RECORDS",A6,"D, LAST RECORD =",I7),    00013900
                 F4   ("FILE ",A6," - ",A6,A1," BY XBASIC"),              00014000
                 F5   ("YOUR WORKFILE IS AS AT LAST RUN COMMAND"),        00014100
                 F6   (I6,X3,"DIM OR FILES STATEMENT OUT OF SEQUENCE"/    00014200
                       "ERR RUN"),                                        00014300
                 F7   ("WORKFILE NOW EMPTY"),                             00014400
                 HD1  ("FURTHER OUTPUT WILL BE PRINTED WHEN YOU SIGN OFF"), 00014500
                 HD3  (X40,"XBASIC  MK XV",X8,"RUN  ",A6,"DAY  ",         00014600
                       A2,"/",A2,"/",A2,/X54,"USER NO. ",A3,A4,/X52,20("*")//), 00014700
                                                                          00014800
                 F9   ("XBASIC IS RUNNING-"),                             00014900
                 F10  ("UNNAMED WORKFILE HAS",I4," RECORDS, LAST RECORD =",I6), 00015000
                 F11  (A6," (WORKFILE) HAS",I4," RECORDS, LAST RECORD =",I6), 00015100
                 F12  ("ERR- ILLEGAL PARAMETER"),                         00015200
                 F13  ("ARE PRESENT CONTENTS OF FILE ",A6," TO BE DESTROYED?"); 00015300
                                                                          00015400
          SWITCH FORMAT NUM:=(U10),(U6),(X20,U10);                        00015500
          SWITCH FORMAT MNP:=("NOT ENOUGH INPUT AT LINE",I6,X5,"TRY AGAIN"), 00015600
                            ("BLANK INPUT AT LINE",I6,X5,"IGNORED");      00015700
                                                                          00015800
          FILE TTY 19(2,10);FILE IN CRD 2(2,10);FILE IN VDU 19(1,240);    00015900
          FILE OUT LIN 1(2,14);SWITCH FILE FN:=TTY,CRD;SWITCH FILE FL:=TTY,LIN; 00016000
                                                                          00016100
          MONITOR INTOVR,EXPOVR,INDEX,FLAG,ZERO;                          00016200
                                                                          00016300
          LABEL SOURCEIN,COMPILE,EXECUTE,STOP,FINSH,ERR,INCST,PER,INER,   00016400
          TOOLONG;                                                        00016500
                                                                          00016600
          LABEL EXS,EQL,LET,CAR,ONX,RON,IFF,FEQ,GOT,GOS,RET,FOX,NEX,      00016700
          DEF,REA,RREA,INP,RAN,RFS,PRI,RPRI,XPRI,DIM,ENX,REM,RFIL,        00016800
          INTVR,QUOTE,RDUM,DAT,RDAT,RDIM,PAG,MAT,IOMT,FLAGR,FIL,CGO;      00016900
```

```
SWITCH OPN:=LET,GOT,GOS,RET,INP,REA,PRI,FOX,NEX,MAT,                          00017000
DEF,DAT,RAN,PAG,RES,REM,ENX,ENX,IFF,ONX;                                      00017100
                                                                              00017200
                                                                              00017300
DEFINE ON(ON1)=IF CHA=ON1 THEN #;                                             00017400
                                                                              00017500
COMMENT:::::::::::::::GLOBAL PROCEDURES:::::::::::::::::::::::::::::::::::   00017600
                                                                              00017700
      ---     CHCONV  CONVERTS CHA TO LETTER CODE 1-26 ;                      00017800
                                                                              00017900
INTEGER PROCEDURE CHCONV(A);VALUE A; INTEGER A;                               00018000
BEGIN CHCONV:=0;IF A GTR 16 AND A LSS 26 THEN CHCONV:=A-16 ELSE               00018100
IF A GTR 32 AND A LSS 42 THEN CHCONV:=A-23 ELSE                               00018200
IF A GTR  49 AND A LSS 58 THEN CHCONV:=A-31 END;                              00018300
COMMENT                                                                       00018400
      ---     NCH     PICKS NEXT CHARACTER FROM SOURCE STRING                 00018500
                      AND STORE IT IN CHA.  BLANKS ARE SKIPPED.               00018600
                      IF LAST CHAR= RETURN "%" AT DELIM.                      00018700
                      CP IS UPDATED.   IOBE[*] IS USED AS HASH.;              00018800
                                                                              00018900
INTEGER PROCEDURE NCH;                                                        00019000
BEGIN INTEGER A;LABEL RPT; POINTER CPR;                                       00019100
IOBE[1]:=0;A:=CP;                                                             00019200
     RPT:    IF A GEQ DELIM THEN CHA:=NCH:="%" ELSE BEGIN                     00019300
CPR:=POINTER(PROG[CS,2])+A;                                                   00019400
REPLACE PIBE BY CPR:CPR FOR 1;A:=A+1;                                         00019500
IF IOBE[1]=48 THEN GO TO RPT;CHA:=NCH:=IOBE[1];CP:=A                          00019600
END END;                                                                      00019700
COMMENT                                                                       00019800
      ---     NMBR      PICKS UP STATEMENT NUMBER          ;                  00019900
INTEGER PROCEDURE NMBR(N);VALUE N;INTEGER N;                                  00020000
BEGIN LABEL DONE,RNB,BLK,NST,SKB;                                             00020100
DEFINE RD(RD1)=REPLACE BPR BY APR:APR FOR 1;                                  00020200
               IF DELTA(PINB,APR) GEQ RD1 THEN GO BLK;IF IOBE[1]#;            00020300
CHA:=NMBR:=IOBE[1]:=0;                                                        00020400
    NST:   RD(N)=48 THEN GO NST;IF (CHA:=IOBE[1]) GEQ 10 THEN GO DONE;        00020500
    RNB:   RD(72) LSS 10 THEN BEGIN CHA:=CHA×10+IOBE[1];GO RNB END;           00020600
NMBR:=CHA;CHA:=IOBE[1];IF CHA NEQ 48 THEN GO DONE;                            00020700
    SKB:   RD(72)=48 THEN GO SKB;CHA:=IOBE[1];GO DONE;                        00020800
                                                                              00020900
                                                                              00021000
     BLK:    CHA:="%";                                                        00021100
    DONE:    END;                                                             00021200
COMMENT                                                                       00021300
      ---     FILECONTROL     DEALS WITH SOURCE FILE OPERATIONS               00021400
                              A=0 "MAKE"/"RENAME",                            00021500
                                1 "SAVE", 2 "LOAD"/"COPY",                    00021600
                                3 "REMOVE", 4 SAVE WORKFILE(AT "RUN")         00021700
                                5 LOAD WORKFILE(AT XBASIC ENTRY),             00021800
                                6 EXPLICIT REMOVE (AT "BYE",ETC)              00021900
                                7 EXPLICIT SAVE, 8 EXPLICIT LOAD;             00022000
                                                                              00022100
                                                                              00022200
PROCEDURE FILECONTROL(A,C,D,L);VALUE A,C,D;INTEGER A;REAL C,D;LABEL L;        00022300
      % C AND D CARRY LABEL EQUATE FOR EXPLICIT OPERATIONS                     00022400
BEGIN INTEGER B,X,Y;                                                          00022500
PROCEDURE FILERR(E);VALUE E;INTEGER E;                                        00022600
```

```
BEGIN SWITCH FORMAT ERR:=("ERR- ",A3,A4,"/",A3,A4,"- NOT ON DISK"),   00022700
      ("ERR- ",A3,A4,"/",A3,A4," - INVALID USER"),                   00022800
      ("ERR- ",A3,A4,"/",A3,A4," - NON-STANDARD"),                   00022900
      ("ERR- ",A3,A4,"/",A3,A4," - ILLEGAL NAME"),                   00023000
      ("ERR- ",A3,A4,"/",A3,A4," - DUPLICATE NAME"),                 00023100
      ("ERR- NO FILENAME"),                                          00023200
      ("ERR- WORKFILE"),                                             00023300
      ("ERR- WORKFILE IS EMPTY");                                    00023400
                                                                     00023500
 IF E>4 THEN WRITE(TTY,ERR[E]) ELSE WRITE(TTY,ERR[E],                00023600
 FOR X:=B,2 DO [IO[X].[41:18],IO[X].[23:24]]);                       00023700
 IF B=0 THEN IO[0]:=0;IF A=1 AND E=3 THEN WRITE(TTY,F1);             00023800
 GO SOURCEIN END;                                                    00023900
LABEL SKIP,MK,SV,LD,RM,EF,SW,LW,EW;                                  00024000
SWITCH OP:=MK,SV,LD,RM,SW,LW,RM,SV,LD;                               00024100
FILE DSK DISK "XBWKFL "(2,10,300,SAVE 7);% NB LABEL EQN ABOVE SKIP   00024200
IF DANGER AND (C="CREATE" OR A=2 OR A=6) THEN BEGIN DANGER:=FALSE;   00024300
WRITE(TTY,F2);GO SOURCEIN END;                                       00024400
B:=IF A=0 THEN 0 ELSE 1;IF A>3 THEN BEGIN                            00024500
FILL DSK WITH C,D;IO[1]:=C;IO[2]:=D;GO SKIP END;                     00024600
IO[B]:="       ";                  % FILENAME                        00024700
SCAN APR:APR FOR 5 UNTIL NEQ " ";IF DELTA(PINB,APR)<12 THEN          00024800
REPLACE POINTER(IO[B])+1 BY APR:APR FOR IF A<2 THEN 6 ELSE 7         00024900
WHILE IN ALPHA;                                                      00025000
IF IO[B]="       " THEN BEGIN    % ON SAVE SAVE WKFILE IF NO NAME     00025100
IF A=1 AND IO[0] NEQ 0 THEN IO[1]:=IO[0] ELSE FILERR(5) END;         00025200
IO[2]:=0;IF A GTR 1 THEN BEGIN     % PROCESS "/USERCODE" IF PRESENT   00025300
SCAN CPR:CPR:=APR FOR 10 WHILE NEQ "/";IF DELTA(PINB,CPR) LSS 15 THEN 00025400
BEGIN IO[2]:="      ";                                               00025500
REPLACE POINTER(IO[2])+1 BY APR:CPR+1 FOR 7 WHILE IN ALPHA END END;  00025600
IF IO[2]=0 THEN IO[2]:=TIME(-1);                                     00025700
IF IO[B]="XBWKFL " OR IO[B].[41:36] LSS "A00000" THEN FILERR(3);     00025800
        FILL DSK WITH IO[B],IO[2];DSK.SAVE:=7;                       00025900
      SKIP:  IF NOT FIRSTOFF THEN BEGIN X:=CHA;Y:=NMBR(72);          00026000
IF Y NEQ 0 OR CHA NEQ "%" THEN GO PER;CHA:=X END;                    00026100
SEARCH(DSK,ANSA[*]);IF (A=5 OR A=6) AND ANSA[0]=-1 THEN GO L;        00026200
IF A=0 AND ANSA[0] NEQ -1 THEN FILERR(4);                            00026300
IF A=1 AND IO[1] NEQ IO[0] AND ANSA[0] NEQ -1 THEN FILERR(4);        00026400
IF A>1 AND A NEQ 4 THEN BEGIN IF ANSA[0] LEQ 0 THEN FILERR(ANSA[0]+1); 00026500
IF ANSA[3] NEQ 10 OR ANSA[4] NEQ 300 THEN FILERR(2) END;             00026600
IF CHA="MAKE00" THEN BEGIN MS:=0;OBJECT:=FALSE END;                  00026700
GO OP[A+1];                                                          00026800
     SV:    IF IO[0]=0 THEN IO[0]:=IO[B];   % NAME WORKFILE          00026900
IF MS=0 THEN FILERR(7);IF ANSA[0] NEQ -1 AND ANSA[0] NEQ 7 THEN      00027000
FILERR(1);DSK.AREAS:=20;DSK.AREASIZE:=10;                            00027100
FOR CS:=1 STEP 1 UNTIL MS DO BEGIN          % SAVE IT IN CANDE FORMAT 00027200
REPLACE POINTER(IOBE[10]) BY SSEQ[CS] FOR 8 DIGITS;                  00027300
REPLACE POINTER(IOBE[*]) BY POINTER(PROG[CS,2]) FOR 9 WORDS;         00027400
WRITE(DSK,10,IOBE[*]) END;LOCK(DSK);DANGER:=FALSE;GO MK;             00027500
     LD:    FOR MS:=1 STEP 1 UNTIL 200 DO BEGIN          % LOAD FROM  00027600
READ(DSK,10,IOBE[*])[EF];READ(IOBE[*],SNUM,SSEQ[MS]);  % CANDE FORMAT 00027700
WRITE(PROG[MS,*],9,IOBE[*]);PROG[MS,11]:=SSEQ[MS] END;               00027800
     EF:    MS:=MS-1;OBJECT:=FALSE;LOCK(DSK);                        00027900
WRITE(TTY,F3,MS,C,SSEQ[MS]);                                         00028000
ON("LOAD00") IO[0]:=IF IO[2]=TIME(-1) THEN IO[1] ELSE 0;GO MK;       00028100
     RM:    IF IO[B]=IO[0] AND A=3 THEN BEGIN IF MS=0 THEN           00028200
IO[0]:=0 ELSE FILERR(6) END;IF ANSA[0] NEQ 7 THEN FILERR(1);         00028300
```

```
WRITE(DSK,*,0);CLOSE(DSK,PURGE);GO MK;                                  00028400
     SW:    DSK.AREAS:=20;DSK.AREASIZE:=11;                             00028500
IF MS=0 THEN FILERR(7);                                                 00028600
WRITE(DSK,*,IO[0]);FOR CS:=1 STEP 1 UNTIL MS DO                         00028700
WRITE(DSK,10,PROG[CS,*]);LOCK(DSK);DANGER:=FALSE;GO MK;                 00028800
     LW:    READ(DSK,*,IO[0]);FOR MS:=1 STEP 1 UNTIL 200 DO             00028900
BEGIN READ(DSK,10,PROG[MS,*])[EW];SSEQ[MS]:=PROG[MS,11] END;            00029000
     EW:    LOCK(DSK);MS:=MS-1;OBJECT:=FALSE;GO MK;                     00029100
      MK:    IF A<4 AND CHA NEQ "COPY00" THEN                           00029200
WRITE(TTY,F4,IO[B].[41:36],C,D);                                        00029300
IF A=5 THEN WRITE(TTY,F5);                                              00029400
IF CHA="COPY00" THEN WRITE(TTY,STP,"COPY  ");                           00029500
GO L END;                                                               00029600
COMMENT                                                                 00029700
      ---      SYNT    DEALS WITH SYNTAX ERRORS                 ;       00029800
                                                                        00029900
PROCEDURE SYNT(A);VALUE A;REAL A;                                       00030000
BEGIN IF SY THEN BEGIN IF IU=0 THEN BEGIN WRITE(TTY,SYER);              00030100
READ(TTY[STOP]) END ELSE WRITE(TTY,SYR);SY:=FALSE END;                  00030200
REPLACE APR:=POINTER(IOBE[*]) BY " " FOR 72;                            00030300
WRITE(IOBE[*],NUM[0],SSEQ[CS]);SCAN APR:APR WHILE NEQ 48;               00030400
IF IU=0 THEN BEGIN REPLACE APR:APR BY POINTER(PROG[CS,*]) FOR 72;        00030500
APR:=POINTER(IOBE[*])+72;                                               00030600
REPLACE APR:APR BY "≠" FOR 1 END ELSE APR:=APR+3;                       00030700
REPLACE POB BY A FOR 8;REPLACE APR BY POB+1 FOR 7;                      00030800
OBJ[1]:=0;REPLACE POB+7 BY POINTER(IOBE[*])+79 FOR 1;                   00030900
IF IU GTR 0 THEN WRITE(TTY,9,IOBE[*]) ELSE IF OBJ[1]=48 THEN            00031000
WRITE(TTY,10,IOBE[*]) ELSE WRITE(TTY[NO],10,IOBE[*]);GO TO ERR END;     00031100
COMMENT                                                                 00031200
      ---      NWC     MODIFIES NCH FOR COMPILE                 ;       00031300
                                                                        00031400
INTEGER PROCEDURE NWC;                                                  00031500
BEGIN ON("%") SYNT("MISG OP");NWC:=NCH END;                             00031600
                                                                        00031700
COMMENT                                                                 00031800
      ---      PUT     STORES CHARACTER IN OBJ                  ;       00031900
                                                                        00032000
PROCEDURE PUT(A);VALUE A;INTEGER A;                                     00032100
BEGIN IF A>63 THEN SYNT("STR >63");                                     00032200
IOBE[1]:=A;REPLACE POB+CO BY PIBE FOR 1;CO:=CO+1;                       00032300
IF CO GEQ 8000 THEN GO TO TOOLONG;IF CO MOD 8=0 THEN CO:=CO+1 END;      00032400
COMMENT                                                                 00032500
      ---     RED      MOVES BACK ONE SPACE IN OBJ;                     00032600
                                                                        00032700
DEFINE RED=CO:=IF CO.[2:3]=1 THEN CO-2 ELSE CO-1#;                      00032800
                                                                        00032900
COMMENT                                                                 00033000
      ---      LOOK    LOOKS AT A STRING IN SOURCE PROG    ;           00033100
                                                                        00033200
INTEGER PROCEDURE LOOK(A);VALUE A;INTEGER A;                            00033300
BEGIN INTEGER B,C,D,E;E:=C:=CHA;B:=CP;                                  00033400
FOR D:=1 STEP 1 UNTIL A-1 DO IF CHA NEQ "%" THEN C:=C×64+NWC;           00033500
LOOK:=C;LP:=CP;CP:=B;CHA:=E END;                                        00033600
COMMENT                                                                 00033700
      ---      NUMB    PICKS UP DIM AND MAT SIZES          ;           00033800
                                                                        00033900
INTEGER PROCEDURE NUMB;                                                 00034000
```

```
BEGIN LABEL RP;INTEGER A;                                                    00034100
A:=0;                                                                        00034200
     RP:    IF NWC LSS 10 THEN BEGIN A:=A×10+CHA;GO TO RP END;               00034300
IF A=0 THEN SYNT("IL STMT");                                                 00034400
NUMB:=A END;                                                                 00034500
COMMENT                                                                      00034600
       ---      ARITH(1)   CONVERTS EXPNS TO REVERSE POLISH   ;              00034700
                                                                             00034800
PROCEDURE ARITH(TT);VALUE TT;INTEGER TT;                                     00034900
BEGIN INTEGER I,A,B,C,J,K,OP,STCK;BOOLEAN INMOK,VOK,STROK;                   00035000
LABEL S,SS,RPT,EXPON,DONE,FINEV,FORM1,FORM2,FORM3,FIN,RP;                    00035100
INTEGER  ARRAY OPK[1:20];                                                    00035200
                                                                             00035300
COMMENT         AN EXPRESSION CONSISTS OF A NUMBER OF PRIMARIES              00035400
       SEPARATED BY OPERATORS, POSSIBLY PRECEDED BY INITIAL - .              00035500
              PRIMARIES:                          OPERATORS:                 00035600
       NUMBER                                       +                      00035700
       VARIABLE                                     -                      00035800
       ARRAY WITH SUBSCRIPT(S)                      *                      00035900
       FUNCTION WITH PARAMETER(S)                   /                      00036000
       STRING                                       **                     00036100
       STRING VARIABLE                              =                      00036200
       STRING ARRAY WITH SUBSCRIPT                                         00036300
       EXPRESSION IN BRACKETS                                              00036400
                                                                           00036500
              EXPRESSION HAS TYPE REAL UNLESS STRING PRIMARY               00036600
       OCCURS:  IF SO IT HAS TYPE STRING.  A VARIABLE PRIMARY IS           00036700
       A PRIMARY OTHER THAN NUMBER, FUNCTION, OR STRING.                   00036800
              IN OBJ AN EXPRESSION HAS FORM                                00036900
                     A OPS A OPS ... A OPS 0                               00037000
       WHERE A IS A CHARACTER BETWEEN 1 AND 16 GIVING THE PRIMARY          00037100
       OR OPERATOR TYPE.  OPS IS 0-2 CHARACTERS GIVING MORE INFORMATION    00037200
       DEPENDING ON A. ;                                                   00037300
                                                                           00037400
STRIN:=FALSE;NDEP:=NDEP+1;IF NDEP=10 THEN SYNT("SIMPLFY");                  00037500
STCK:=OP:=0;INMOK:=VOK:=TRUE;STROK:=TT=0;                                    00037600
                                                                             00037700
COMMENT                                          PRIMARIES:      ;           00037800
                                                                             00037900
     SS:    CHA:=NWC;                                                        00038000
COMMENT                                          BRACKETED EXPRESSION;       00038100
      S:    ON(29) BEGIN ARITH(1);IF CHA NEQ 45 THEN SYNT("NO ) ,A");        00038200
RED;GO TO FORM1 END;                                                         00038300
COMMENT                                          1   NUMBER      ;           00038400
IF CHA LSS 10 OR CHA = 26 THEN BEGIN R:=I:=0;                                00038500
      RPT:    ON(26) BEGIN I:=1;CHA:=NWC END;                                00038600
IF CHA GEQ 10 THEN GO TO EXPON;IF I>0 THEN BEGIN R:=R+CHA×10*(-I);           00038700
I:=I+1 END ELSE R:=R×10+CHA;CHA:=NWC; GO TO RPT;                             00038800
      EXPON:    IF CHA NEQ 21 THEN GO TO DONE;I:=1;CHA:=NWC;J:=0;            00038900
ON(44) BEGIN I:=-1;CHA:=NWC END ELSE ON(16) CHA:=NWC;J:=CHA;                 00039000
IF CHA GEQ 10 THEN SYNT("NO EXPN");CHA:=NWC;                                 00039100
IF CHA LSS 10 THEN BEGIN J:=J×10+CHA;CHA:=NWC END;R:=R×10*(I×J);             00039200
      DONE:    PUT(1);NCON:=NCON+1;CONST[NCON]:=R;                           00039300
PUT(NCON.[11:6]);PUT(NCON.[5:6]);GO FORM2 END;                               00039400
COMMENT                                          8   INITIAL -   ;           00039500
ON(44) BEGIN IF NOT INMOK THEN SYNT ("MISPL -");PUT(8);                      00039600
GO TO FORM2 END;                                                             00039700
```

```
COMMENT                                          16  STRING      ;          00039800
ON(63) BEGIN IF NOT STROK THEN SYNT("ILL STR");PUT(16);                     00039900
SCAN CPR:APR:=POINTER(PROG[CS,2])+CP WHILE NEQ 63;A:=DELTA(APR,CPR);        00040000
STRIN:=TRUE;IF A>14 THEN SYNT("LONGSTR");PUT(A);PUT(CP);                    00040100
CP:=CP+A+1;CHA:=NWC;VAR:=FALSE;GO TO FORM3 END;                             00040200
IF CHCONV(CHA)=0 THEN SYNT("ILL NUM");                                      00040300
B:=CHA;A:=LOOK(2) MOD 64;IF CHCONV(A) NEQ 0 THEN BEGIN                      00040400
A:=LOOK(3) MOD 4096;CHA:=B;                                                 00040500
IF A NEQ "ST" AND A NEQ "TH" AND A NEQ"TO" AND A NEQ "GO" THEN              00040600
BEGIN LABEL EQL,FNQ,RDUM;                                                   00040700
COMMENT                                          5   STANDARD FNS;          00040800
INTEGER B,AS,AP;                                                            00040900
B:=LOOK(3);CP:=LP;                                                          00041000
FOR A:=21 STEP 1 UNTIL 32 DO IF B=KEY[A] THEN GO TO EQL;                    00041100
GO TO FNQ;                                                                  00041200
     EQL:    IF NWC NEQ 29 THEN SYNT("NO PARM");ARITH(1);RED;               00041300
PUT(5);PUT(A-2);                                                            00041400
IF CHA NEQ 45 THEN SYNT("NO )  A");GO TO FORM1;                             00041500
COMMENT                                          4   USER FNS    ;          00041600
     FNQ:    K:=B DIV 64; IF K NEQ "FN" THEN SYNT("UNRC FN");               00041700
B:=CHCONV(B MOD 64);IF SUB[B]=0 THEN SYNT("UNDC FN");                       00041800
IF NWC NEQ 29 THEN SYNT("NO PARM");A:=0;                                    00041900
    RDUM:    A:=A+1;ARITH(1);RED;ON(58) GO TO RDUM;                         00042000
IF CHA NEQ 45 OR A NEQ SUB[B] THEN SYNT("PARAMTR");                         00042100
CHA:=NWC;PUT(4);PUT(B);GO TO FORM2 END END;                                 00042200
B:=CHCONV(B);IF B=0 THEN SYNT("INV VAR");CHA:=NWC;                          00042300
COMMENT                                          3   ARRAY       ;          00042400
ON(29) BEGIN IF ARR[B,1]=0 THEN SYNT("UNDC AR");                            00042500
ARITH(1);RED;ON(58) BEGIN IF ARR[B,2]=0 THEN SYNT("SUBSCPT");               00042600
ARITH(1);RED END;IF CHA NEQ 45 THEN SYNT("SUBSCPT");STROK:=FALSE;           00042700
PUT(3);PUT(B);CHA:=NWC END                                                  00042800
ELSE ON("$") BEGIN IF NOT STROK THEN SYNT("ILL STR");                       00042900
COMMENT                                          15  STRING ARRAY;         00043000
CHA:=NWC;ON(29)BEGIN IF STRAR[B,1]=0 THEN SYNT("UNDSTAR");ARITH(1);         00043100
RED;                                                                        00043200
IF CHA NEQ 45 THEN SYNT("NO ) ,S");PUT(15);CHA:=NWC END ELSE                00043300
PUT(14);PUT(B);STRIN:=TRUE;VAR:=VOK;GO TO FORM3 END                         00043400
ELSE BEGIN PUT(2);PUT(B);IF CHA LSS 10 THEN BEGIN PUT(C:=CHA+1);            00043500
COMMENT                                          14 STRING VBLE             00043600
                                                 2 VARIABLE      ;          00043700
CHA:=NCH END ELSE PUT(C:=0);IF STCK=0 THEN ADDR:=11×(B-1)+C;                00043800
STROK:=FALSE END;VAR:=VOK;GO TO FORM3;                                      00043900
     FORM1:    CHA:=NWC;                                                    00044000
     FORM2:     STROK:=VAR:=FALSE;IF STRIN THEN SYNT("ILL STR");            00044100
COMMENT                                          7-13 OPERATORS:            00044200
        DANGER:  REVERSE POLISH SECTION                          ;          00044300
     FORM3:    BEGIN LABEL RPT,TEST,BOP,XOP;                                00044400
STCK:=STCK+1;INMOK:=FALSE;                                                  00044500
     RPT:    I:=0;ON(16) I:=3 ELSE ON(44) I:=4 ELSE                         00044600
ON(43) BEGIN IF NWC=43 THEN I:=7 ELSE BEGIN CP:=CP-1;I:=5 END;              00044700
END ELSE ON(49) I:=6 ELSE IF CHA=61 AND EQOK THEN BEGIN INMOK:=TRUE;        00044800
IF NOT VAR THEN SYNT("ILL ASN");I:=1 END;VOK:=I LEQ 1;                      00044900
IF NOT VOK THEN BEGIN STROK:=FALSE;IF STRIN THEN SYNT("ILL STR") END;       00045000
     TEST:    IF OP=0 THEN GO TO BOP; IF OP LSS 0                          00045100
THEN SYNT("  ARITH");                                                       00045200
J:=OPK[OP];IF I×J NEQ 1 AND (I+1) DIV 2 LEQ (J+1) DIV 2 THEN BEGIN          00045300
OP:=OP-1;GO TO XOP END;                                                     00045400
```

```
      BOP:    IF I=0 THEN GO TO FIN;OP:=OP+1;OPK[OP]:=1;GO TO SS;        00045500
      XOP:    VAR:=FALSE;STCK:=STCK-1;                                   00045600
PUT(J+6);IF STCK LEQ 0 THEN SYNT("  ARITH");                           00045700
GO TO TEST END;                                                        00045800
COMMENT                                          6   END EXPN    ;     00045900
      FIN:    NDEP:=NDEP-1;IF STCK NEQ 1 THEN SYNT("  ARITH");        00046000
PUT(0) END;                                                            00046100
COMMENT                                                                00046200
      ---     SKIP    SKIPS GIVEN STRING IF FOUND              ;       00046300
                                                                       00046400
PROCEDURE SKIP(A,B);VALUE A,B;INTEGER A,B;                             00046500
BEGIN INTEGER C,D,E;E:=CP;C:=NWC;                                      00046600
FOR D:=1 STEP 1 UNTIL A-1 DO IF CHA NEQ "%" THEN C:=C×64+NWC;          00046700
IF C NEQ B THEN BEGIN CP:=E;CHA:=0 END END;                            00046800
COMMENT                                                                00046900
      ---     CHMAT   CHECK USED IN MAT STATEMENT              ;       00047000
                                                                       00047100
INTEGER PROCEDURE CHMAT(A);VALUE A;INTEGER A;                          00047200
BEGIN A:=CHCONV(A);IF A=0 THEN SYNT("ILL ARR");                        00047300
IF ARR[A,1]=0 THEN SYNT("UNDC AR");IF ARR[A,2]=0 THEN                  00047400
SYNT("   TYPE");CHMAT:=A END;                                          00047500
                                                                       00047600
                                                                       00047700
COMMENT       ----   CFN   FOR FILE INPUT AND OUTPUT;                  00047800
                                                                       00047900
PROCEDURE CFN;                                                         00048000
BEGIN LABEL L,M;                                                       00048100
CHA:=NWC;IF LOOK(4)="FILE" THEN BEGIN CP:=LP;D:=IF A=15 THEN 5 ELSE A; 00048200
FOR C:=1 STEP 1 UNTIL NF DO BEGIN CHA:=NWC;B:=LOOK(FNM[2,C]);          00048300
IF B=FNM[1,C] THEN BEGIN IF FNM[3,C]=0 THEN FNM[3,C]:=D ELSE IF D NEQ  00048400
FNM[3,C] THEN SYNT("IN+OUT?");CP:=LP;PUT(A);IF D=5 THEN                00048500
INFILTOG:=TRUE;IF A=7 THEN OUTFILTOG:=TRUE;                            00048600
IF NWC=":" THEN BEGIN A:=NUMB;FOR B:=ACS STEP 1 UNTIL MS DO            00048700
IF A=SSEQ[B] THEN BEGIN PUT(B.[11:6]);PUT(B.[5:6]);GO ON(58) L ELSE M  00048800
END;SYNT("UNDF GO") END;PUT(0);PUT(0);ON(58) GO L;                      00048900
GO TO M END ELSE BEGIN CP:=CP-1;CHA:=0 END END;SYNT("UNDC FL") END;    00049000
PUT(0);ON("%") GO TO M;CP:=CP-1;CHA:=0;GO TO L;                        00049100
    M:    IF A=5 THEN SYNT("NOINPUT");PUT(0);GO TO INCST;              00049200
     L:    END;                                                        00049300
                                                                       00049400
COMMENT-------------------------------------------------------------   00049500
----------       XBASIC STARTS HERE        ----------------           00049600
-------------------------------------------------------------;         00049700
                                                                       00049800
FIRSTOFF:=IU NEQ 2;IF IU=2 THEN TTY.TYPE:=1;                           00049900
PIBE:=POINTER(IOBE[1])+7; PINB:=POINTER(IOB[1]);                       00050000
DELIM:=72;TIM:=10800;LL:=-1;                                           00050100
OBJECT:=HDDR:=FALSE;OU:=0;                                             00050200
                                                                       00050300
FILL KEY[*] WITH "LET","GOT","GOS","RET","INP",                        00050400
"REA","PRI","FOR","NEX","MAT","DEF","DAT",                             00050500
"RAN","PAG","RES","REM","STO","END","IF ","ON ","SIN","COS",           00050600
"TAN","ATN","EXP","LOG","ABS","INT","SQR","FIX","SGN","RND",           00050700
"EQ","LT","LE","GT","GE","NE";                                         00050800
                                                                       00050900
COMMENT-------------------------------------------------------------   00051000
-----------SOURCEIN:    FOR INPUT OF SOURCE PROGRAM    -------         00051100
```

```
------------               AND EXECUTION OF COMMANDS       -------
------------------------------------------------------------------;

COMMENT:           COMMANDS ALLOWED IN XBASIC

          HELLO         SAME AS BYE
          BYE           TERMINATES XBASIC.  PRINTER OUTPUT IS SCHEDULED
          RUN           EXECUTES PROGRAM IF FREE OF SYNTAX ERRORS
          SCR           DELETES WORKFILE
          DELETE        SAME AS SCR
          LIST          LISTS ENTIRE WORKFILE
          LIST E        WHERE E IS A NUMBER OF ELEMENTS OF FORM
                        N    OR     M-N     (M,N STATEMENT NUMBERS)
                        SEPARATED BY COMMAS.   LISTS PART OF PROGRAM
          MAKE NNNNNN   INITIALISES AND NAMES WORKFILE
          SAVE          SAVES WORKFILE IF NAMED
          SAVE NNNNNN   SAVES WORKFILE IN NNNNNN. NAMES WORKFILE
                          IF NOT ALREADY NAMED
          LOAD NNNNNN            LOADS WORKFILE AND NAMES IT
          LOAD NNNNNN/UUUUUUU    COPIES WORKFILE FROM NNNNNN/UUUUUUU
                                 WORKFILE BECOMES UNNAMED
          COPY NNNNNN            COPIES NNNNNN INTO WORKFILE
          COPY NNNNNN/UUUUUUU    SAME FOR NNNNNN/UUUUUUU
          REMOVE NNNNNN          REMOVES FILE NNNNNN/USER NO.
          RENAME NNNNNN          RENAMES WORKFILE
          PLOP          RESETS WORKFILE TO LAST RUN STATUS
          WHATS         OBTAINS WORKFILE STATUS
          TTY           INPUT UNIT IS TTY
          VDU           INPUT UNIT IS VDU
          SEND          DIVERTS OUTPUT TO PRINTER
          NOSEND        TERMINATES DIVERSION OF OUTPUT
          TIME N        RESETS MAX EXECUTION TIME TO N MINUTES
------------------------------------------------------------;

      SOURCEIN:  BEGIN LABEL SOURCEIN,EF,COPY,NEWL,SOURCE,RMOB,
FST;

INTOVR:=INER;
IF FIRSTOFF THEN BEGIN FILECONTROL(5,"XBWKFL ",TIME(-1),FST);
      FST:     IF IU=0 THEN WRITE(TTY,INVIT);FIRSTOFF:=FALSE END;
IF LL=-1 THEN GO TO SOURCE;
      SOURCEIN:     IF IU GTR 0 THEN GO TO SOURCE;PINB:=PINB+80;
LL:=LL+1;IF LL LSS 24 THEN GO TO NEWL;
      SOURCE:     LL:=0;IF IU GTR 0 THEN READ(FN[IU-1][STOP],10,IOB[*])
ELSE READ(VDU[STOP],240,IOB[*]);
IF IU=2 THEN WRITE(TTY,10,IOB[*]);
PINB:=POINTER(IOB[*]);IOBE[1]:=0;
      NEWL:  C:=IF MS>0 THEN SSEQ[MS] ELSE 0;APR:=PINB;BPR:=PIBE;
A:=NMBR(6);IF A=0 THEN BEGIN IF CHA="%" THEN GO TO SOURCEIN;
COMMENT   NONVOID INPUT WITH ZERO STATEMENT NO. MUST BE COMMAND;
IOBE[1]:=0;REPLACE BPR-5 BY APR:CPR:=APR-1 FOR 1;
REPLACE BPR-4 BY APR:APR FOR 5 WHILE IN ALPHA;CHA:=IOBE[1];
ON("RUN000") BEGIN IF OBJECT THEN BEGIN IF NMBR(72) NEQ 0 OR
                        CHA NEQ "%" THEN GO PER;GO EXECUTE END
ELSE IF IU=2 THEN GO COMPILE
                        ELSE BEGIN WRITE(TTY,WT);
                  FILECONTROL(4,"XBWKFL ",TIME(-1),COMPILE) END END;
```

```
ON("DELETE" OR CHA="SCR000") BEGIN C:=NMBR(72);                          00056900
          IF C NEQ 0 OR CHA NEQ "%" THEN GO PER;WRITE(TTY,F7);           00057000
          CS:=MS:=IO[0]:=0;DANGER:=OBJECT:=FALSE;GO SOURCEIN END;        00057100
ON("LIST00") BEGIN                                                      00057200
                                                                        00057300
COMMENT   PROCESS LIST COMMAND;                                         00057400
                                                                        00057500
LABEL NEX,LEX;                                                          00057600
IF OU=1 THEN WRITE(TTY,DVO);                                            00057700
WRITE(FL[OU],SPC);                                                      00057800
      NEX:    BEG:=NMBR(72);IF CHCONV(CHA) NEQ 0 THEN GO PER;           00057900
EN:=ON("%" AND BEG=0) 1000000 ELSE ON(44) NMBR(72) ELSE BEG;            00058000
IF CHCONV(CHA) NEQ 0 THEN GO PER;                                       00058100
FOR A:=1 STEP 1 UNTIL MS DO IF SSEQ[A] LEQ EN                           00058200
AND SSEQ[A] GEQ BEG THEN BEGIN                                          00058300
REPLACE POINTER(IOBE[1]) BY " " FOR 112;                                00058400
WRITE(IOBE[*],NUM[2×OU],SSEQ[A]);                                       00058500
SCAN CPR:POINTER(IOBE[1])+20×OU FOR 20 WHILE NEQ 48;                    00058600
REPLACE CPR BY POINTER(PROG[A,2]) FOR 72;WRITE(FL[OU],14,IOBE[*])       00058700
END;ON(58) GO TO NEX;WRITE(TTY,STP,"LIST  ");GO TO SOURCEIN END;        00058800
ON("RENAME") FILECONTROL(0," NAMED"," ",SOURCEIN);                      00058900
ON("MAKE00") FILECONTROL(0,"CREATE","D",SOURCEIN);                      00059000
ON("SAVE00") FILECONTROL(1," SAVED"," ",SOURCEIN);                      00059100
                                                                        00059200
ON("LOAD00") FILECONTROL(2," LOADE","D",SOURCEIN);                      00059300
ON("REMOVE") FILECONTROL(3,"REMOVE","D",SOURCEIN);                      00059400
ON("COPY00") FILECONTROL(2," COPIE",0,SOURCEIN);                        00059500
ON("BYE000" OR CHA="HELL00") BEGIN IF IU=2 THEN GO FINSH;               00059600
      FILECONTROL(6,"XBWKFL ",TIME(-1),FINSH) END;                      00059700
ON("SEND00") BEGIN OU:=1;IF NOT HDDR THEN BEGIN                         00059800
WRITE(LIN,HD3,TIME(6),TIME(5).[23:12],TIME(5).[35:12],TIME(5).[11:12],  00059900
TIME(-1).[41:18],TIME(-1).[23:24]);                                     00060000
HDDR:=TRUE END;WRITE(TTY,HD1);GO SOURCEIN END;                          00060100
ON("NOSEND") BEGIN OU:=0;WRITE(TTY,STP,"NOSEND");GO TO SOURCEIN END;    00060200
ON("TTY000") BEGIN IU:=1;WRITE(TTY,STP,"SETTTY");GO TO SOURCE END;      00060300
ON("VDU000") BEGIN IU:=0;WRITE(TTY,STP,"SETVDU");GO TO SOURCEIN END;    00060400
ON("TIME00") BEGIN TIM:=3600×NMBR(10);WRITE(TTY,STP,"SETTIM");          00060500
GO TO SOURCEIN END;                                                     00060600
ON("PLOP00") FILECONTROL(5,0,0,SOURCEIN);                               00060700
ON("WHATS0") BEGIN WRITE(TTY,F9);IF IO[0]=0 THEN                        00060800
    WRITE(TTY,F10,MS,SSEQ[MS]) ELSE                                     00060900
    WRITE(TTY,F11,IO[0].[41:36],MS,SSEQ[MS]);                           00061000
    GO SOURCEIN END;                                                    00061100
                    % ILLEGAL COMMAND                                   00061200
WRITE(TTY,WHT,"      "&CHA[35:35:6×DELTA(CPR,APR)]);GO SOURCEIN END;    00061300
                                                                        00061400
COMMENT    PROCESS SOURCE STATEMENT;                                    00061500
                                                                        00061600
OBJECT:=FALSE;DANGER:=TRUE;                                             00061700
                                                                        00061800
COMMENT      DELETE STATEMENT;                                          00061900
                                                                        00062000
ON("%") BEGIN CHA:=A;FOR A:=1 STEP 1 UNTIL MS DO                        00062100
ON(SSEQ[A]) BEGIN MS:=MS-1;FOR B:=A STEP 1 UNTIL MS DO                  00062200
BEGIN SSEQ[B]:=SSEQ[B+1];WRITE(PROG[B,*],10,PROG[B+1,*]) END END;        00062300
GO TO SOURCEIN END;                                                     00062400
CHA:=A;APR:=APR-1;                                                      00062500
```

```
COMMENT    ADD NEW LAST STATEMENT;

IF CHA GTR C THEN BEGIN CS:=MS;MS:=A:=MS+1;
IF MS GTR 200 THEN GO TOOLONG;
GO TO COPY END;

COMMENT    REPLACE EARLIER STATEMENT;
FOR A:=1 STEP 1 UNTIL MS DO ON(SSEQ[A]) GO TO COPY
ELSE IF CHA LSS SSEQ[A] THEN BEGIN MS:=MS+1;

COMMENT       INSERT STATEMENT;

IF MS GTR 200 THEN GO TO TOOLONG;
FOR B:=MS STEP -1 UNTIL A+1 DO
BEGIN SSEQ[B]:=SSEQ[B-1];WRITE(PROG[B,*],10,PROG[B-1,*]) END;
GO TO COPY END;
  COPY:    PROG[A,11]:=SSEQ[A]:=CHA;
REPLACE BPR:CPR:=POINTER(PROG[A,2]) BY " " FOR 1;
B:=DELTA(APR,PINB+80);IF B>71 THEN B:=71;
REPLACE BPR:BPR BY APR FOR B WHILE NEQ "≠";
REPLACE BPR BY " " FOR 72-DELTA(CPR,BPR);GO SOURCEIN END SOURCEIN;
      INER:    WRITE(TTY,INTR);GO TO SOURCEIN;
      TOOLONG:    WRITE(TTY,LNGPRG,SSEQ[CS]);GO TO SOURCEIN;
    PER:    WRITE(TTY,F12);GO SOURCEIN;
COMMENT-----------------------------------------------------------
------------------- END SOURCEIN      --------------------------
-----------------------------------------------------------------
-----------     COMPILE:       SEARCH FOR SYNTAX ERRORS    ---------
-----------                    AND MAKE PSEUDO-OBJECT CODE ---------
-------------------------------------------------------------------;
COMMENT
                SYNTAX ERROR MESSAGES:  OUTPUT AFTER "RUN"
                                        WITH NEW FAULTY PROGRAM

         ARITH          MISSING OPERATOR OR OPERAND IN ARITHMETIC
                        EXPRESSION          (SHOULD NOT OCCUR)
         FILES          PROGRAM CAN HAVE ONLY ONE INPUT AND ONE OUTPUT
                        FILE
         IL GOSB        THIS STATEMENT HAS BEEN ILLEGALLY REFERENCED
                        BY A GOSUB STATEMENT (IT IS IN A FOR LOOP)
         IL RELN        AN ILLEGAL RELATION OF FORM ×?? HAS BEEN
                        FOUND IN AN IF STATEMENT
         ILL ARR        ARRAY NAME EXPECTED BUT CHARACTER IS NOT
                        A LETTER
         ILL ASN        AN ASSIGNMENT IS ATTEMPTED BUT LEFT HAND SIDE
                        IS NOT A VARIABLE
         ILL FN         THE NAME OF A DEFINED FUNCTION MUST BE OF FORM
                        FN LETTER.  PARAMETER(S) MUST BE SUPPLIED.
         ILL FOR        A FOR STATEMENT IS ALREADY IN OPERATION
                        FOR THIS VARIABLE
         ILL NEX        NEXT MUST REFER TO AN UNSUBSCRIPTED REAL VBLE
         ILL NUM        A PRIMARY IS MISSING OR ILLEGAL
         ILL STR        A STRING PRIMARY HAS BEEN ENCOUNTERED IN
                        A REAL EXPRESSION
         IL STMT        ILLEGAL STATEMENT
         INV IF         STRINGS CAN ONLY BE COMPARED FOR EQUALITY
```

```
                    OR INEQUALITY                                          00068300
INV PAR             A FORMAL PARAMETER IN A DEF STATEMENT                  00068400
                    MUST BE A VARIABLE                                     00068500
INV VAR             A PRIMARY IS MISSING OR ILLEGAL.  IN A READ            00068600
                    STATEMENT EVERY EXPRESSION MUST CONSIST OF             00068700
                    A SINGLE VARIABLE PRIMARY.                             00068800
IN+OUT?             A FILE IS BEING USED FOR INPUT AND OUTPUT              00068900
LONGSTR             A STRING CONTAINS MORE THAN 14 CHARACTERS              00069000
                    IN A STATEMENT OTHER THAN PRINT.                       00069100
MISG OP             AN OPERAND ESSENTIAL TO THIS STATEMENT HAS             00069200
                    BEEN OMITTED (END OF STATEMENT ERROR).                 00069300
MISP =              MISPLACED OR MISSING = IN DEF STATEMENT                00069400
MISPL -             A - SIGN HAS BEEN PLACED ILLEGALLY IN AN               00069500
                    EXPRESSION (E.G.  A*-B).                               00069600
NAMF                PROBABLY CAUSED BY ILLEGAL FILENAME                    00069700
NESTING             INCORRECTLY NESTED FOR AND NEXT STATEMENTS             00069800
NO ) X              MISSING PARENTHESES: X=A IN ARITH EXPRESSION           00069900
NO ( X                                   P IN FUNCTION PARAMETER           00070000
                                         S IN SUBSCRIPT                    00070100
                                         F IN FILE DECLARATION             00070200
NO EXPN             E HAS BEEN FOUND IN A NUMBER BUT NO EXPONENT           00070300
                    FOLLOWS                                                00070400
NO FOR              A NEXT STATEMENT HAS NO CORRESPONDING FOR              00070500
NOINPUT             INPUT STATEMENT MUST HAVE LIST OF VARAIBLES            00070600
NO NEXT             A FOR STATEMENT EARLIER IN PROGRAM HAS NO NEXT         00070700
NO PARM             EVERY FUNCTION MUST HAVE PARAMETER(S) IN               00070800
                    BRACKETS                                               00070900
NO PROG             THERE IS NO PROGRAM TO RUN                             00071000
NO RELN             NO RELATION HAS BEEN FOUND IN AN IF STATEMENT          00071100
NO SEPR             CONSECUTIVE ARITH EXPRESSIONS IN PRINT                 00071200
                    STATEMENT MUST BE SEPARATED BY , OR SEMICOLON          00071300
NO TO               A FOR STATEMENT MUST HAVE A FINAL VALUE                00071400
                    (FOR X=1 TO 10 ETC.)                                   00071500
NOT END             THE LAST STATEMENT MUST BE AN END STATEMENT            00071600
NOTLAST             THE END STATEMENT MUST BE THE LAST STATEMENT           00071700
OVERFLW             A NUMBER IS TOO LARGE                                  00071800
QUOTES              MISMATCHED STRING QUOTES                               00071900
REDC AR             ARRAY TWICE DIMENSIONED                                00072000
REDC FN             A FUNCTION HAS BEEN DEFINED MORE THAN ONCE             00072100
SAMEFIL             A FILE HAS BEEN DECLARED MORE THAN ONCE                00072200
PARAMTR             WRONG NUMBER OF PARAMETERS IN A FUNCTION CALL          00072300
SIMPLFY             AN EXPRESSION IS NESTED TO A DEPTH OF 10 OR            00072400
                    MORE AND SHOULD BE BROKEN UP                           00072500
SIZE                AN ARRAY DIMENSION IS GREATER THAN 64 OR               00072600
                    (IN MAT STATEMENT) IS GREATER THAN THE DECLARED        00072700
                    DIMENSION OF THE ARRAY                                 00072800
STORAGE             ARRAY STORAGE HAS BEEN EXCEEDED (700 WORDS OF          00072900
                    TYPE REAL, 70 OF TYPE ALPHA)                           00073000
STR >63             ILLEGAL OBJECT CHARACTER (E.G. PRINT STRING            00073100
                    HAS >63 CHARS OR STARTS LATER THAN COL 63)             00073200
STR=STR             A STRING CAN ONLY BE COMPARED WITH ANOTHER              00073300
SUBSCPT             AN ARRAY REFERENCE HAS THE WRONG NUMBER OF             00073400
                    SUBSCRIPTS                                             00073500
TOO MCH             PROCESSING OF THE SOURCE STATEMENT HAS NOT             00073600
                    USED UP ALL THE INFORMATION IN IT, (CAN BE             00073700
                    CAUSED BY OMISSION OF AN OPERATOR IN AN                00073800
                    EXPRESSION)                                            00073900
```

```
          TYPE            AN ATTEMPT HAS BEEN MADE TO USE A MAT STATEMENT 00074000
                          FOR A 1-DIMENSIONAL ARRAY                       00074100
          UNDC AR         AN ARRAY HAS BEEN REFERENCED BUT NOT DECLARED   00074200
          UNDC FL         A FILE HAS BEEN REFERENCED BUT NOT DECLARED     00074300
          UNDC FN         A FUNCTION OF TYPE FN* HAS BEEN REFERENCED BUT  00074400
                          NOT DECLARED                                    00074500
          UNDF GO         THERE IS NO STATEMENT IN THE PROGRAM WITH THE   00074600
                          SEQUENCE NUMBER REFERENCED BY THIS STATEMENT     00074700
          UNDSTAR         A STRING ARRAY HAS BEEN REFERENCED BUT NOT      00074800
                          DECLARED                                        00074900
          UNRC FN         A FUNCTION NAME OR KEYWORD HAS BEEN MISSPELLED  00075000
                          OR MISPLACED                                    00075100
          3 FILES         NOT MORE THAN TWO FILES MAY BE DECLARED         00075200
          11 FORS         FOR STATEMENTS NESTED TOO DEEP                  00075300
     ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::; 00075400
                                                                          00075500
          COMPILE:                                                        00075600
                                                                          00075700
     COMMENT         IN THE FOLLOWING COMMENTS,                           00075800
                        <E> IS ANY EXPRESSION (POSSIBLY WITH =)           00075900
                        <L> IS ANY LETTER                                 00076000
                        <V> IS ANY VARIABLE PRIMARY                       00076100
                        <N> IS ANY UNSIGNED INTEGER                       00076200
                        <S> IS A STATEMENT NUMBER                         00076300
                        <F> IS A FILENAME                                 00076350
                                                                          00076400
                     ALLOWED VARIANTS ARE LISTED BELOW. THREE LETTER      00076500
          ABBREVIATIONS OF INITIAL KEYWORDS ARE ALLOWED WHERE UNAMBIGUOUS.00076600
          SPACES ARE IGNORED EXCEPT INSIDE STRING QUOTES;                 00076700
                                                                          00076800
     INTOVR:=INTVR;INDEX:=QUOTE;FLAG:=FLAGR;                              00076900
     MSTO:=287;MSTR:=27;INFILTOG:=OUTFILTOG:=FALSE;                       00077000
     FOR A:=1 STEP 1 UNTIL 26 DO STRAR[A,1]:=ARR[A,1]:=ARR[A,2]:=SUB[A]:=0; 00077100
     IO[1]:=IO[2]:=FNM[2,1]:=FNM[2,2]:=FNM[3,1]:=FNM[3,2]:=AREASIZE:=0;   00077200
     NDEP:=CO:=1;POB:=POINTER(OBJ[*]);                                    00077300
     CS:=0;SY:=EQOK:=TRUE;                                                00077400
     IF MS =0 THEN SYNT("NO PROG");                                       00077500
     FORE:=FORC:=NCON:=0;FOR A:=1 STEP 1 UNTIL 200 DO STYP[A]:=0;         00077600
                                                                          00077700
     COMMENT                                                 FILES        00077800
                     <N>  FILES <D>,..                                    00077900
             WHERE <D> IS  <F>       (EXISTING FILE)                      00078000
                       OR  <F>(<N>)  (FILE TO BE CREATED: N=MAX NO OF RECS) 00078100
             WHERE <F> IS A CANDE FILENAME                                00078200
     THERE CAN ONLY BE ONE FILES STATEMENT. IT MUST BE THE FIRST STATEMENT. 00078300
     THERE CAN BE AT MOST ONE INPUT FILE AND ONE OUTPUT FILE;             00078400
                                                                          00078500
             FIL:    ACS:=CS:=CS+1;CP:=CHA:=0;CHA:=NWC;B:=LOOK(3);        00078600
     IF B="REM" THEN GO FIL;IF B="FIL" THEN BEGIN CP:=LP;SKIP(2,"ES");A:=0; 00078700
         RFIL:    A:=A+1;IF A>3 THEN SYNT("3 FILES");                     00078800
     CHA:=NWC;IO[A]:="       ";BPR:=POINTER(PROG[CS,2])+CP-1;             00078900
     SCAN APR:BPR FOR 6 WHILE IN ALPHA;FNM[2,A]:=B:=DELTA(BPR,APR);       00079000
     REPLACE POINTER(IO[A])+1 BY BPR FOR B;FNM[1,A]:=LOOK(B);CP:=LP;      00079100
     NF:=A;CHA:=NWC;ON("(") BEGIN AREASIZE:=NUMB/20+1;FNM[3,A]:=7;        00079200
     IF CHA NEQ ")" THEN SYNT("NO ) ,F");CHA:=NWC END;ON(",") GO TO RFIL; 00079300
     IF A=2 AND IO[1]=IO[2] THEN SYNT("SAMEFIL");IF CHA NEQ "%" THEN      00079400
     SYNT("TOO MCH") END ELSE CS:=0;                                      00079500
```

```
IF ACS>MS THEN SYNT("NO PROG");
COMMENT                                                        DIM
                     <N>  DIM <M>(<S>),...
                     <N>  DIMENSION <M>(<T>),...
         WHERE <M> IS <L> OR <L>$
                 <T> IS <N> OR <N>,<N>
   DIM STATEMENTS MUST PRECEDE ALL EXECUTABLE STATEMENTS
   STRING ARRAY MUST BE ONE-DIMENSIONAL.
   ALL DIMENSIONS MUST BE <64.
   713 WORDS REAL AND 73 STRINGS ARE AVAILABLE FOR ARRAYS        ;

       DIM:     ACS:=CS:=CS+1;CP:=CHA:=0;CHA:=NWC;
B:=LOOK(3);IF B="REM" THEN GO DIM;
IF B="DIM" THEN BEGIN CP:=LP;SKIP(6,"ENSION");
      RDIM:     K:=CHCONV(NWC);IF K=0 THEN SYNT("ILL ARR");
IF NWC="$" THEN BEGIN STRAR[K,0]:=MSTR;IF NWC NEQ "(" THEN
SYNT("NO ( ,S");A:=STRAR[K,1]:=NUMB;MSTR:=MSTR+A;
IF CHA NEQ ")" THEN SYNT("NO ) ,S");IF NWC="," THEN GO TO RDIM;
GO TO DIM END; ARR[K,0]:=MSTO;IF CHA NEQ "(" THEN SYNT("NO ( ,S");
IF ARR[K,1] NEQ 0 THEN SYNT("REDC AR");
B:=ARR[K,1]:=NUMB;IF B>64 THEN SYNT(" SIZE  ");
ON(",") BEGIN A:=ARR[K,2]:=NUMB;B:=B×A END;
MSTO:=MSTO+B;IF MSTO GTR 1000 THEN SYNT("STORAGE");
IF CHA NEQ ")" THEN SYNT("NO ) ,S");IF NWC="," THEN GO TO RDIM;
GO TO DIM END;IF ACS>MS THEN SYNT("NO PROG");

COMMENT          PROGRAM COMPILATION BEGINS HERE     ;

FOR CS:=ACS STEP 1 UNTIL MS DO BEGIN
COMMENT         IF GOSUB ENTRY THEN FIX FOR LEVEL     ;
      EXS:     IF STYP[CS] NEQ 0 THEN BEGIN IF FORC NEQ FORE THEN
SYNT("IL GOSB");FORE:=FORC:=STYP[CS] END;
COMMENT         IDENTIFY STATEMENT TYPE                        ;
NDEP:=CP:=CHA:=0;CHA:=NWC;B:=LOOK(3);EQOK:=TRUE;
FOR A:=1 STEP 1 UNTIL 18 DO IF B=KEY[A] THEN GO TO EQL;
IF B="DIM" OR B="FIL" THEN BEGIN WRITE(TTY,F6,SSEQ[CS]);
                                   GO SOURCEIN END;
%IF B="TRA" THEN BEGIN STYP[CS]:=20;GO REM END;
%IF B="UNT" THEN BEGIN STYP[CS]:=21;GO REM END;
B:=B DIV 64;
IF B="IF" THEN BEGIN LP:=LP-1;A:=19;GO TO EQL END;
IF B="ON" THEN BEGIN LP:=LP-1;A:=20;GO TO EQL END;
LP:=0;A:=1;
       EQL:     CP:=LP;STYP[CS]:=A;SPOB[CS]:=CO;
IF CS=MS AND A NEQ 18 THEN SYNT("NOT END");GO TO OPN[A];
COMMENT                                                 1  LET
                     <N>  LET  <E>
                     <N>  <E>
       WHERE E MAY BE A STRING ASSIGNMENT                           ;

       LET:   ARITH(0);GO TO INCST;
       CAR:    ARITH(1); GO TO INCST;
COMMENT                                                 20    ON
                     <N>  ON <E> GO TO <S>,...                         ;

       ONX:    ARITH(1);SKIP(3,"OTO");
       RON:    A:=NUMB;FOR B:=ACS STEP 1 UNTIL MS DO IF A=SSEQ[B]
```

```
THEN BEGIN PUT(B.[11:6]);PUT(B.[5:6]);
IF STYP[CS]=3 AND STYP[B]<FORC THEN STYP[B]:=FORC ELSE
COMMENT         SEE FOR STATEMENT FOR THIS TRICK;
IF STYP[CS]=20 THEN BEGIN ON(58) GO RON ELSE BEGIN PUT(0);PUT(0)
END END;GO INCST END;SYNT("UNDF GO");
COMMENT                                         19    IF
                <N>  IF <X><R><E> THEN <S>
                <N>  IF <X><R><E> GO TO <S>
              <X> IS AN EXPRESSION WITHOUT =
              <R> IS ONE OF THE FOLLOWING
                         ×EQ   ×GT   ×LT   ×GE   ×LE   ×NE
                          =     >     <     >=    <=    <>
        STRINGS CAN BE COMPARED ONLY FOR EQUALITY OR INEQUALITY;

      IFF:    EQOK:=FALSE;ARITH(0);EQOK:=TRUE;
C:=IF STRIN THEN 0 ELSE 1;
B:=0;IF CHA=30 THEN BEGIN B:=B+2;CHA:=NWC END;
IF CHA=14 THEN BEGIN B:=B+4;CHA:=NWC END;IF CHA=61 THEN B:=B+1
ELSE CP:=CP-1;IF B=0 THEN CHA:=NWC ELSE GO TO FEQ;
IF CHA NEQ "×" THEN SYNT("NO RELN");CHA:=NWC;
A:=CHA×64+NWC;FOR B:=1 STEP 1 UNTIL 6 DO IF A=KEY[B+32]
THEN GO TO FEQ;SYNT("IL RELN");
      FEQ:  PUT(B);ARITH(C);IF C=0 AND NOT STRIN THEN SYNT("STR=STR");
IF STRIN AND B NEQ 1 AND B NEQ 6 THEN SYNT("INV IF ");
SKIP(3,"OTO");SKIP(3,"HEN");GO TO RON;
COMMENT                                          2     GO TO
                <N>  GO TO <S>                                ;

     GOT:     SKIP(1,"O");GO TO RON;
COMMENT                                          3     GOSUB
                <N>  GOSUB <S>                               ;

      GOS:     SKIP(2,"UB");GO TO RON;
COMMENT                                          4    RETURN
                <N>  RETURN                                  ;

      RET:     SKIP(3,"URN");CHA:=NWC;GO TO INCST;
COMMENT                                          8    FOR
                <N>  FOR <E> TO <E> STEP <E>
                <N>  FOR <E> TO <E>                          ;

COMMENT         DURING COMPILE, FOR INFO IS STORED IN FORX AS FOLLOWS:
                        2                        3        4
            OBJPOINTER TO STORE NEXTLINE      FORLINE     ADDR

     EACH FOR IS ASSIGNED A LEVEL (=FORC), WHICH IS USED TO CHECK
     NESTING AND DURING EXECUTION.  IN SUBROUTINES FORC IS INITIALISED
     TO MAX FORC OF CALLING (GOSUB) STATEMENT: THIS LEVEL IS STORED
     TEMPORARILY IN STYP. OBJECT FORM OF FOR STATEMENT IS AS FOLLOWS:
            LEVEL  ASSIGN  FINAL  INCREMENT  NEXTLINE            ;

      FOX:     FORC:=FORC+1;PUT(FORC);IF FORC>10 THEN SYNT("11 FORS");
ARITH(1);FORX[FORC,4]:=ADDR+1;IF CHA NEQ "T" THEN SYNT("  NO TO");
SKIP(1,"O");FOR A:=1 STEP 1 UNTIL FORC-1 DO IF FORX[A,4]=ADDR THEN
SYNT("ILL FOR");ARITH(1);IF CHA="S" THEN BEGIN SKIP(3,"TEP");ARITH(1)
END ELSE PUT(0);FORX[FORC,2]:=CO;PUT(0);PUT(0);FORX[FORC,3]:=CS;
GO INCST;
```

```
COMMENT                                        9   NEXT
                <N>  NEXT <V>
        WHERE <V> IS A VARIABLE NAME                             ;

      NEX:   SKIP(1,"T");B:=CHCONV(NWC)-1;IF B=-1 THEN SYNT("ILL NEX");
IF NWC LSS 10 THEN BEGIN B:=11×B+CHA+1;CHA:=NWC END ELSE B:=11×B;
IF FORC=FORE THEN SYNT("NO FOR ");
IF B+1 NEQ FORX[FORC,4] THEN SYNT("NESTING");
A:=CO;CO:=FORX[FORC,2];PUT(CS,[11:6]);PUT(CS,[5:6]);
CO:=A;SPOB[CS]:=FORC+16×FORX[FORC,3];FORX[FORC,4]:=0;FORC:=FORC-1;
GO INCST;
COMMENT                                        11 DEF
                <N>  DEF FN<L>(<L>,...)=<E>
        FORMAL PARAMETERS MUST BE SINGLE LETTERS             ;

       DEF:    SKIP(3,"INE");SKIP(2,"FN");K:=CHCONV(NWC);
IF K=0 OR NWC NEQ 29 THEN SYNT(" ILL FN");A:=0;PUT(K);
IF SUB[K] NEQ 0 THEN SYNT("REDC FN");
     RDUM:  A:=A+1;B:=11×CHCONV(NWC)-11;IF B=-11 THEN SYNT("INV PAR");
CONST[NCON+A]:=B;IF NWC=58 THEN GO RDUM;IF CHA NEQ 45 THEN SYNT
(" NO ) P");SUB[K]:=A;FOR B:=A STEP -1 UNTIL 1 DO BEGIN
PUT((C:=CONST[NCON+B]).[11:6]);PUT(C.[5:6]) END;
IF NWC NEQ "=" THEN SYNT("MISP = ");ARITH(1);SUB[K]:=A;GO TO INCST;
COMMENT                                        6    READ
                <N>  READ <V>,...                                ;

       REA:    SKIP(1,"D");
       RREA:   ARITH(0);IF NOT VAR THEN SYNT("INV VAR");
ON(58) GO TO RREA;PUT(0);GO TO INCST;
COMMENT                                        5   INPUT
                <N>  INPUT <V>,...
                <N>  INPUT FILE <F>,<V>,...
        WHERE <F> IS THE FILENAME;
       INP:    SKIP(2,"UT");CFN;GO TO RREA;
COMMENT                                        13  RANDOMISE
                <N>  RANDOMISE
                <N>  RANDOMIZE                                   ;
       RAN:    SKIP(6,"DOMISE");SKIP(6,"DOMIZE");CHA:=NWC;GO TO INCST;
COMMENT                                        15  RESTORE
                <N>  RESTORE
                <N>  RESTORE FILE <F>
            WHERE <F> IS THE INPUT FILE                          ;
       RES:    SKIP(4,"TORE");CFN; GO TO INCST;
COMMENT                                        7   PRINT
                <N>  PRINT <P>
                <N>  PRINT FILE <F>,<Q>,...
        WHERE <F> IS A FILENAME
        WHERE <P> IS A NUMBER OF ELEMENTS OF FORM
                        "<Z>"                        4
                        <E>,                         2 OR 6, 1
                        <E>;                 COMMENT 2 OR 6
                         ,                           1 (1,5 IF TRAILS)
                         ;                   COMMENT 5 IF TRAILS
                      TAB(<E>)                       3
        WHERE <Z> IS ANY STRING
        WHERE <Q> IS    "<Z>"     OR    <E>                  ;
```

```
      PRI:    SKIP(2,"NT");CFN;CHA:=NWC;AA:=FALSE;
      RPRI:    ON("%") BEGIN PUT(0);GO TO INCST END;
ON(58) BEGIN PUT(1);AA:=FALSE;
IF NWC="%" THEN BEGIN PUT(5); GO TO INCST END
ELSE GO TO RPRI END;
ON(";" OR CHA=":") BEGIN AA:=FALSE;IF NWC="%" THEN BEGIN PUT(5);
GO INCST END;GO TO RPRI END;
ON(63) BEGIN AA:=FALSE;PUT(4);PUT(CP);CPR:=POINTER(PROG[CS,2])+CP;
SCAN APR:CPR UNTIL=63;B:=DELTA(CPR,APR);PUT(B);
CP:=CP+B+1;CHA:=NWC;GO TO RPRI END;
IF LOOK(4)="TAB(" THEN BEGIN AA:=FALSE;PUT(3);CP:=LP;ARITH(1);
IF CHA NEQ 45 THEN SYNT("NO )  P");CHA:=NWC;GO TO RPRI END;
CP:=CP-1;IF AA THEN SYNT("NO SEPR");AA:=TRUE;
B:=CO;PUT(2);ARITH(0);IF STRIN THEN
REPLACE POB+B BY "6" FOR 1;GO TO RPRI;
COMMENT                                              14  PAGE
               <N>  PAGE                                         ;
    PAG:     SKIP(1,"E");CHA:=NWC;GO TO INCST;
COMMENT                                              10  MAT
               <N>  MAT READ <L>                                 1
               <N>  MAT READ <L>(<N>,<N>)                        1
               <N>  MAT PRINT <L>                                2
               <N>  MAT PRINT <L>(<N>,<N>)                       2
               <N>  MAT LET <L>=(<E>)*<L>                        4
               <N>  MAT <L>=(<E>)*<L>                            4
               <N>  MAT LET <L>=<M>                              3
               <N>  MAT <L>=<M>                                  3
       WHERE <M> IS AS DESCRIBED IN EXECUTE(PROCEDURE MATOP)  ;
     MAT:     CHA:=NWC;A:=LOOK(4);IF A="READ" THEN BEGIN PUT(1);CP:=LP;
GO TO IOMT END;IF A="PRIN" THEN BEGIN PUT(2);CP:=LP;SKIP(1,"T");
GO TO IOMT END;CP:=CP-1;SKIP(3,"LET");A:=CP;B:=CHMAT(NWC);CHA:=NWC;
IF NWC=29 THEN BEGIN PUT(4);PUT(B);ARITH(1);CHA:=NWC;PUT(CHMAT(NWC));
END ELSE BEGIN PUT(3);PUT(A) END;GO TO REM;
     IOMT:     K:=CHMAT(NWC);PUT(K);IF NWC=29 THEN BEGIN
FOR B:=1,2 DO BEGIN A:=NUMB;IF A LSS 0 OR A GTR ARR[K,B] THEN
SYNT("   SIZE");PUT(A) END;CHA:=NWC;
END ELSE BEGIN PUT(ARR[K,1]);PUT(ARR[K,2]) END;
PUT(ON(";") 1 ELSE 0);ON(";") CHA:=NWC;GO TO INCST;
COMMENT                                              12 DATA
                       <N>  DATA <E>,...                         ;
      DAT:    SKIP(1,"A");
     RDAT:    ARITH(0);ON(58) GO TO RDAT;PUT(0);GO INCST;
COMMENT                                              17 STOP
               <N>  STOP
                                                     18  END
               <N>  END                                          ;
 ENX:    IF CS NEQ MS AND A=18 THEN SYNT("NOTLAST");CHA:=NWC;
IF A=17 THEN CHA:=NWC;
   INCST:   IF CHA NEQ "%" THEN SYNT("TOO MCH");
COMMENT                                              16  REM
               <N>  REM <Z>
      WHERE <Z> IS ANYTHING                                      ;
    REM:ERR:  END;

                                % SORT OUT FILES IF 2 TO BE USED
CS:=MS;IF NF=2 AND FNM[3,1]=FNM[3,2] THEN SYNT(" FILES ");
IF FNM[3,1]=7 THEN BEGIN IO[3]:=IO[1];IO[1]:=IO[2];
```

```
IO[2]:=IO[3] END;
IF NOT SY THEN BEGIN WRITE(TTY,STP,"ERRORS");GO TO SOURCEIN END;
OBJECT:=TRUE;GO TO EXECUTE;

   INTVR:    SYNT("OVERFLW");  QUOTE:   SYNT(" QUOTES");
       FLAGR: SYNT("NAME    ");


COMMENT---------------------------------------------------------------------
-------------              END    COMPILE              -------------------------
--------------------------------------------------------------------------------
-------------           EXECUTE:          EXECUTION OF      ---------------------
-------------                             USERS PROGRAM     ---------------------
-------------------------------------------------------------------------------;
      EXECUTE:     BEGIN

FILE IN          FIL1     DISK                   " "(2,10,300);
FILE OUT         FIL2     DISK[20:AREASIZE] " "(2,10,300,SAVE 7);

INTEGER ARRAY    SVE[1:10],                 % HOLDS GOSUB CALLS
                 FUNC[1:26],                %       DEFINES
                 STRGS[-1:100,0:2],         %       STRINGS
                 IOB[1:14],                 % I/O PSEUDOBUFFER
                 IOF[1:10],                 %       "
                 ADR[0:20];                 % ADDRESS STCK FOR EVAL

ARRAY            STORE[0:1000],             % HOLDS VARIABLE VALUES ETC
                 STK[0:20];                 % VALUE STACK FOR EVAL

INTEGER          XRND,             % PSEUDO-RANDOM NUMBER INDEX
                 CO,               % CURRENT POSITION IN OBJ[*]
                 RDAT,             %          DATA STATEMENT
                 NGOT,             % GO COUNTER
                 SLVE,             % GOSUB COUNTER
                 RDTP,             % POSITION IN DATA STATEMENT
                 MSTO,             % TOP OF STORE[*]
                 MSTR,             % TOP OF STRGS[*, ]
                 IR,               % INPUT FILE SEQUENCE NO
                 NR,               % OUTPUT FILE COUNTER
                 RT,               % RUN TERMINATION TIME
                 MF,               % FILE (0=TTY,OTHERWISE DISK)
                 STCK,             % STACK POINTER FOR EVAL
                 A,B,C,D,I,J,K,L,U,V,W,X,Y,Z,AS;

REAL             R,S,T;            % HASH

POINTER          PIOB,             % CURRENT POSITION IN IOB[*]
                 POUB,             % INITIAL
                 PBR,IPR;

LABEL INCST;

                          %  TRACE PACKAGE
%FORMAT T1(I6,X2,A3,X5,"VALUE ASSIGNED= "U),T2(I6,X2,A3,X5," TO STMT "
%    ,I6),T3(I6,X2,A3);
 BOOLEAN TRACEON,TLIN;%POINTER ITR;
%PROCEDURE DSTR(A);VALUE A;INTEGER A;
```

```
%BEGIN WRITE(IOF[*],T3,SSEQ[CS],IF TLIN THEN KEY[STYP[CS]] ELSE "   "); 00107800
%REPLACE ITR:POINTER(IOF[*])+14 BY "STRING ASSIGNED=";REPLACE ITR:ITR   00107900
%BY """ FOR 1;REPLACE ITR:ITR BY POINTER(STRGS[A,1]) FOR STRGS[A,0];    00108000
%REPLACE ITR BY """ FOR 1;WRITE(FL[OU],9,IOF[*]);TLIN:=FALSE END;       00108100
 DEFINE TRO(TRO1,TRO2)=#,%IF TRO1 THEN BEGIN WRITE(FL[OU],TRO2,SSEQ[CS],00108200
                    % IF TLIN THEN KEY[STYP[CS]] ELSE "   "#,          00108300
        TR1(TR11)=#, %=TRO(TRACEON,T1),TR11);TLIN:=FALSE END#,          00108400
        TR2(TR21)=#, %=TRO(TRACEON,T2),SSEQ[TR21]);TLIN:=FALSE END#,    00108500
        TR3      =#, %=TRO(TLIN,T3));TLIN:=FALSE END#,                  00108600
        TR4(TR41)=#; %=IF TRACEON THEN DSTR(TR41)#;                     00108700
                                                                        00108800
 %      PROCEDURES FOR EXECUTE:                                         00108900
                                                                        00109000
 COMMENT                                                                00109100
           ---      GET     GETS NEXT CHARACTER FROM OBJ          ;     00109200
                                                                        00109300
 DEFINE GET=0&OBJ[CO,[46:44]][5:47-CO,[2:3]x6:6];CO:=CO+1;              00109400
 IF CO,[2:3]=0 THEN CO:=CO+1#;                                          00109500
 COMMENT                                                                00109600
        ---      ERROR   DEALS WITH EXECUTION TIME ERRORS;              00109700
                                                                        00109800
  PROCEDURE ERROR(A);VALUE A;INTEGER A;                                 00109900
 BEGIN SWITCH FORMAT ERR:=("ERRO",I6),  % SHOULD NOT OCCUR..     %0     00110000
 ("SUBSCRIPT OUT OF BOUNDS AT LINE ",I6),                        %1     00110100
 ("LOG OF NEGATIVE OR ZERO NUMBER AT LINE ",I6),                 %2     00110200
 ("SQR OF NEGATIVE NUMBER AT LINE ",I6),                         %3     00110300
 ("UNDEFINED FUNCTION AT LINE ",I6),                             %4     00110400
 ("INPUT STATEMENT ATTEMPTED IN BATCH MODE AT LINE",I6),                00110500
 ("GO TO UNDEFINED STATEMENT NUMBER AT LINE ",I6),               %6     00110600
 ("RETURN WITHOUT GOSUB AT LINE ",I6),                           %7     00110700
 ("ARGUMENT FOR SIN,COS,TAN OR EXP EXCEEDS 158 AT LINE",I6),     %8     00110800
 ("INCREMENT UNDEFINED OR ZERO AT LINE ",I6),                    %9     00110900
 ("NEXT WITHOUT FOR AT LINE ",I6),                               %10    00111000
 ("STORAGE EXCEEDED AT LINE ",I6),                               %11    00111100
 ("INTEGER OVERFLOW AT LINE ",I6),                               %12    00111200
 ("INVALID ADDRESS AT LINE ",I6),                                %13    00111300
 ("DIVIDE BY ZERO AT LINE ",I6),                                 %14    00111400
 ("ILLEGAL EXPONENTIATION AT LINE ",I6),                         %15    00111500
 ("FLOATING-POINT OVERFLOW AT LINE ",I6),                        %16    00111600
 ("GOSUBS NESTED TOO DEEP (MORE THAN 10) AT LINE ",I6),          %17    00111700
 ("ILLEGAL EXPONENT ON INPUT AT LINE ",I6),                      %18    00111800
 ("MISPLACED STRING IN INPUT AT LINE ",I6),                      %19    00111900
 ("INPUT STRING TOO LONG AT LINE ",I6),                          %20    00112000
 ("OUT OF DATA AT LINE ",I6),("ERR22",I6), % ERR22 SHOULDNT OCCUR 21,22 00112100
 ("ILLEGAL MATRIX OPERATION AT LINE ",I6),                       %23    00112200
 ("INVERSE OF ILL-CONDITIONED MATRIX AT LINE ",I6),              %24    00112300
 ("INSUFFICIENT SPARE STORAGE FOR MAT OP AT LINE ",I6),          %25    00112400
 ("ILLEGAL FILE OPERATION AT LINE ",I6),                         %26    00112500
 ("INPUT FILE NOT ON DISK AT LINE",I6),                          %27    00112600
 ("INPUT FILE - INVALID USER AT LINE",I6),                       %28    00112700
 ("INPUT FILE IS NON-STANDARD AT LINE",I6),                      %29    00112800
 ("OUTPUT FILE - DUPLICATE NAME AT LINE",I6);                    %30    00112900
 COMMENT   LAST MESSAGE HERE IS NO. 30  ;                               00113000
 FORMAT DUR("THE FOLLOWING LINE WAS AWAITING OUTPUT:"),                 00113100
        FILAT(A6," FILE SEQUENCE NO.",I8);                              00113130
 WRITE(TTY,ERR[A],SSEQ[CS]);                                            00113150
 IF INFILTOG THEN WRITE(TTY,FILAT," INPUT",IR);                         00113170
```

```
IF OUTFILTOG THEN WRITE(TTY,FILAT,"OUTPUT",NR);                          00113200
IF DELTA(POUB,PIOB) GTR 0 THEN BEGIN WRITE(TTY,DUR);                     00113300
WRITE(TTY,9,IOB[*]) END;                                                 00113400
LOCK(FIL1);LOCK(FIL2);GO TO STOP END;                                    00113500
                                                                         00113600
COMMENT    ---      EVAL     EVALUATES ARITHMETIC EXPRESSION             00113700
                             (REVERSE POLISH DECODER)              ;     00113800
                                                                         00113900
REAL PROCEDURE EVAL;                                                     00114000
BEGIN                                                                    00114100
LABEL EQ,DONE,EXPON,RPT,SS,S,NUM,VR,AR,SF,UF,AD,SU,MU,DI,EX,FIN,         00114200
INM,STRGA,STRGC,STRGV;                                                   00114300
SWITCH TYP:=FIN,NUM,VR,AR,UF,SF,FIN,EQ,INM,AD,SU,MU,DI,EX,STRGV,         00114400
STRGA,STRGC;                                                             00114500
DEFINE TOP=STK[STCK];STCK:=STCK-1#;                                      00114600
DEFINE STACK(STACK1)=STCK:=STCK+1;STK[STCK]:=STACK1;GO TO SS#;           00114700
COMMENT            ADDR RETURNS RESULT ADDRESS (IN STORE IF REAL, STRGS  00114800
          IF STRING).  INTERMEDIATE RESULTS AND ADDRESSES ARE STACKED    00114900
          IN STK AND ADR RESECTIVELY.                              ;     00115000
STRIN:=FALSE;STCK:=0;                                                    00115100
      SS:   ADR[STCK]:=ADDR;CHA:=GET;GO TO TYP[CHA+1];                   00115200
      NUM:    A:=GET;A:=A×64+GET;STACK(CONST[A]);                        00115300
      STRGV:    ADDR:=GET;STRIN:=TRUE;STACK(0);                          00115400
      STRGA:    K:=GET;A:=TOP;IF A LEQ 0 OR A GTR STRAR[K,1]             00115500
THEN ERROR(1);ADDR:=STRAR[K,0]+A;STRIN:=TRUE;STACK(0);                   00115600
      STRGC:    A:=GET;K:=GET;REPLACE POINTER(STRGS[0,1])BY POINTER      00115700
(PROG[CS,2])+K FOR A;STRGS[0,0]:=A;STRIN:=TRUE;ADDR:=0;STACK(0);         00115800
      INM:    STACK(0);                                                  00115900
      VR:     K:=GET;K:=K-1;ADDR:=11×K+GET;STACK(STORE[ADDR]);           00116000
      AR:     K:=GET;A:=B:=TOP;IF ARR[K,2] NEQ 0 THEN BEGIN              00116100
A:=TOP;IF B LEQ 0 OR B GTR ARR[K,2] THEN ERROR(1) END;                   00116200
IF A LEQ 0 OR A GTR ARR[K,1] THEN ERROR(1);                              00116300
ADDR:=ARR[K,0]+(A-1)×ARR[K,2]+B-1;STACK(STORE[ADDR]);                    00116400
      SF:                                                                00116500
BEGIN LABEL SQR,SIF,COF,TAF,ATF,EXF,LOF,ABF,ENF,FNQ,DC,                  00116600
FIX,SGN,RND;                                                             00116700
SWITCH SFUN:=SIF,COF,TAF,ATF,EXF,LOF,ABF,ENF,SQR,FIX,SGN,RND;            00116800
DEFINE TEST=IF ABS(R)>158 THEN ERROR(8)#;                                00116900
A:=GET;R:=TOP;GO TO SFUN[A-18];                                          00117000
      SIF:    TEST;R:=SIN(R);GO TO DC;                                   00117100
      COF:    TEST;R:=COS(R); GO TO DC;                                  00117200
      TAF:    TEST;R:=SIN(R)/COS(R); GO TO DC;                           00117300
      ATF:    R:=ARCTAN(R); GO TO DC;                                    00117400
      EXF:    TEST;R:=EXP(R); GO TO DC;                                  00117500
      LOF:    IF R LEQ 0 THEN ERROR(2);R:=LN(R); GO TO DC;               00117600
      ABF:    R:=ABS(R); GO TO DC;                                       00117700
      SQR:    IF R LSS 0 THEN ERROR(3);R:=R*.5; GO TO DC;                00117800
      ENF:    R:=ENTIER(R);GO TO DC;                                     00117900
      FIX:    R:=ENTIER(R);IF R LSS 0 THEN R:=R+1;GO TO DC;              00118000
      SGN:    R:=IF R LSS 0 THEN -1 ELSE IF R GTR 0 THEN 1 ELSE 0;       00118100
GO TO DC;                                                                00118200
      RND:    XRND:=XRND×28^9;XRND:=XRND.[23:23];                        00118300
R:=XRND×2*(-23);GO TO DC;                                                00118400
      DC:   STACK(R) END;                                                00118500
COMMENT        USER FUNCTIONS SECTION  ;                                 00118600
      UF:   BEGIN INTEGER AS,SVSK,SVADDR;                                00118700
ARRAY DUM[1:20,1:4],SVSTK,SVADR[0:20];                                   00118800
```

```
                                                                        00118900
K:=GET;AS:=CS;CS:=FUNC[K];IF CS=0 THEN BEGIN CS:=AS;ERROR(4) END;       00119000
B:=CO;CO:=SPOB[CS];C:=GET;FOR A:=1 STEP 1 UNTIL SUB[K] DO BEGIN         00119100
COMMENT         SAVE VALUES OF FORMAL PARAMETERS AND STORE ACTUALS;     00119200
R:=TOP;C:=GET;C:=64×C+GET;DUM[A,1]:=C;DUM[A,2]:=STORE[C];STORE[C]:=R     00119300
END;SVADDR:=ADDR;SVSK:=STCK;                                            00119400
FOR A:=0 STEP 1 UNTIL 20 DO BEGIN SVSTK[A]:=STK[A];SVADR[A]:=ADR[A]     00119500
END;                                                                    00119600
COMMENT         NOW EVALUATE FUNCTION AND RESTORE FORMAL PARAMETERS;    00119700
R:=EVAL;FOR A:=1 STEP 1 UNTIL SUB[K] DO STORE[DUM[A,1]]:=DUM[A,2];      00119800
FOR A:=0 STEP 1 UNTIL 20 DO BEGIN STK[A]:=SVSTK[A];ADR[A]:=SVADR[A]     00119900
END;ADDR:=SVADDR;STCK:=SVSK;                                            00120000
CS:=AS;CO:=R;STACK(R) END;                                              00120100
     EQ:    IF STRIN THEN BEGIN ADDR:=A:=ADR[STCK];STCK:=STCK-1;        00120200
B:=ADR[STCK];REPLACE POINTER(STRGS[B,*]) BY POINTER(STRGS[A,*])         00120300
FOR 3 WORDS;TR4(B);GO TO SS END;R:=TOP;ADDR:=ADR[STCK];                 00120400
STORE[ADDR]:=STK[STCK]:=R;TR1(R);GO TO SS;                              00120500
     AD:    R:=TOP;R:=R+TOP;STACK(R);                                   00120600
     SU:    R:=TOP;R:=-R+TOP;STACK(R);                                  00120700
     MU:    R:=TOP;R:=R×TOP;STACK(R);                                   00120800
     DI:    R:=TOP;R:=1/R×TOP;STACK(R);                                 00120900
     EX:    T:=TOP;R:=TOP;IF T NEQ ENTIER(T) AND R LSS 0                00121000
THEN ERROR(15);STACK(R*T);                                              00121100
     FIN:   EVAL:=STK[1] END;                                           00121200
COMMENT                                                                 00121300
       ---      OUTP    OUTPUTS CONTENTS OF PSEUDO-BUFFER       ;       00121400
                                                                        00121500
PROCEDURE OUTP;                                                         00121600
BEGIN IF MF>0 THEN ERROR(26);TR3;                                       00121700
WRITE(FL[OU],14,IOB[*]);REPLACE PIOB:=POUB:=POINTER(IOB[*]) BY          00121800
" " FOR 112;IF OU=1 THEN PIOB:=POUB:=POUB+20 END;                       00121900
COMMENT                                                                 00122000
       ---      MORE    FALSE IF END OF STATEMENT               ;       00122100
                                                                        00122200
BOOLEAN PROCEDURE MORE;                                                 00122300
BEGIN INTEGER A,B;A:=CO;B:=GET;MORE:=B NEQ 0;CO:=A END;                  00122400
COMMENT                                                                 00122500
       ---      OUTNUM  PLACES NUMBER IN PSEUDO-BUFFER          ;       00122600
                                                                        00122700
PROCEDURE OUTNUM(A,F);VALUE A,F;REAL A;INTEGER F;                       00122800
BEGIN CHA:=0;CP:=CP-1;WRITE(IOBE[*],NUM[F],A); % CARE:                  00122900
SCAN APR:CPR FOR 16 UNTIL ="@";                                         00123200
IF DELTA(CPR,APR) NEQ 16 THEN REPLACE APR BY "E" FOR 1;                 00123300
IF DELTA(POUB,PIOB) GTR 60+F×4 THEN OUTP;                               00123400
REPLACE PIOB:PIOB BY POINTER(IOBE[1]) FOR 14 UNTIL=48;                  00123500
REPLACE PIOB:PIOB BY " " FOR 1 END;                                     00123600
COMMENT                                                                 00123700
       ---    QUO   PLACES " IN OUTPUT BUFFER          ;               00123800
DEFINE QUO=IF MF>0 THEN REPLACE PIOB:PIOB BY """#;                      00123900
COMMENT                                                                 00124000
       ---      MATOP   PROCESSES MOST MAT STATEMENTS           ;       00124100
                                                                        00124200
PROCEDURE MATOP;                                                        00124300
BEGIN INTEGER U,V,W,X,Y,Z,T,J;                                          00124400
COMMENT                 MAT STATEMENTS CONCERNED HAVE FORMAT            00124500
                <N>  MAT LET <L>=<M>                                    00124600
                <N>  MAT <L>=<M>                                        00124700
```

```
        THE SECOND CHARACTER IN <M> IS USED TO IDENTIFY ACTION TAKEN     00124800
        THIS CAN BE     *  +  -  E  0  D  R  N  %                        00124900
        (RECALL % IS END-STATEMENT CHARACTER)                            ;00125000
LABEL DONE,ADSU,EQM,CONS,EX,EY;REAL PIVOT,AI;                            00125100
INTEGER ARRAY IR[1:72];REAL ARRAY TEM[1:72];                             00125200
DEFINE AA(AA1,AA2)=STORE[ARR[K,0]+(AA1-1)×U+AA2-1]#;                     00125300
CP:=GET;CHA:=0;K:=CHCONV(NCH);IF K=0 THEN ERROR(23);                     00125400
IF NCH NEQ 61 THEN ERROR(23);A:=CHCONV(NCH);CHA:=NCH;                    00125500
COMMENT                  SWITCH OCCURS HERE                              00125600
                                        *  MATRIX MULTIPLICATION         00125700
                <M>  IS  <L>*<L>                                         00125800
      HASH STORAGE IS USED TO AVOID TROUBLE WITH A=A*B ETC               ;00125900
ON("*") BEGIN B:=CHCONV(NCH);                                            00126000
U:=ARR[K,1]-1;V:=ARR[K,2]-1;W:=ARR[A,1]-1;X:=ARR[A,2]-1;                 00126100
Y:=ARR[B,1]-1;Z:=ARR[B,2]-1;IF U NEQ W OR V NEQ Z OR X NEQ Y             00126200
THEN ERROR(23);IF MSTO+(U+1)×(V+1) GTR 1000 THEN ERROR(25);              00126300
FOR W:=0 STEP 1 UNTIL U DO FOR Z:=0 STEP 1 UNTIL V DO BEGIN R:=0;        00126400
FOR Y:=0 STEP 1 UNTIL X DO R:=R+STORE[ARR[A,0]+X×W+W+Y]×                 00126500
STORE[ARR[B,0]+Y×V+Y+Z];STORE[MSTO+W×V+W+Z]:=R END;                      00126600
FOR W:=0 STEP 1 UNTIL U DO FOR Z:=0 STEP 1 UNTIL V DO                    00126700
STORE[ARR[K,0]+W×V+W+Z]:=STORE[MSTO+W×V+W+Z];                            00126800
GO TO DONE END;                                                          00126900
COMMENT                                 +  MATRIX ADDITION               00127000
                <M>  IS  <L>+<L>                                         ;00127100
ON("+") BEGIN Z:=1;GO TO ADSU END;                                       00127200
COMMENT                                 -  MATRIX SUBTRACTION            00127300
                <M>  IS  <L>-<L>                                         ;00127400
ON("-") BEGIN Z:=-1;GO TO ADSU END;                                      00127500
COMMENT                                    0  ALL ONES                   00127600
                <M>  IS  CON                                             ;00127700
ON("0") BEGIN Z:=Y:=1;GO TO CONS END;                                    00127800
COMMENT                                    D  IDENTITY MATRIX            00127900
                <M>  IS  IDN                                             ;00128000
ON("D") BEGIN IF ARR[K,1] NEQ ARR[K,2] THEN ERROR(23);Z:=1;Y:=0;         00128100
COMMENT                                    E  ZERO MATRIX                00128200
                <M>  IS  ZER                                             ;00128300
GO TO CONS END;       ON("E") BEGIN Z:=Y:=0;GO TO CONS END;              00128400
COMMENT                                    R  TRANSPOSITION              00128500
                <M>  IS  TRN(<L>)                                        00128600
       HASH STORAGE USED TO AVOID TROUBLE WITH A=TRN(A)                  ;00128700
ON("R") BEGIN CHA:=NCH;CHA:=NCH;A:=CHCONV(NCH);IF A=0 THEN ERROR(23);     00128800
U:=ARR[K,1]-1;V:=ARR[K,2]-1;W:=ARR[A,1]-1;X:=ARR[A,2]-1;                 00128900
IF U NEQ X OR V NEQ W THEN ERROR(23);                                    00129000
IF MSTO+(U+1)×(V+1) GTR 1000 THEN ERROR(25);                             00129100
FOR U:=0 STEP 1 UNTIL X DO FOR V:=0 STEP 1 UNTIL W DO                    00129200
STORE[MSTO+U×W+U+V]:=STORE[ARR[A,0]+V×X+V+U];                            00129300
FOR U:=0 STEP 1 UNTIL X DO FOR V:=0 STEP 1 UNTIL W DO                    00129400
STORE[ARR[K,0]+U×W+U+V]:=STORE[MSTO+U×W+U+V];                            00129500
GO TO DONE END;                                                          00129600
COMMENT                                    N   INVERSION                 00129700
                <M>  IS  INV(<L>)                                        ;00129800
ON("N") BEGIN CHA:=NCH;CHA:=NCH;A:=CHCONV(NCH);                          00129900
U:=ARR[K,1];IF U NEQ ARR[K,2] OR U NEQ ARR[A,1] OR U NEQ ARR[A,2]        00130000
THEN ERROR(23);FOR I:=0 STEP 1 UNTIL U-1 DO FOR J:=0 STEP 1 UNTIL U-1    00130100
DO AA(I+1,J+1):=STORE[ARR[A,0]+I×U+J];                                   00130200
FOR V:=1 STEP 1 UNTIL U DO BEGIN PIVOT:=0;                               00130300
FOR I:=1 STEP 1 UNTIL U DO BEGIN IF V NEQ 1 THEN BEGIN                   00130400
```

```
FOR X:=1 STEP 1 UNTIL V-1 DO IF I=IR[X] THEN GO TO EX END;                    00130500
IF ABS(AA(I,V))GTR ABS(PIVOT) THEN BEGIN PIVOT:=AA(I,V);Y:=IR[V]:=I           00130600
END;                                                                          00130700
      EX:    END;IF ABS(PIVOT) LSS .0001 THEN ERROR(24);                      00130800
FOR J:=1 STEP 1 UNTIL U DO AA(Y,J):=AA(Y,J)/PIVOT;AA(Y,V):=1/PIVOT;           00130900
FOR I:=1 STEP 1 UNTIL U DO IF I NEQ Y THEN BEGIN AI:=AA(I,V);                 00131000
AA(I,V):=-AI/PIVOT;FOR J:=1 STEP 1 UNTIL U DO IF J NEQ V THEN                 00131100
AA(I,J):=AA(I,J)-AI×AA(Y,J) END END;                                          00131200
FOR I:=1 STEP 1 UNTIL U DO                                                    00131300
BEGIN FOR J:=1 STEP 1 UNTIL U DO TEM[J]:=AA(I,J);                             00131400
FOR J:=1 STEP 1 UNTIL U DO AA(I,IR[J]):=TEM[J] END;                           00131500
FOR J:=1 STEP 1 UNTIL U DO BEGIN FOR I:=1 STEP 1 UNTIL U DO                   00131600
TEM[I]:=AA(IR[I],J);FOR I:=1 STEP 1 UNTIL U DO AA(I,J):=TEM[I]                00131700
END;GO TO DONE END;                                                           00131800
                                                  %   EQUALITY                00131900
COMMENT          <M>  IS  <L>                                    ;            00132000
ON("%") BEGIN B:=A;Z:=0; GO TO EQM END;                                       00132100
ERROR(23);                                                                    00132200
      ADSU:    B:=CHCONV(NCH);IF B=0 THEN ERROR(23);                          00132300
IF ARR[A,1] NEQ ARR[B,1] OR ARR[A,2] NEQ ARR[B,2] THEN ERROR(23);             00132400
      EQM:     U:=ARR[K,1];V:=ARR[K,2];IF U NEQ ARR[A,1]                      00132500
OR V NEQ ARR[A,2] THEN ERROR(23);                                             00132600
FOR I:=0 STEP 1 UNTIL U-1 DO FOR J:=0 STEP 1 UNTIL V-1 DO                     00132700
AA(I+1,J+1):=STORE[ARR[A,0]+I×U+J]+Z×STORE[ARR[B,0]+I×U+J];                   00132800
GO TO DONE;                                                                   00132900
      CONS:   U:=ARR[K,1];FOR I:=1 STEP 1 UNTIL ARR[K,1] DO                   00133000
FOR J:=1 STEP 1 UNTIL ARR[K,2] DO AA(I,J):=IF I=J THEN Z ELSE Y;              00133100
      DONE:    GO TO INCST END;                                               00133200
 LABEL RPT,REM,DAT,EXS,LET,RLET,ONX,                                          00133300
INP,PRI,RPRI,XPRI,MAT,ENX,RREA,XREA,QDAT,FREA,RES,NDAT,                       00133400
STRV,FOL,INTVR,INDEXR,DZER,EXPVR,                                             00133500
NM,TAB,COM,STR,EPRI,OUD,OUF,                                                  00133600
IFF,GOT,GOX,GOS,RET,FOX,NEX,FD,DEF,REA,EREA,RAN,PAG;                          00133700
SWITCH OPN:=LET,GOT,GOS,RET,INP,REA,EPRI,FOX,NEX,MAT,                         00133800
DEF,DAT,RAN,PAG,RES,REM,ENX,ENX,IFF,ONX;                                      00133900
                                                                              00134000
SWITCH TYP:=XPRI,COM,NM,TAB,STR,INCST,STRV;                                   00134100
                                                                              00134200
COMMENT----------------------------------------------------------------------00134300
----------------        EXECUTE  BEGINS HERE       --------------------00134400
----------------------------------------------------------------------------;00134500
                                                                              00134600
INTOVR:=INTVR;EXPOVR:=EXPVR;ZERO:=DZER;INDEX:=INDEXR;                         00134700
WRITE(TTY,MESS);RT:=TIME(2)+TIM;IF OU=1 THEN WRITE(TTY,DVO);                  00134800
WRITE(FL[OU],SPC);NR:=IR:=RDAT:=RDTP:=NGOT:=0;TRACEON:=TLIN:=FALSE;           00134900
FORC:=SLVE:=0;XRND:=101;CS:=ACS-1;                                            00135000
POUB:=PIOB:=POINTER(IOB[*])+20×OU;                                            00135100
REPLACE POINTER(IOB[*]) BY " " FOR 112;                                       00135200
                                                                              00135300
                    % GET FILES IF NEEDED:                                    00135400
IF INFILTOG THEN BEGIN FILL FIL1 WITH IO[1],TIME(-1);                         00135500
SEARCH(FIL1,ANSA[*]);IF ANSA[0] LEQ 0 THEN ERROR(28+ANSA[0]);                 00135600
IF ANSA[3] NEQ 10 OR ANSA[4] NEQ 300 THEN ERROR(29) END;                      00135700
IF OUTFILTOG THEN BEGIN FILL FIL2 WITH IO[2],TIME(-1);                        00135800
SEARCH(FIL2,ANSA[*]);IF ANSA[0] NEQ -1 THEN BEGIN                             00135900
WRITE(TTY,F13,IO[2].[41:36]);                                                 00136000
IF IU=2 THEN U:=0 ELSE                                                        00136100
```

```
READ(TTY,REP,U);IF U NEQ "YES" THEN ERROR(30) END END;                    00136200
                                                                          00136300
COMMENT         RETURN TO HERE AFTER EACH STATEMENT;                      00136400
                                                                          00136500
      REM:DAT:INCST:     TR3;CS:=CS+1;                                    00136600
      EXS:               MF:=0;   % FIRST SEE IF EXCESS TIME              00136700
IF TIME(2) GTR RT THEN BEGIN WRITE(TTY,BK);GO TO ENX END;                 00136800
IF STYP[CS]>19 THEN BEGIN TRACEON:=STYP[CS]=20;GO INCST END;              00136900
IF TRACEON THEN TLIN:=TRUE;                                               00137000
                                                                          00137100
                                                                          00137200
U:=STYP[CS];CO:=SPOB[CS];GO TO OPN[U];  % NOW GO TO APPROPRIATE PLACE     00137300
                                        % LET STATEMENT                   00137400
     LET:    R:=EVAL;GO TO INCST;                                         00137500
                                        % ON STATEMENT                    00137600
      ONX:    U:=EVAL;                                                    00137700
FOR V:=1 STEP 1 UNTIL 2×U-2 DO BEGIN S:=GET END;GO GOT; %(STET)           00137800
                                         % IF STATEMENT                   00137900
      IFF:    R:=EVAL;IF STRIN THEN BEGIN                                 00138000
COMMENT                                STRING IF       ;                  00138100
U:=GET;                                                                   00138200
REPLACE IPR:=POINTER(STRGS[-1,*]) BY POINTER(STRGS[ADDR,*]) FOR 24;       00138300
R:=EVAL;GO IF IPR=POINTER(STRGS[ADDR,*])                                  00138400
FOR STRGS[-1,0]+8 EQV U=1 THEN GOT ELSE INCST END;                        00138500
COMMENT                               REAL IF       ;                   00138600
U:=GET;R:=R-EVAL;                                                         00138700
IF R GTR 0 AND U.[2:1]=1 THEN GO TO GOT ELSE                              00138800
IF R LSS 0 AND U.[1:1]=1 THEN GO TO GOT ELSE                              00138900
IF R = 0 AND U.[0:1]=1 THEN GO TO GOT;GO TO INCST;                        00139000
                                        % GOTO STATEMENT                  00139100
     GOT:    U:=GET;U:=64×U+GET;IF U=0 THEN ERROR(6);                     00139200
      GOX:    NGOT:=NGOT+1;TR2(U);                                        00139300
COMMENT                          MONITOR FOR EXCESS LOOPING;              00139400
IF NGOT=100 AND IU NEQ 2 THEN BEGIN WRITE(TTY,WRN);READ(TTY,REP,W);       00139500
IF W NEQ "YES" THEN GO TO STOP END;                                       00139600
CS:=U;GO EXS;                                                             00139700
                                        % GOSUB STATEMENT                 00139800
      GOS:    SLVE:=SLVE+1;IF SLVE GTR 10 THEN ERROR(17);                 00139900
SVE[SLVE]:=CS;GO GOT;                                                     00140000
                                        % RETURN STATEMENT                00140100
      RET:    IF SLVE=0 THEN ERROR(7);                                    00140200
CS:=SVE[SLVE];SLVE:=SLVE-1;GO TO INCST;                                   00140300
                                        % FOR STATEMENT                   00140400
      FOX:    FORC:=GET;R:=EVAL;                                          00140500
                                                                          00140600
COMMENT         FORX CONTROL INFO IS STORED AS FOLLOWS:                   00140700
                 1       2       3       4                                00140800
                ADDR    STEP    FINAL   FORLINE                          00140900
          A FOR LOOP IS EXECUTED ZERO TIMES IN THE RIGHT CIRCUMSTANCES;   00141000
                                                                          00141100
V:=FORX[FORC,1]:=ADDR;S:=FORX[FORC,3]:=EVAL;                              00141200
T:=FORX[FORC,2]:=IF MORE THEN EVAL ELSE 1;                                00141300
W:=FORX[FORC,4]:=CS;IF T=0 THEN ERROR(9);                                 00141400
IF T×R LEQ T×S THEN GO INCST;       % ELSE SKIP LOOP                     00141500
U:=GET;U:=GET;CS:=64×U+GET;GO TO INCST;                                   00141600
                                                                          00141700
                                        % NEXT STATEMENT                  00141800
```

```
                % SPOB STORES (NEXTS LEVEL IN FORX)+16×FORLINE                  00141900
      NEX:      U:=SPOB[CS].[3:4];V:=SPOB[CS].[41:38];                         00142000
IF V NEQ FORX[U,4] THEN ERROR(10);L:=FORX[U,1];                                00142100
T:=FORX[U,2];R:=STORE[L]+T;                                                    00142200
IF T×R LEQ T×FORX[U,3] THEN BEGIN STORE[L]:=R;TR1(R);T:=FORX[U,4];             00142300
                    TR2(T+1);CS:=T END ELSE FORX[U,4]:=0;                      00142400
GO TO INCST;                                                                   00142500
                                                    % DEFINE STATEMENT         00142600
      DEF:      U:=GET;FUNC[U]:=CS;GO TO INCST;                                00142700
                                                    % READ STATEMENT           00142800
      REA:      U:=0;                                                          00142900
COMMENT          THIS SECTION IS COMPLICATED BECAUSE OF SWITCHING              00143000
OF ATTENTION FROM READ STATEMENT TO DATA STATEMENT AND BACK ETC;                00143100
      RREA:     R:=EVAL;L:=ADDR;U:=CS;V:=CO;                                   00143200
IF RDTP=0 THEN GO TO QDAT;CO:=RDTP;CS:=RDAT;                                   00143300
     XREA:     IF STRIN THEN BEGIN R:=EVAL;CS:=U;IF NOT STRIN THEN             00143400
ERROR(20);REPLACE POINTER(STRGS[L,*])BY POINTER(STRGS[ADDR,*])                 00143500
FOR 3 WORDS;TR4(L) END ELSE BEGIN R:=EVAL;IF STRIN THEN ERROR(20);             00143600
STORE[L]:=R;CS:=U;TR1(R) END;RDTP:=IF MORE THEN CO ELSE 0;CO:=V;               00143700
IF MORE THEN GO TO RREA ELSE GO TO INCST;                                      00143800
COMMENT          FIND ANOTHER DATA STATEMENT;                                  00143900
      QDAT:      FOR CS:=RDAT+1 STEP 1 UNTIL MS DO BEGIN                       00144000
IF STYP[CS]=12 THEN GO TO FREA END;                                            00144100
CS:=U;ERROR(21);                                                               00144200
     FREA:    RDAT:=CS;CO:=SPOB[CS];GO TO XREA;                                00144300
                                                    % INPUT STATEMENT         00144400
                % "STOP" AT START OF INPUT STREAM STOPS A RUN                   00144500
      INP:      BEGIN LABEL RINP,EVINP,RPT,EXPON,DONE,FINP;                    00144600
MF:=GET;IF MF=0 THEN BEGIN IF IU=2 THEN ERROR(5);                              00144700
IF DELTA(POUB,PIOB) GTR 0 THEN BEGIN                                           00144800
REPLACE POINTER(IOBE[*]) BY POUB FOR 72;WRITE(TTY[STOP],9,IOBE[*]);            00144900
REPLACE PIOB:=POUB:=POINTER(IOB[*]) BY " " FOR 112;IF OU=1 THEN                00145000
PIOB:=POUB:=PIOB+20 END;READ(TTY,9,IOBE[*]);                                   00145100
READ(IOBE[*],REP,V);IF V="STO" THEN GO TO STOP END                             00145200
ELSE BEGIN Z:=GET;Z:=64×Z+GET;READ(FIL1,10,IOBE[*])[OUD];                      00145300
           READ(IOBE[*],SNUM,IR) END;                                          00145350
REPLACE POINTER(PROG[0,2]) BY POINTER(IOBE[1]) FOR 72;X:=0;                    00145400
      RINP:     R:=EVAL;L:=ADDR;U:=CS;CS:=0;CP:=X;CHA:=0;                      00145500
IF NCH="%" THEN BEGIN IF MF=0 THEN BEGIN WRITE(TTY,MNP[IF X=0 THEN 1           00145600
ELSE 0],PROG[U,1]);CS:=U;GO TO EXS END ELSE BEGIN                              00145700
READ(FIL1,9,IOBE[*])[OUD];                                                     00145800
REPLACE POINTER(PROG[0,2]) BY POINTER(IOBE[*]) FOR 72;CP:=1 END END;           00145900
COMMENT          INPUT STRING MAY OR MAY NOT HAVE ""      ;                    00146000
      EVINP:    CP:=CP-1;CHA:=NCH;IF STRIN THEN BEGIN                          00146100
ON(63) SCAN APR:APR:=CPR:=POINTER(PROG[0,2])+CP FOR 15 WHILE NEQ 63            00146200
ELSE BEGIN CP:=CP-2;SCAN APR:APR:=CPR:=POINTER(PROG[0,2])+CP+1 FOR 15          00146300
WHILE IN ALPHA END;                                                            00146400
V:=DELTA(CPR,APR);IF V>14 THEN BEGIN CS:=U;ERROR(20) END;                      00146500
STRGS[L,0]:=V;REPLACE POINTER(STRGS[L,1]) BY CPR FOR V;                        00146600
CP:=CP+V+1;CHA:=NCH;CS:=U;TR4(L);GO FINP END;                                  00146700
T:=R:=Y:=0;                                                                    00146800
COMMENT          INPUT NUMBER                                ;                 00146900
ON(44) BEGIN T:=1;CHA:=NCH END;                                                00147000
       RPT:     ON(26) BEGIN Y:=1;CHA:=NCH END;                                00147100
IF CHA GEQ 10 THEN GO TO EXPON;IF Y GTR 0 THEN                                 00147200
BEGIN R:=R+CHA×10*(-Y);                                                        00147300
Y:=Y+1 END ELSE R:=R×10 +CHA;CHA:=NCH; GO TO RPT;                              00147400
```

```
      EXPON:    IF CHA NEQ 21 THEN GO TO DONE;Y:=1;CHA:=NCH;Z:=0;
ON(44)BEGIN Y:=-1;CHA:=NCH END ELSE ON(16) CHA:=NCH;
IF CHA GEQ 10 THEN BEGIN CS:=U;ERROR(18) END;Z:=CHA;CHA:=NCH;
IF CHA LSS 10 THEN BEGIN Z:=Z×10+CHA;CHA:=NCH END;
R:=R×10*(Y×Z);
      DONE:     IF T=1 THEN R:=-R;DELIM:=72;STORE[L]:=R;CS:=U;TR1(R);
      FINP:     X:=CP;IF CHA NEQ 58 AND CHA NEQ "%" THEN ERROR(19);
IF MORE THEN GO TO RINP ELSE GO TO INCST END;
                                                % RANDOMISE STATEMENT
      RAN:     XRND:=(2×TIME(1)+1).[23:23];GO TO INCST;
                                                % RESTORE STATEMENT
      RES:     MF:=GET;IF MF=0 THEN RDTP:=RDAT:=0 ELSE REWIND(FIL1);
GO INCST;
                                                % PRINT STATEMENT
      EPRI:     MF:=GET;IF MF>0 THEN BEGIN PBR:=PIOB;
TR3;Z:=GET;Z:=64×Z+GET;
POUB:=POINTER(IOF[*1);REPLACE PIOB:=POINTER(IOF[*]) BY " " FOR 72 END;
      PRI:     CHA:=GET;IF MF>0 AND CHA=5 THEN ERROR(26);
GO TO TYP[CHA+1];
Z:=GET;Z:=64×Z+GET;
COMMENT          * IN PRINT MOVES TO NEXT 14-SPACE COLUMN.

                 (SEMICOLON IN MIDDLE OF PRINT IS JUST DELIMITER);
     COM:     IF MF>0 THEN REPLACE PIOB:PIOB BY "," ELSE BEGIN
V:=DELTA(POUB,PIOB);IF V GTR 56 THEN BEGIN
OUTP;V:=0 END ELSE V:=14-(V MOD 14);
FOR U:=1 STEP 1 UNTIL V DO REPLACE PIOB:PIOB BY " " END;
GO TO PRI;
COMMENT                  PLACE STRING IN PSEUDO-BUFFER    ;
      STR:     CP:=GET;
CPR:=POINTER(PROG[CS,2])+CP;
V:=72-DELTA(POUB,PIOB);W:=GET;
IF W GTR V THEN BEGIN REPLACE PIOB:PIOB BY CPR:CPR FOR V;
OUTP;W:=W-V END;
QUO;REPLACE PIOB:PIOB BY CPR:CPR FOR W;QUO;
GO TO PRI;

COMMENT                  TAB OVERWRITES ON TELETYPE AND LINE-PRINTER
                         BUT REPLACES ON VIDEO UNIT.                ;

      TAB:     IF MF>0 THEN ERROR(26);U:=EVAL-1;U:=U MOD 72;
IF IU+OU NEQ 0 THEN BEGIN TR3;WRITE(FL[OU][NO],9,IOB[*]);
REPLACE POUB BY " " FOR 72 END;PIOB:=POUB+U; GO TO PRI;
      STRV:  R:=EVAL;QUO;REPLACE PIOB:PIOB BY POINTER(STRGS[ADDR,1])
FOR STRGS[ADDR,0];QUO;GO TO PRI;
      NM:     OUTNUM(EVAL,0);GO TO PRI;
     XPRI:     IF MF>0 THEN BEGIN REPLACE PIOB BY ",";
NR:=NR+10;REPLACE POINTER(IOF[10]) BY NR FOR 8 DIGITS;
WRITE(FIL2,10,IOF[*])[OUF];PIOB:=PBR;POUB:=POINTER(IOB[*])+20×OU END
ELSE OUTP;GO TO INCST;
                                                % PAGE STATEMENT
      PAG:     IF OU=1 THEN WRITE(LIN[PAGE]);GO TO INCST;
                                                % MAT STATEMENT
                  %  MAT   1=READ, 2=PRINT, 3=MATOP, 4=SCALAR MULTIPLE
      MAT:     BEGIN INTEGER E,F,G;LABEL QDAT,FREA,XREA,RREA;
L:=GET;IF L=1 THEN BEGIN E:=CS;L:=GET;U:=GET;V:=GET;
FOR W:=0 STEP 1 UNTIL U-1 DO FOR X:=0 STEP 1 UNTIL V-1 DO BEGIN
```

```
IF RDTP=0 THEN GO TO QDAT;CO:=RDTP;CS:=RDAT;                                    00153200
     XREA:    STORE[ARR[L,0]+W×ARR[L,2]+X]:=EVAL;                               00153300
RDTP:=IF MORE THEN CO ELSE 0;GO TO RREA;                                        00153400
     QDAT:    FOR CS:=RDAT+1 STEP 1 UNTIL MS DO BEGIN IF STYP[CS]               00153500
=12 THEN GO TO FREA END;CS:=E;ERROR(21);                                        00153600
     FREA:    RDAT:=CS;CO:=SPOB[CS];GO TO XREA;                                 00153700
     RREA:    END;CS:=E;GO TO INCST END;                                        00153800
IF L=2 THEN BEGIN                                                               00153900
IF DELTA(POUB,PIOB) GTR 0 THEN OUTP;                                            00154000
L:=GET;U:=GET;V:=GET;E:=GET;G:=14-E×7;FOR W:=0 STEP 1 UNTIL U-1 DO               00154100
BEGIN FOR X:=0 STEP 1 UNTIL V-1 DO BEGIN OUTNUM(STORE[ARR[L,0]+                 00154200
W×ARR[L,2]+X],E);PIOB:=POUB+G×((DELTA(POUB,PIOB)-1) DIV G +1)                   00154300
END;OUTP END;GO TO INCST END;                                                   00154400
IF L=3 THEN MATOP;                                                              00154500
L:=GET;R:=EVAL;W:=GET;                                                          00154600
IF ARR[L,1] NEQ ARR[W,1] OR ARR[L,2] NEQ ARR[W,2] THEN ERROR(23);               00154700
FOR U:=0 STEP 1 UNTIL ARR[L,1]-1 DO FOR V:=0 STEP 1 UNTIL ARR[L,2]-1            00154800
DO STORE[ARR[L,0]+U×ARR[L,2]+V]:=R×STORE[ARR[W,0]+U×ARR[L,2]+V];                00154900
GO TO INCST END;                                                                00155000
                                                % STOP OR END STATEMENT        00155100
     ENX:     TR3;IF DELTA(POUB,PIOB)>0 THEN OUTP;                              00155200
LOCK(FIL1);LOCK(FIL2);GO TO STOP;                                               00155300
    OUD:  IF Z=0 THEN ERROR(21);TR2(Z);CS:=Z;GO EXS;                            00155400
    OUF:  IF Z=0 THEN ERROR(26);TR2(Z);CS:=Z;GO EXS;                            00155500
    INTVR: ERROR(12);INDEXR: ERROR(13);                                         00155600
    DZER: ERROR(14);EXPVR: ERROR(16)  END;                                      00155700
     STOP:    WRITE(TTY,STP,"RUN   ");GO TO SOURCEIN;                           00155800
COMMENT                                                                         00155900
                PROGRAM WRITTEN BY MALCOLM CROWE                                00156000
                LANGUAGE DETAILS DECIDED BY IAN MILLER AND JOHN FURLONG         00156100
                MATRIX INVERSION CORRECTED BY ARTHUR MACDIVITT        ;         00156200
                                                                                00156300
FINSH:     END.                                                                 00156400
```