000001

```
(*$U+ COMPILE UP TO COLUMN 72 ONLY*)

(*************************************************************
 *                                                           *
 *                                                           *
 *          COMPILER FOR PASCAL 6000 - 3.4                   *
 *          ******************************                   *
 *                                                           *
 *                                                           *
 *                RELEASE 2  MARCH 1976                      *
 *                                                           *
 *                                                           *
 *                                                           *
 *          CDC SCIENTIFIC CHAR SET VERSION                  *
 *      (00B AND 63B ARE TREATED IDENTICALLY)                *
 *                                                           *
 *     AUTHOR:   URS AMMANN                                  *
 *               INSTITUT FUER INFORMATIK                    *
 *               EIDG. TECHNISCHE HOCHSCHULE                 *
 *       CH-8006 ZUERICH                                     *
 *                                                           *
 *************************************************************)



(*$T-   COMPILE WITHOUT RUNTIME TESTS*)
(*$P-   COMPILE WITHOUT PMD INFORMATION*)
(*$X5   PASS UP TO FIVE PARAMETERS IN X-REGISTERS*)
(*$E+ USE ACTUAL PROCEDURE NAMES RATHER THAN PRCNNNN*)


PROGRAM PASCAL(INPUT+,OUTPUT,LGO);


LABEL 13;   (*EXIT IF EOF ENCOUNTERED*)


CONST DISPLIMIT = 20; MAXLEVEL = 10; MAXADDR = 377777B;
   WORDSIZE = 60;
   EFETSZ = 10; CHEFETSZ = 22;
   CHEFET = 13; BINEFET = 1; (*RELATIVE ADDRESSES OF EFETES*)
   RESWORDS = 35;
   NRSTDPROC = 23; NRSTDFUNC = 18; NREXFUNC = 7;
   TWOTO17 = 400000B;
   CODEMAX = 150; RCODEMAX = 10 (*CODEMAX DIV 15*) ;
   ASSERR = 46B; INXERR = 47B; DIVERR = 50B; STOVLERR = 51B;
   HALTERR = 52B; OVLERR = 53B; PNTERR = 54B;
   ALFALENG = 10;
   ERRMAX = 420;
   TITLEMSG = ≡ PASCAL VER 3.00-≡;
   SUBVERSION = ≡01 OF 76OCT22≡;


TYPE                          (*DESCRIBING:*)
                              (*************)


                               (*BASIC SYMBOLS*)
                               (***************)

   SYMBOL = (IDENT,INTCONST,REALCONST,CHARCONST,STRINGCONST,NOTSY,MULOP,
        ADDOP,RELOP,LPARENT,RPARENT,LBRACK,RBRACK,COMMA,SEMICOLON,
        PERIOD,ARROW,COLON,BECOMES,LABELSY,CONSTSY,TYPESY,VARSY,
        FUNCTIONSY,PROCEDURESY,SETSY,PACKEDSY,ARRAYSY,RECORDSY,FILESY,
        BEGINSY,IFSY,CASESY,REPEATSY,WHILESY,FORSY,WITHSY,
        GOTOSY,ENDSY,ELSESY,UNTILSY,OFSY,DOSY,TOSY,DOWNTOSY,
        THENSY,PROGRAMSY,SEGMENTEDSY,OTHERSY);
   OPERATOR = (MUL,RDIV,ANDOP,IDIV,IMOD,PLUS,MINUS,OROP,LTOP,LEOP,GEOP,
        GTOP,NEOP,EQOP,INOP,NOOP);
   SETOFSYS = SET OF SYMBOL;

                               (*CONSTANTS*)
                               (***********)

   CODERANGE = 0..CODEMAX; POSRANGE = 1..4;
   ADDRRANGE = 0..MAXADDR;
   SHRTINT = -377777B..377777B;
   CSTCLASS = (INT,BOOL,REEL,PSET,STRG);
```

```
CSP = ↑ CSTHEADREC; LOCOFREF = ↑LOCREC;                                    COMP        8
CTAILP = ↑ CSTTAILREC;                                                     COMP        8
CSTHEADREC = PACKED RECORD NXTCSP: CSP;                        000002      COMP        8
               CSTP: CTAILP;                                               COMP        8
               CREF: LOCOFREF                                              COMP        8
             END;                                                          COMP        8
CSTTAILREC = RECORD NXTCSP: CTAILP; CSVAL: INTEGER END;                    COMP        8
ERRINDEX = 1 .. ERRMAX;                                                    COMP        8
ERLISTT = PACKED ARRAY [ERRINDEX] OF BOOLEAN;                              COMP        8
                                                                           COMP        8
VALU = RECORD CASE CSTCLASS OF                                             COMP        9
         INT: (IVAL: INTEGER);                                             COMP        9
         BOOL: (BVAL: BOOLEAN);                                            COMP        9
         REEL: (RVAL: REAL);                                               COMP        9
         PSET: (PVAL: SET OF 0..58); (*IMPL. DEPENDANT RANGE*)             COMP        9
         STRG: (VALP: CTAILP)                                              COMP        9
       END;                                                                COMP        9
                                                                           COMP        9
                                (*DATA STRUCTURES*)                        COMP        9
                                (****************)                        COMP        9
LEVRANGE = 0..MAXLEVEL;                                                    COMP       10
BITRANGE = 0..59 (*=WORDSIZE-1*);                                          COMP       10
STRUCTFORM = (SCALAR,SUBRANGE,POINTER,POWER,ARRAYS,RECORDS,FILES,          COMP       10
       TAGFIELD,VARIANT);                                                  COMP       10
DECLKIND = (STANDARD,DECLARED);                                            COMP       10
WBSIZE = PACKED RECORD WORDS: ADDRRANGE;                                   COMP       10
          BITS: BITRANGE                                                   COMP       10
         END;                                                              COMP       10
STP = ↑ STRUCTREC; CTP = ↑ IDENTREC;                                       COMP       10
                                                                           COMP       10
STRUCTREC = PACKED RECORD                                                  COMP       11
       FTYPE: BOOLEAN;                                                     COMP       11
       SIZE: WBSIZE;                                                       COMP       11
       CASE FORM: STRUCTFORM OF                                            COMP       11
        SCALAR:    (CASE SCALKIND: DECLKIND OF                             COMP       11
                    DECLARED: (FCONST: CTP));                              COMP       11
        SUBRANGE: (RANGETYPE: STP; MIN,MAX: VALU);                         COMP       11
        POINTER:  (ELTYPE: STP);                                           COMP       11
        POWER:    (PCKDSET: BOOLEAN; ELSET: STP);                          COMP       11
        ARRAYS:   (AELTYPE,INXTYPE: STP;                                   COMP       11
                   CASE PCKDARR: BOOLEAN OF                                COMP       12
                    TRUE: (CASE PARTWORDELS: BOOLEAN OF                    COMP       12
                           TRUE: (ELSPERWORD: 2..60)));                    COMP       12
        RECORDS:  (PCKDREC: BOOLEAN; FIELDS,FSTFLD: CTP;                   COMP       12
                   RECVAR: STP);                                           COMP       12
        FILES:    (PCKDFIL,TEXTFILE,SEGFILE: BOOLEAN;                      COMP       12
                   FILTYPE: STP);                                          COMP       12
        TAGFIELD: (TAGFIELDP: CTP; FSTVAR: STP);                           COMP       12
        VARIANT:  (FSTVARFLD: CTP; NXTVAR,SUBVAR: STP;                     COMP       12
                   VARVAL: VALU)                                           COMP       12
       END;                                                                COMP       13
                                                                           COMP       13
EXTIDP = ↑EXTID; EXTREFP = ↑EXTREF;                                        COMP       13
EXTREF = PACKED RECORD LOC: 0..7777777777B; LINK: EXTREFP END;             COMP       13
EXTID = PACKED RECORD                                                      COMP       13
         ID: ALFA; L,R: EXTIDP; REF: EXTREFP                               COMP       13
        END;                                                               COMP       13
                                                                           COMP       13
                                (*NAMES*)                                  COMP       13
                                (*******)                                  COMP       13
                                                                           COMP       14
IDCLASS = (TYPES,KONST,VARS,FIELD,PROC,FUNC);                              COMP       14
SETOFIDS = SET OF IDCLASS;                                                 COMP       14
IDKIND = (ACTUAL,FORMAL);                                                  COMP       14
ACCESSKIND = (DRCT,INDRCT,INXD);                                           COMP       14
DRCTINDRCT = DRCT..INDRCT;                                                 COMP       14
                                                                           COMP       14
IDENTREC = PACKED RECORD                                                   COMP       14
       NAME: ALFA; LLINK: CTP; RLINK: CTP;                                 COMP       14
       IDTYPE: STP; NEXT: CTP;                                             COMP       14
       CASE KLASS: IDCLASS OF                                              COMP       15
        KONST:  (VALUES: VALU);                                            COMP       15
        VARS:   (VKIND: DRCTINDRCT; VLEV: LEVRANGE;                        COMP       15
                 VADDR: ADDRRANGE);                                        COMP       15
        FIELD:  (FLDADDR: ADDRRANGE;                                       COMP       15
                 CASE PCKDFLD: BOOLEAN OF                                  COMP       15
                  TRUE: (BITADDR: BITRANGE));                              COMP       15
        PROC,                                                              COMP       15
        FUNC:   (CASE PFDECKIND: DECLKIND OF                               COMP       15
                 STANDARD: (KEY: 1..25); (*SAME RANGE AS LKEY*)            COMP       15
                 DECLARED: (PFLEV: LEVRANGE; PFADDR: ADDRRANGE;            COMP       16
```

```
                                PFXOPT : 0..6;
                                CASE PFKIND: IDKIND OF
                                 ACTUAL: (PFDECL: (DECL,FORWDECL,
                                                   EXTDECL,FTNDECL);
                                          PFCNT: 0..7777B)))
        END;

EXTFILEP = ↑ FILEREC;
FILEREC = PACKED RECORD                                          000003
            FILENAME: ALFA;
            NXTP: EXTFILEP;
            DECLARED,READONLY: BOOLEAN;
            SYSLOC: 2..63B
          END;

DISPRANGE = 0..DISPLIMIT;
WHERE = (BLCK,REC);


                                (*LABELS*)
                                (********)
LBP = ↑LABREC;
LABREC = PACKED RECORD
           LABVAL: INTEGER; NEXTLAB: LBP;
           LABLEV: LEVRANGE; LCNT: 0..36;
           CASE DEFINED: BOOLEAN OF
            TRUE:  (LABADDR: ADDRRANGE);
            FALSE: (FSTOCC: LOCOFREF)
         END;


           (*TO DESCRIBE EXPRESSION CURRENTLY COMPILED*)
           (*****************************************)

   ATTRKIND = (CST,VARBL,COND,EXPR);
   REGKIND = (NONE,XREG);
   REGNR = 0..7;

   ATTR = RECORD TYPTR: STP;
        CASE KIND: ATTRKIND OF
         CST:   (CVAL: VALU);
         VARBL: (WORDACC: ACCESSKIND;
                 VLEVEL: LEVRANGE; DPDISPL: SHRTINT;
                 VWDISPL: REGNR;
                 CASE PCKD: BOOLEAN OF
                  TRUE: (CBDISPL: SHRTINT;
                         BITREG: REGKIND; VBDISPL: REGNR));
         COND:  (CDR: REGNR; CONDCD: (ZR,NZ,PL,NG));
         EXPR:  (EXPREG: REGNR)
        END;


           (*TO DESCRIBE REGISTER STATUS*)
           (****************************)

   ARGSTR = (SIMPADDR,INDADDR,UNSPECADDR);
   XRGSTR = (AVAIL,SHRTCST,LONGCST,SIMPVAR,INDVAR,OTHER);
    REMXRG = SHRTCST..INDVAR;
   BRGSTR = (FREE,BASADDR,SPECPURP);

   ARGSTAT =
      PACKED RECORD CASE ACONT: ARGSTR OF
               SIMPADDR: (ALEV: LEVRANGE; AADDR: ADDRRANGE);
               INDADDR:  (AREG: REGNR; ADISPL: ADDRRANGE)
              END;

   XRGSTAT =
     PACKED RECORD
      CASE XCONT: XRGSTR OF
       SHRTCST,LONGCST,
       SIMPVAR,INDVAR,
       OTHER:
        (REFNR: 0..100; LASTREF: ADDRRANGE;
         CASE REMXRG OF
         SHRTCST:
           (CSTVAL: SHRTINT);
         LONGCST:
           (CPTR: CTAILP);
         SIMPVAR,
         INDVAR:
```

```
COMP     16
COMP     16
COMP     16
COMP     16
COMP     16
COMP     16
COMP     16
COMP     16
COMP     17
COMP     17
COMP     17
COMP     17
COMP     17
COMP     17
COMP     17
COMP     17
COMP     17
COMP     18
COMP     18
COMP     18
COMP     18
COMP     18
COMP     18
COMP     18
COMP     18
COMP     19
COMP     19
COMP     19
COMP     19
COMP     19
COMP     19
COMP     19
COMP     19
COMP     19
COMP     20
COMP     20
COMP     20
COMP     20
COMP     20
COMP     20
COMP     20
COMP     20
COMP     21
COMP     21
COMP     21
COMP     21
COMP     21
COMP     21
COMP     21
COMP     21
COMP     21
COMP     21
COMP     21
COMP     22
COMP     22
COMP     22
COMP     22
COMP     22
COMP     22
COMP     22
COMP     22
COMP     22
COMP     23
COMP     23
COMP     23
COMP     23
COMP     23
COMP     23
COMP     23
COMP     23
COMP     23
COMP     23
COMP     24
COMP     24
```

```
                (SHFTCNT: BITRANGE;
                CASE DRCT:INDRCT OF
                  DRCT:
                    (XLEV: LEVRANGE; XADDR: ADDRRANGE;                    (* 000004
                     VPADDR: BOOLEAN);
                  INDRCT:
                    (XREG: REGNR; XDISPL: ADDRRANGE)))
          END;

        BRGSTAT =
          PACKED RECORD CASE BCONT: BRGSTR OF
                   BASADDR: (BLEV: LEVRANGE)
                 END;

        ARGSTATUS = ARRAY [REGNR] OF ARGSTAT;
        XRGSTATUS = ARRAY [REGNR] OF XRGSTAT;
        BRGSTATUS = ARRAY [REGNR] OF BRGSTAT;
        BASREGS = ARRAY [LEVRANGE] OF REGNR;


    PLACE = PACKED RECORD SIX: ADDRRANGE;
                    CIX: CODERANGE; CP: POSRANGE
                  END;

    LOCREC = PACKED RECORD NXTREF: LOCOFREF; LOC: PLACE END;

                (*MISCELLANEOUS*)
                (***************)

      PCRP = ^PTRCOMPREC;                       (*TO AVOID NEVER ENDING*)
      PTRCOMPREC = RECORD NEXT: PCRP;           (*RECURSION IN COMPTYPES*)
                     PTR1,PTR2: STP
                   END;

(*-------------------------------------------------------------------*)

VAR
                (*RETURNED BY SOURCE PROGRAM SCANNER
                 INSYMBOL:
                 **********)

    SY: SYMBOL;                             (*LAST SYMBOL*)
    OP: OPERATOR;                           (*CLASSIFICATION OF LAST SYMBOL*)
    IVAL: INTEGER;                          (*VALUE OF LAST INTEGER CONSTANT*)
    RVAL: REAL;                             (*VALUE OF LAST REAL CONSTANT*)
    CONSTP: CTAILP;                         (*POINTER TO LAST STRING*)
    LGTH: INTEGER;                          (*LENGTH OF LAST STRING CONSTANT*)
    ID: ALFA;                               (*LAST IDENT (POSSIBLY TRUNCATED)*)
    CH: CHAR;                               (*LAST CHARACTER*)

                (*COUNTERS:*)
                (***********)

    CHCNT,MAXCHCNT: INTEGER;                (*CHARACTER COUNTER*)
    LC,IC: INTEGER;                         (*DATA LOCATION AND INSTR CNTER*)
    FILECNT,
    LABCNT: 0..36;                          (*NUMBER OF LOC FILES IN NESTED USE*)
    EXTFILS: 0..50;                         (*NUMBER OF EXTERNAL LABELS*)
    BUFFAC: INTEGER;                        (*NUMBER OF EXTERNAL FILES*)
    XPARMAX: INTEGER;                       (*FACTOR TO COMPUTE FILE BUFFER SIZE*)
    PCNT: INTEGER;                          (*NR OF PARAMS TO BE PASSED IN X-RG*)
    B6DPL: ADDRRANGE;                       (*NUMBER OF PROCEDURES/FUNCTIONS*)

                (*SWITCHES:*)
                (************)

    DP,
    PRTERR: BOOLEAN;                        (*DECLARATION PART*)
                                            (*TO ALLOW FORWARD REFERENCES
                                            BY SUPPRESSING ERROR MESSAGE*)
    DEBUG,EXTON,LISTON,PMD: BOOLEAN;

                (*POINTERS:*)
                (************)
    INTPTR,REALPTR,CHARPTR,ALFAPTR,STEXTPTR,
    BOOLPTR,NILPTR,TEXTPTR: STP;            (*POINTERS TO ENTRIES OF STD IDS*)
```

```
UTYPPTR,UCSTPTR,UVARPTR;
UFLDPTR,UPRCPTR,UFCTPTR;          (*POINTERS TO ENTRIES FOR UNDECL IDS*)
INPUTPTR,OUTPUTPTR,               (*ENTRIES FOR INPUT AND OUTPUT*)
FWPTR: CTP;                       (*HEAD OF CHAIN OF FORW  TYPE IDS*)
FSTLABP: LBP;                     (*HEAD OF LABEL CHAIN*)
FEXFILP: EXTFILEP;                (*HEAD OF LIST OF EXTERNAL FILES*)
FSTCSP: CSP;                      (*HEAD OF CONSTANT CHAIN*)
 FSTPCRP: PCRP;                    (*HEAD OF LIST OF POINTER PAIRS*)


              (*BOOKKEEPING OF DECLARATION LEVELS:*)
              (*********************************)

LEVEL: LEVRANGE;                  (*CURRENT STATIC LEVEL*)
DISX,                             (*LEVEL OF LAST ID SRCHD BY SEARCHID*)
TOP: DISPRANGE;                   (*TOP OF DISPLAY*)

DISPLAY:                          (*WHERE:    MEANS:*)
 ARRAY [DISPRANGE] OF
  PACKED RECORD                  (*=BLCK:    ID IS VARIABLE ID*)
   FNAME: CTP;                  (*=REC:    ID IS FIELD ID IN RECORD*)
   CASE OCCUR: WHERE OF
    REC: (WACC: DRCTINDRCT;
      LEV: LEVRANGE; CHDSPL: ADDRRANGE;
      CASE PKD: BOOLEAN OF
        TRUE: (BACC: DRCTINDRCT; BDSPL: SHRTINT))
   END;


              (*ERROR MESSAGES:*)
              (*****************)

ERRINX: 0..10;                    (*NR OF ERRORS IN CURR SOURCE LINE*)
ERRORS: BOOLEAN;
ERRLIST:
 ARRAY [1..10] OF
  PACKED RECORD POS: 1..1000000;
       NMR: ERRINDEX
     END;
ERLIST : ERLISTT;


              (*CODE GENERATION*)
               (*****************)

GATTR: ATTR;
ARGS: ARGSTATUS; XRGS: XRGSTATUS; BRGS: BRGSTATUS;
BRG: BASREGS;
LEVELS: SET OF LEVRANGE;
PC: PLACE; RBUF,CBUF: INTEGER;


              (*CODEFILE AND TABLES FOR EXT. REFERENCES*)
              (*****************************************)

(*$B1  WILL NOT ALLOW LGO TAPES*)
LGO: SEGMENTED FILE OF INTEGER;
PROGNAME: ALFA;
EXT, EXTROOT: EXTIDP; EXTIDX, EXTRX: INTEGER;
CADDR,                            (*ADDRESS FOR NEXT TEXTTABLE*)
CODEADDR: ADDRRANGE;              (*ADDRESS OF
                                   CURRENT CODE SEGMENT*)
ALFINT: RECORD CASE BOOLEAN OF
        FALSE: (A: ALFA);
        TRUE:  (I: INTEGER)
        END;
CURDATE,CURTIME:ALFA;


                                                          000005

              (*STRUCTURED CONSTANTS:*)
              (************************)

LETTERS: SET OF ΞAΞ..ΞZΞ; DIGITS: SET OF Ξ0Ξ..Ξ9Ξ;
CONSTBEGSYS,SIMPTYPEBEGSYS,TYPEBEGSYS,BLOCKBEGSYS,SELECTSYS,FACBEGSYS,
STATBEGSYS,TYPEDELS: SETOFSYS;
RW: ARRAY [1..RESWORDS] OF ALFA;
LRW: ARRAY [0..ALFALENG] OF 0..RESWORDS;
RSY: ARRAY [1..RESWORDS] OF SYMBOL;
```

```
  SSY: ARRAY [≡+≡..≡;≡] OF SYMBOL;
  ROP: ARRAY [1..RESWORDS] OF OPERATOR;
  SOP: ARRAY [≡+≡..≡;≡] OF OPERATOR;
  PASCF,PASCL,PNAME: ALFA;



(*-----------------------------------------------------------------*)


PROCEDURE WRITEERRORS;
 VAR LASTPOS,FREEPOS,CURRPOS,CURRNMR,F,K: INTEGER;
BEGIN WRITE(≡ ***   ≡);
 LASTPOS := 0; FREEPOS := 1;
 FOR K := 1 TO ERRINX DO
  BEGIN
   WITH ERRLIST[K] DO
    BEGIN CURRPOS := POS; CURRNMR := NMR END;
   IF CURRPOS = LASTPOS THEN WRITE(≡,≡)
   ELSE
    BEGIN
     WHILE FREEPOS < CURRPOS DO
      BEGIN WRITE(≡ ≡); FREEPOS := FREEPOS + 1 END;
     WRITE(≡↑≡);
      LASTPOS := CURRPOS
    END;
   IF CURRNMR < 10 THEN F := 1
   ELSE IF CURRNMR < 100 THEN F := 2
    ELSE F := 3;
   WRITE(CURRNMR:F);
   FREEPOS := FREEPOS + F + 1
  END;
 WRITELN; ERRINX := 0
END (*WRITEERRORS*) ;

PROCEDURE ENDOFLINE;
BEGIN IF LISTON THEN WRITELN;
 IF ERRINX > 0 THEN WRITEERRORS;
 IF LISTON THEN
  BEGIN WRITE(≡ ≡);
   IF DP THEN WRITE(LC:6 OCT) ELSE WRITE(IC:6 OCT);
   WRITE(≡ ≡);
  END;
 IF EOS(INPUT) THEN
  BEGIN WRITELN; WRITELN(≡ ***     EOF ENCOUNTERED≡); ERRORS:=TRUE;
   GOTO 13
  END;
 CHCNT := 0
END  (*ENDOFLINE*) ;

PROCEDURE ERROR(FERRNR: ERRINDEX);
BEGIN ERRORS := TRUE; ERLIST[FERRNR] := TRUE;
 IF ERRINX >= 9 THEN
  BEGIN ERRLIST[10].NMR := 255; ERRINX := 10 END
 ELSE
  BEGIN ERRINX := ERRINX + 1;
   ERRLIST[ERRINX].NMR := FERRNR
  END;
 ERRLIST[ERRINX].POS := CHCNT
END (*ERROR*) ;

PROCEDURE SPACES;
BEGIN
 IF NOT EOLN(INPUT) THEN WRITE(≡ ≡:CHCNT+8)
END (* SPACES *);

PROCEDURE INSYMBOL;
 (*READ NEXT BASIC SYMBOL OF SOURCE PROGRAM AND RETURN ITS DESCRIPTION
 IN THE GLOBAL VARIABLES SY, OP, ID, IVAL, RVAL, SVAL AND LGTH*)
 LABEL 1,2;
 CONST DIGMAX = 20; MAX8 = 77777777777777777B;
     MAX10 = 314631463146314631 3B;
 VAR I,K,SCALE,EXP: INTEGER;
    R,FAC: REAL; SIGN: BOOLEAN;
    DIGIT: ARRAY [1..DIGMAX] OF 0..9;
    APO,STRINGEND: BOOLEAN; NXTP,TAILP: CTAILP;
    OA: RECORD
          CASE BOOLEAN OF
           FALSE: (A: ALFA);
            TRUE:  (I: INTEGER)
```

```
      END;                                                                    COMP      48
                                                                              COMP      48
PROCEDURE NEXTCH;                                                    000007   COMP      48
BEGIN                                                                         COMP      48
GET(INPUT); CH := INPUT↑; IF LISTON THEN WRITE(CH);                           COMP      48
CHCNT := CHCNT + 1;                                                           COMP      48
IF CHCNT > MAXCHCNT THEN                                                      COMP      49
 BEGIN                                                                        COMP      49
  WHILE NOT EOLN(INPUT) DO                                                    COMP      49
   BEGIN GET(INPUT); IF LISTON THEN WRITE(INPUT↑) END;                        COMP      49
  CH := ' '                                                                   COMP      49
 END                                                                          COMP      49
END (*NEXTCH*) ;                                                              COMP      49
                                                                              COMP      49
                                                                              COMP      49
PROCEDURE OPTIONS;                                                            COMP      49
 VAR ENDOPTIONS: BOOLEAN; CH1: CHAR;                                          COMP      50
 BEGIN ENDOPTIONS := FALSE;                                                   COMP      50
 REPEAT NEXTCH;                                                               COMP      50
  IF CH IN ['B','E','L','P','T','U','X'] THEN                                 COMP      50
   BEGIN CH1 := CH; NEXTCH;                                                   COMP      50
    CASE CH1 OF                                                               COMP      50
     'B': IF CH IN ['1'..'9'] THEN BUFFAC := ORD(CH) - ORD('0')               COMP      50
          ELSE ENDOPTIONS := TRUE;                                            COMP      50
     'E': IF CH IN ['+','-'] THEN EXTON := CH = '+'                           COMP      50
          ELSE ENDOPTIONS := TRUE;                                            COMP      50
     'L': IF CH IN ['+','-'] THEN                                             COMP      51
           BEGIN LISTON := CH = '+';                                          COMP      51
            IF LISTON THEN WRITE(' ') ELSE WRITELN                            COMP      51
           END                                                                COMP      51
          ELSE ENDOPTIONS := TRUE;                                            COMP      51
     'P': IF CH IN ['+','-'] THEN PMD := CH = '+'                             COMP      51
          ELSE ENDOPTIONS := TRUE;                                            COMP      51
     'T': IF CH IN ['+','-'] THEN DEBUG := CH = '+'                           COMP      51
          ELSE ENDOPTIONS := TRUE;                                            COMP      51
     'U': IF CH IN ['+','-'] THEN                                             COMP      51
           IF CH = '+' THEN MAXCHCNT := 72 ELSE MAXCHCNT := 120               COMP      52
          ELSE ENDOPTIONS := TRUE;                                            COMP      52
     'X': IF CH IN ['0'..'6'] THEN XPARMAX := ORD(CH) - ORD('0')              COMP      52
          ELSE ENDOPTIONS := TRUE;                                            COMP      52
    END;                                                                      COMP      52
    IF NOT ENDOPTIONS THEN                                                    COMP      52
     BEGIN NEXTCH; ENDOPTIONS := CH <> ',' END                                COMP      52
   END                                                                        COMP      52
  ELSE ENDOPTIONS := TRUE                                                     COMP      52
 UNTIL ENDOPTIONS;                                                            COMP      52
END (*OPTIONS*);                                                              COMP      53
                                                                              COMP      53
BEGIN (*INSYMBOL*)                                                            COMP      53
1:                                                                            COMP      53
 CASE CH OF                                                                   COMP      53
    ' ':                                                                      COMP      53
     BEGIN                                                                    COMP      53
     IF LISTON THEN                                                           COMP      53
     BEGIN                                                                    COMP      53
       REPEAT                                                                 COMP      53
        IF EOLN(INPUT) THEN ENDOFLINE;                                        COMP      54
        GET(INPUT); CH := INPUT↑; WRITE(CH);                                  COMP      54
        CHCNT := CHCNT + 1                                                    COMP      54
       UNTIL CH <> ' ';                                                       COMP      54
       IF CHCNT > MAXCHCNT THEN                                               COMP      54
        BEGIN                                                                 COMP      54
        WHILE NOT EOLN(INPUT) DO                                              COMP      54
         BEGIN GET(INPUT); WRITE(INPUT↑) END;                                 COMP      54
        CH := INPUT↑                                                          COMP      54
        END                                                                   COMP      54
     END                                                                      COMP      55
     ELSE                                                                     COMP      55
      BEGIN                                                                   COMP      55
       REPEAT                                                                 COMP      55
        IF EOLN(INPUT) THEN ENDOFLINE;                                        COMP      55
        GET(INPUT); CHCNT := CHCNT + 1;                                       COMP      55
       UNTIL INPUT↑ <> ' ';                                                   COMP      55
       IF CHCNT > MAXCHCNT THEN                                               COMP      55
        WHILE NOT EOLN(INPUT) DO GET(INPUT);                                  COMP      55
       CH := INPUT↑                                                           COMP      55
      END;                                                                    COMP      56
     GOTO 1                                                                   COMP      56
    END;                                                                      COMP      56
  'A','B','C','D','E','F','G','H','I',                                       COMP      56
  'J','K','L','M','N','O','P','Q','R',                                        COMP      56
```

```
≡S≡,≡T≡,≡U≡,≡V≡,≡W≡,≡X≡,≡Y≡,≡Z≡:
 BEGIN K := 0; ID := ≡          ≡;
  REPEAT
   IF K < ALFALENG THEN
    BEGIN K := K + 1; ID[K] := CH END;
   NEXTCH
  UNTIL NOT (CH IN LETTERS+DIGITS);
  FOR I := LRW[K-1] + 1 TO LRW[K] DO
   IF RW[I] = ID THEN
    BEGIN SY := RSY[I]; OP := ROP[I]; GOTO 2 END;
   SY := IDENT; OP := NOOP;
2:  END;
≡0≡,≡1≡,≡2≡,≡3≡,≡4≡,≡5≡,≡6≡,≡7≡,≡8≡,≡9≡:
 BEGIN SY := INTCONST; OP := NOOP;
  I := 0;
  REPEAT I := I + 1;
   IF I <= DIGMAX THEN DIGIT[I] := ORD(CH) - ORD(≡0≡)
   ELSE
    BEGIN ERROR(203); I := 1 END;
   NEXTCH
  UNTIL NOT (CH IN DIGITS);
  IVAL := 0;
  IF CH = ≡B≡ THEN
   BEGIN
    FOR K := 1 TO I DO
     IF IVAL <= MAX8 THEN
      BEGIN IF DIGIT[K] IN [8,9] THEN ERROR(204);
       IVAL := 8*IVAL + DIGIT[K]
      END
     ELSE BEGIN ERROR(203); IVAL := 0 END;
    NEXTCH
   END
  ELSE
   BEGIN
    FOR K := 1 TO I DO
     IF IVAL <= MAX10 THEN
      IVAL := 10*IVAL + DIGIT[K]
     ELSE BEGIN ERROR(203); IVAL := 0 END;
    SCALE := 0;
    IF CH = ≡.≡ THEN
     BEGIN NEXTCH;
      IF CH = ≡.≡ THEN CH := ≡:≡
      ELSE
       BEGIN RVAL := IVAL; SY := REALCONST;
        IF IVAL DIV 10000000000000000B <> 0 THEN ERROR(206);
        IF NOT (CH IN DIGITS) THEN ERROR(201)
        ELSE
         REPEAT RVAL := 10.0*RVAL + (ORD(CH) - ORD(≡0≡));
          SCALE := SCALE - 1; NEXTCH
         UNTIL NOT (CH IN DIGITS)
       END
     END;
    IF CH = ≡E≡ THEN
     BEGIN
      IF SCALE = 0 THEN
       BEGIN RVAL := IVAL; SY := REALCONST END;
      SIGN := FALSE; NEXTCH;
      IF CH = ≡+≡ THEN NEXTCH
      ELSE
       IF CH = ≡-≡ THEN
        BEGIN SIGN := TRUE; NEXTCH END;
      EXP := 0;
      IF NOT (CH IN DIGITS) THEN ERROR(201)
      ELSE
       REPEAT EXP := 10*EXP + (ORD(CH) - ORD(≡0≡));
        NEXTCH
       UNTIL NOT (CH IN DIGITS);
      IF SIGN THEN SCALE := SCALE - EXP
      ELSE SCALE := SCALE + EXP
     END;
    IF SCALE <> 0 THEN
     BEGIN R := 1.0;    (*NOTE POSSIBLE OVERFLOW+UNDERFLOW*)
      SIGN := SCALE < 0; SCALE := ABS(SCALE); FAC := 10.0;
      REPEAT IF ODD(SCALE) THEN R := R*FAC;
       FAC := SQR(FAC); SCALE := SCALE DIV 2
      UNTIL SCALE = 0;   (*NOW R = 10↑SCALE*)
      IF SIGN THEN RVAL := RVAL/R ELSE RVAL := RVAL*R
     END;
   END
 END;
====
```

```
    :                                                                    COMP      64
  BEGIN OP := NOOP; APO := FALSE; STRINGEND := FALSE;                    COMP      64
   LGTH := 0; I := 0; CONSTP := NIL;                                     COMP      64
   NEXTCH;                                                    000009     COMP      64
   REPEAT                                                                COMP      65
    IF EOLN(INPUT) THEN                                                  COMP      65
     BEGIN ERROR(202); STRINGEND := TRUE END                             COMP      65
    ELSE                                                                 COMP      65
     IF (CH <> ''')OR APO THEN                                           COMP      65
      BEGIN                                                              COMP      65
       IF I = ALFALENG THEN                                              COMP      65
        BEGIN NEW(TAILP);                                                COMP      65
         WITH TAILP↑ DO                                                  COMP      65
          BEGIN NXTCSP := CONSTP; CSVAL := OA.I END;                     COMP      65
         CONSTP := TAILP; I := 0                                         COMP      66
        END;                                                             COMP      66
       I := I + 1; LGTH := LGTH + 1; APO := FALSE;                       COMP      66
       OA.A[I] := CH;                                                    COMP      66
       NEXTCH                                                            COMP      66
      END                                                                COMP      66
     ELSE                                                                COMP      66
       BEGIN APO := TRUE;                                                COMP      66
       NEXTCH; STRINGEND :=                                              COMP      66
                            CH <> ''''                                   COMP      66
      END                                                                COMP      67
   UNTIL STRINGEND;                                                      COMP      67
   SY:=STRINGCONST;                                                      COMP      67
   IF LGTH = 0 THEN ERROR(205)                                           COMP      67
  ELSE                                                                   COMP      67
   IF LGTH = 1 THEN                                                      COMP      67
     BEGIN SY := CHARCONST; IVAL := ORD(OA.A[1]) END                     COMP      67
   ELSE                                                                  COMP      67
     BEGIN FOR I := I+1 TO ALFALENG DO OA.A[I] := CHR(0);                COMP      67
     NEW(TAILP);                                                         COMP      67
     WITH TAILP↑ DO                                                      COMP      68
      BEGIN NXTCSP := CONSTP; CSVAL := OA.I END;                         COMP      68
     (*REVERSE POINTERS:*)                                               COMP      68
     CONSTP := NIL;                                                      COMP      68
     WHILE TAILP <> NIL DO                                               COMP      68
      WITH TAILP↑ DO                                                     COMP      68
       BEGIN NXTP := NXTCSP; NXTCSP := CONSTP;                           COMP      68
        CONSTP := TAILP; TAILP := NXTP                                   COMP      68
       END;                                                              COMP      68
     END                                                                 COMP      68
  END;                                                                   COMP      69
COL,':'                                                                  COMP      69
    :                                                                    COMP      69
  BEGIN OP := NOOP; NEXTCH;                                              COMP      69
   IF CH = '=' THEN                                                      COMP      69
    BEGIN SY := BECOMES; NEXTCH END                                      COMP      69
   ELSE SY := COLON                                                      COMP      69
  END;                                                                   COMP      69
'.':                                                                     COMP      69
  BEGIN OP := NOOP; NEXTCH;                                              COMP      69
   IF CH = '.' THEN                                                      COMP      70
    BEGIN SY := COLON; NEXTCH END                                        COMP      70
   ELSE SY := PERIOD                                                     COMP      70
  END;                                                                   COMP      70
'{':                                                                     COMP      70
  BEGIN NEXTCH;                                                          COMP      70
   IF CH = '$' THEN OPTIONS;                                             COMP      70
   IF EOLN(INPUT) THEN ENDOFLINE;                                        COMP      70
   WHILE CH <> '}' DO                                                    COMP      70
    BEGIN NEXTCH;                                                        COMP      70
     IF EOLN(INPUT) THEN ENDOFLINE;                                      COMP      71
    END;                                                                 COMP      71
   NEXTCH; GOTO 1                                                        COMP      71
  END;                                                                   COMP      71
'(':                                                                     COMP      71
  BEGIN NEXTCH;                                                          COMP      71
   IF CH = '*' THEN                                                      COMP      71
    BEGIN NEXTCH;                                                        COMP      71
     IF CH = '$' THEN OPTIONS;                                           COMP      71
     REPEAT                                                              COMP      71
      (*LOOP UNTIL CH = '*':*)                                           COMP      72
       IF EOLN(INPUT) THEN ENDOFLINE;                                    COMP      72
      WHILE CH <> '*' DO                                                 COMP      72
       BEGIN NEXTCH;                                                     COMP      72
         IF EOLN(INPUT) THEN ENDOFLINE;                                  COMP      72
       END;                                                              COMP      72
      NEXTCH                                                             COMP      72
```

```
      UNTIL CH = ≡)≡;
      NEXTCH; GOTO 1
     END;
    SY := LPARENT; OP := NOOP
   END;
  ≡<≡:
   BEGIN NEXTCH; SY := RELOP;
    IF CH = ≡=≡ THEN
     BEGIN OP := LEOP; NEXTCH END
    ELSE
     IF CH = ≡>≡ THEN
      BEGIN OP := NEOP; NEXTCH END
     ELSE OP := LTOP
   END;
  ≡>≡:
   BEGIN NEXTCH; SY := RELOP;
    IF CH = ≡=≡ THEN
     BEGIN OP := GEOP; NEXTCH END
    ELSE OP := GTOP
   END;
  ≡¬≡,≡*≡,≡/≡,≡^≡,≡+≡,≡-≡,
  ≡v≡,≡≤≡,≡≥≡,≡≠≡,≡=≡,≡)≡,
  ≡[≡,≡]≡,≡,≡,≡;≡,≡↑≡,≡$≡,≡↓≡
                                        :
   BEGIN SY := SSY[CH]; OP := SOP[CH];
    NEXTCH
   END;
 END (*CASE*)
END (*INSYMBOL*) ;

PROCEDURE ENTERID(FCP: CTP);
 (*ENTER ID POINTED AT BY FCP INTO THE NAME-TABLE,
  WHICH ON EACH DECLARATION LEVEL IS ORGANISED AS
  AN UNBALANCED BINARY TREE*)
 VAR NAM: ALFA; LCP, LCP1: CTP; LLEFT: BOOLEAN;
BEGIN NAM := FCP↑.NAME;
 LCP := DISPLAY[TOP].FNAME;
 IF LCP = NIL THEN
  DISPLAY[TOP].FNAME := FCP
 ELSE
  BEGIN
   REPEAT LCP1 := LCP;
    IF LCP↑.NAME = NAM THEN   (*NAME CONFLICT, FOLLOW RIGHT LINK*)
     BEGIN ERROR(101); LCP := LCP↑.RLINK; LLEFT := FALSE END
    ELSE
     IF LCP↑.NAME < NAM THEN
      BEGIN LCP := LCP↑.RLINK; LLEFT := FALSE END
     ELSE BEGIN LCP := LCP↑.LLINK; LLEFT := TRUE END
   UNTIL LCP = NIL;
   IF LLEFT THEN LCP1↑.LLINK := FCP ELSE LCP1↑.RLINK := FCP
  END;
 FCP↑.LLINK := NIL; FCP↑.RLINK := NIL
END (*ENTERID*) ;

PROCEDURE SEARCHSECTION(FCP: CTP; VAR FCP1: CTP);
 (*TO FIND RECORD FIELDS AND FORWARD DECLARED PROCEDURE ID=S
  --> PROCEDURE PROCEDUREDECLARATION
  --> PROCEDURE SELECTOR*)
 LABEL 1;
BEGIN
 WHILE FCP <> NIL DO
  IF FCP↑.NAME = ID THEN GOTO 1
  ELSE IF FCP↑.NAME < ID THEN FCP := FCP↑.RLINK
   ELSE FCP := FCP↑.LLINK;
1:  FCP1 := FCP
END (*SEARCHSECTION*) ;

PROCEDURE SEARCHID(FIDCLS: SETOFIDS; VAR FCP: CTP);
 LABEL 1;
 VAR LCP: CTP;
BEGIN
 FOR DISX := TOP DOWNTO 0 DO
  BEGIN LCP := DISPLAY[DISX].FNAME;
   WHILE LCP <> NIL DO
    IF LCP↑.NAME = ID THEN
     IF LCP↑.KLASS IN FIDCLS THEN GOTO 1
     ELSE
      BEGIN IF PRTERR THEN ERROR(103);
       LCP := LCP↑.RLINK
      END
    ELSE
```

```
      IF LCP↑.NAME < ID THEN                                                    COMP       80
       LCP := LCP↑.RLINK                                                        COMP       80
      ELSE LCP := LCP↑.LLINK                                                    COMP       81
   END;                                                          000011         COMP       81
  (*SEARCH NOT SUCCSESSFUL; SUPPRESS ERROR MESSAGE IN CASE                      COMP       81
   OF FORWARD REFERENCED TYPE ID IN POINTER TYPE DEFINITION                     COMP       81
   OR VARIANTS WITHOUT TAGFIELDS                                                COMP       81
   --> PROCEDURE FIELDLIST                                                      COMP       81
   --> PROCEDURE SIMPLETYPE*)                                                   COMP       81
  IF PRTERR THEN                                                               COMP       81
   BEGIN ERROR(104);                                                           COMP       81
    (*TO AVOID RETURNING NIL, REFERENCE AN ENTRY                               COMP       81
     FOR AN UNDECLARED ID OF APPROPRIATE CLASS                                 COMP       82
     --> PROCEDURE ENTERUNDECL*)                                               COMP       82
    IF TYPES IN FIDCLS THEN LCP := UTYPPTR                                     COMP       82
    ELSE                                                                       COMP       82
     IF VARS IN FIDCLS THEN LCP := UVARPTR                                     COMP       82
     ELSE                                                                      COMP       82
      IF FIELD IN FIDCLS THEN LCP := UFLDPTR                                   COMP       82
      ELSE                                                                     COMP       82
       IF KONST IN FIDCLS THEN LCP := UCSTPTR                                  COMP       82
       ELSE                                                                    COMP       82
        IF PROC IN FIDCLS THEN LCP := UPRCPTR                                  COMP       83
        ELSE LCP := UFCTPTR;                                                   COMP       83
   END;                                                                        COMP       83
1:  FCP := LCP                                                                 COMP       83
 END (*SEARCHID*) ;                                                            COMP       83
                                                                               COMP       83
                                                                               COMP       83
 PROCEDURE GETBOUNDS(FSP: STP; VAR FMIN,FMAX: INTEGER);                        COMP       83
  (*GET INTERNAL BOUNDS OF SUBRANGE OR SCALAR TYPE*)                           COMP       83
  (*ASSUME (FSP <> INTPTR) AND (FSP <> REALPTR)*)                              COMP       83
  BEGIN                                                                        COMP       84
   IF FSP <> NIL THEN                                                          COMP       84
    WITH FSP↑ DO                                                               COMP       84
     IF FORM = SUBRANGE THEN                                                   COMP       84
      BEGIN FMIN := MIN.IVAL; FMAX := MAX.IVAL END                             COMP       84
     ELSE                                                                      COMP       84
      BEGIN FMIN := 0; FMAX := 0;                                              COMP       84
       IF FORM = SCALAR THEN                                                   COMP       84
        BEGIN                                                                  COMP       84
         IF SCALKIND = STANDARD THEN                                           COMP       84
          BEGIN IF FSP = CHARPTR THEN FMAX := 63                               COMP       85
          END                                                                  COMP       85
         ELSE                                                                  COMP       85
          IF FSP↑.FCONST <> NIL THEN                                           COMP       85
           FMAX := FSP↑.FCONST↑.VALUES.IVAL                                    COMP       85
        END                                                                    COMP       85
      END                                                                      COMP       85
  END (*GETBOUNDS*);                                                           COMP       85
                                                                               COMP       85
 PROCEDURE SKIP(FSYS: SETOFSYS);                                               COMP       85
  (*SKIP INPUT STRING UNTIL RELEVANT SYMBOL FOUND*)                            COMP       86
 BEGIN WHILE NOT (SY IN FSYS) DO INSYMBOL                                      COMP       86
 END (*SKIP*) ;                                                                COMP       86
                                                                               COMP       86
 PROCEDURE TEST1( X: SYMBOL; Y: INTEGER);                                      COMP       86
 BEGIN IF SY = X THEN INSYMBOL ELSE ERROR(Y)                                   COMP       86
 END (*TEST1*) ;                                                               COMP       86
                                                                               COMP       86
 PROCEDURE TEST2(X: SETOFSYS; Y: INTEGER; Z: SETOFSYS);                        COMP       86
 BEGIN IF NOT (SY IN X) THEN                                                   COMP       86
  BEGIN ERROR(Y); SKIP(X+Z) END                                                COMP       87
 END (*TEST2*) ;                                                               COMP       87
                                                                               COMP       87
                                                                               COMP       87
 PROCEDURE BLOCK(FSYS: SETOFSYS; FSY: SYMBOL; FPROCP: CTP);                    COMP       87
  VAR LSY: SYMBOL; FLABP: LBP; LFSTCSP: CSP;                                   COMP       87
      TRAPSET: BOOLEAN; TRADD: ADDRRANGE; TRAPLAB: INTEGER;                    COMP       87
      LFORWCNT: INTEGER;                                                       COMP       87
                                                                               COMP       87
      PROCEDURE CHECKFORW(FCP: CTP);                                           COMP       87
        (*PRINT ERROR MESSAGE FOR FORWARD DECLARED PROCEDURE*)                 COMP       88
      BEGIN                                                                    COMP       88
       IF FCP <> NIL THEN                                                      COMP       88
        WITH FCP↑ DO                                                           COMP       88
         BEGIN                                                                 COMP       88
          IF KLASS IN [PROC,FUNC] THEN                                         COMP       88
           IF FFKIND = ACTUAL THEN                                             COMP       88
            IF PFDECL = FORWDECL THEN                                          COMP       88
             BEGIN ERROR(117); WRITELN;                                        COMP       88
```

```
            WRITELN(Ξ ***    UNDECLARED PROCEDURE: Ξ,NAME);                  COMP
            SPACES                                                           COMP
          END;                                                               COMP
        CHECKFORW(LLINK); CHECKFORW(RLINK)                                   COMP
      END                                                                    COMP
  END (*CHECKFORW*);                                                         COMP

                                                                             COMP
                                                                             COMP
FUNCTION NROFBITS(FVAL: INTEGER) : INTEGER;                                  COMP
 (*COMPUTE NR OF BITS NECESSARY TO REPRESENT 0..FVAL*)                       COMP
 VAR B: INTEGER;                                                             COMP
BEGIN B := 0;                                                                COMP
 REPEAT FVAL := FVAL DIV 2; B := B + 1                                       COMP
 UNTIL FVAL = 0;                                                             COMP
 NROFBITS := B                                                               COMP
END (*NROFBITS*);                                                            COMP

FUNCTION FULLWORDS(FSIZE: WBSIZE) : INTEGER;                                 COMP
BEGIN                                                                        COMP
 WITH FSIZE DO                                                               COMP
  IF BITS <> 0 THEN                                                          COMP
   FULLWORDS := WORDS + 1                                                    COMP
  ELSE FULLWORDS := WORDS                                                    COMP
END (*FULLWORDS*) ;                                                          COMP

FUNCTION COMPTYPES(FSP1,FSP2: STP; FPCKD: BOOLEAN) : BOOLEAN;                COMP
 (*DECIDE WHETHER STRUCT POINTED AT BY FSP1 AND FSP2 ARE COMPATIBLE;         COMP
  COMPREHENDING STRUCTURES ARE FPCKD (FALSE AT INITIAL CALL)*)               COMP
  LABEL 1;                                                                   COMP
  VAR NXT1,NXT2: CTP; DUMMYP: STP; COMP: BOOLEAN; LPCRP: PCRP;               COMP

 FUNCTION COMPLISTS(FCP1,FCP2: CTP; FSP1,FSP2: STP; FPCKD: BOOLEAN)          COMP
  : BOOLEAN;                                                                 COMP
  (*DECIDE WHETHER FIELDLISTS ARE COMPATIBLE*)                               COMP
  (*FCP1, FCP2: HEADS OF FIELDLISTS; FSP1, FSP2: POINTERS TO HEAD OF         COMP
   VARIANT CHAIN*)                                                           COMP
  VAR COMP: BOOLEAN; NXT1, NXT2: STP;                                        COMP
 BEGIN COMP := TRUE;                                                         COMP
  WHILE COMP AND (FCP1 <> NIL)AND (FCP2 <> NIL) DO                           COMP
   BEGIN COMP := COMP AND COMPTYPES(FCP1↑.IDTYPE,FCP2↑.IDTYPE,FPCKD);        COMP
    FCP1 := FCP1↑.NEXT; FCP2 := FCP2↑.NEXT                                   COMP
   END;                                                                      COMP
  COMP := COMP AND (FCP1 = FCP2);                                            COMP
  IF (FSP1 <> NIL)AND (FSP2 <> NIL) THEN                                     COMP
   BEGIN                                                                     COMP
    IF (FSP1↑.TAGFIELDP <> NIL)AND (FSP2↑.TAGFIELDP <> NIL) THEN             COMP
     COMP := COMP AND COMPTYPES(FSP1↑.TAGFIELDP↑.IDTYPE,                     COMP
       FSP2↑.TAGFIELDP↑.IDTYPE,FPCKD);                                       COMP
    NXT1 := FSP1↑.FSTVAR; NXT2 := FSP2↑.FSTVAR;                              COMP
    WHILE COMP AND (NXT1 <> NIL)AND (NXT2 <> NIL) DO                         COMP
     BEGIN COMP := COMP AND COMPLISTS(NXT1↑.FSTVARFLD,NXT2↑.FSTVARFLD,       COMP
           NXT1↑.SUBVAR,NXT2↑.SUBVAR,FPCKD);                                 COMP
      NXT1 := NXT1↑.NXTVAR; NXT2 := NXT2↑.NXTVAR                             COMP
     END;                                                                    COMP
    COMPLISTS := COMP AND (NXT1 = NXT2)                                      COMP
   END                                                                       COMP
  ELSE COMPLISTS := COMP AND (FSP1 = FSP2)                                   COMP
 END (*COMPLISTS*) ;                                                         COMP

 FUNCTION EQUALBOUNDS(FSP1,FSP2: STP) : BOOLEAN;                             COMP
  VAR LMIN1,LMIN2,LMAX1,LMAX2: INTEGER;                                      COMP
 BEGIN GETBOUNDS(FSP1,LMIN1,LMAX1);                                          COMP
  GETBOUNDS(FSP2,LMIN2,LMAX2);                                               COMP
  EQUALBOUNDS := (LMIN1 = LMIN2)AND (LMAX1 = LMAX2)                          COMP
 END (*EQUALBOUNDS*) ;                                                       COMP

BEGIN (*COMPTYPES*)                                                          COMP
 IF FSP1 = FSP2 THEN COMPTYPES := TRUE                                       COMP
 ELSE                                                                        COMP
  IF (FSP1 <> NIL)AND (FSP2 <> NIL) THEN                                     COMP
   IF FSP1↑.FORM = FSP2↑.FORM THEN                                           COMP
    CASE FSP1↑.FORM OF                                                       COMP
     SCALAR:                                                                 COMP
      IF (FSP1↑.SCALKIND = STANDARD)OR(FSP2↑.SCALKIND = STANDARD) THEN       COMP
       COMPTYPES := FALSE                                                    COMP
      ELSE                                                                   COMP
       BEGIN NXT1 := FSP1↑.FCONST; NXT2 := FSP2↑.FCONST;                     COMP
        COMP := TRUE;                                                        COMP
        WHILE COMP AND (NXT1 <> NIL)AND (NXT2 <> NIL) DO                     COMP
         BEGIN COMP := COMP AND (NXT1↑.NAME = NXT2↑.NAME);                   COMP
          NXT1 := NXT1↑.NEXT; NXT2 := NXT2↑.NEXT                             COMP
```

000013

```
          END;
        COMPTYPES := COMP AND (NXT1 = NXT2)
       END;
     SUBRANGE:
      BEGIN COMP := COMPTYPES(FSP1↑.RANGETYPE,FSP2↑.RANGETYPE,FPCKD);
       IF COMP AND FPCKD AND (FSP1↑.RANGETYPE <> REALPTR) THEN
        COMP := COMP AND EQUALBOUNDS(FSP1,FSP2);
       COMPTYPES := COMP
      END;
     POINTER:
       BEGIN LPCRP := FSTPCRP; COMP := TRUE;
        WHILE LPCRP <> NIL DO
         WITH LPCRP↑ DO
          BEGIN
           IF (FSP1 = PTR1) AND (FSP2 = PTR2) THEN GOTO 1
           ELSE
            IF (FSP1 = PTR2) AND (FSP2 = PTR1) THEN GOTO 1;
           LPCRP := NEXT
          END;
        IF FSTPCRP = NIL THEN NEW(DUMMYP);
         NEW(LPCRP);
        WITH LPCRP↑ DO
         BEGIN NEXT := FSTPCRP; PTR1 := FSP1; PTR2 := FSP2 END;
        FSTPCRP := LPCRP;
        COMP := COMPTYPES(FSP1↑.ELTYPE,FSP2↑.ELTYPE,FALSE);
        FSTPCRP := FSTPCRP↑.NEXT;
        IF FSTPCRP = NIL THEN RELEASE(DUMMYP);
1:      COMPTYPES := COMP
       END;
     POWER:
      COMPTYPES := (FSP1↑.PCKDSET = FSP2↑.PCKDSET)
             AND COMPTYPES(FSP1↑.ELSET,FSP2↑.ELSET,FSP1↑.PCKDSET);
     ARRAYS:
      BEGIN COMP := (FSP1↑.PCKDARR = FSP2↑.PCKDARR)
            AND COMPTYPES(FSP1↑.AELTYPE,FSP2↑.AELTYPE,FSP1↑.PCKDARR)
            AND COMPTYPES(FSP1↑.INXTYPE,FSP2↑.INXTYPE,FALSE);
        COMPTYPES := COMP AND EQUALBOUNDS(FSP1↑.INXTYPE,
                       FSP2↑.INXTYPE);
      END;
     RECORDS:
      COMPTYPES := (FSP1↑.PCKDREC = FSP2↑.PCKDREC)
             AND COMPLISTS(FSP1↑.FSTFLD,FSP2↑.FSTFLD,
            FSP1↑.RECVAR,FSP2↑.RECVAR,FSP1↑.PCKDREC);
     FILES:
      COMPTYPES := (FSP1↑.PCKDFIL = FSP2↑.PCKDFIL)
           AND COMPTYPES(FSP1↑.FILTYPE,FSP2↑.FILTYPE,FSP1↑.PCKDFIL)
    END (*CASE*)
   ELSE (*FSP1↑.FORM <> FSP2↑.FORM*)
    IF FSP1↑.FORM = SUBRANGE THEN
     BEGIN COMP := COMPTYPES(FSP1↑.RANGETYPE,FSP2,FPCKD);
      IF COMP AND FPCKD AND (FSP2 <> REALPTR) THEN
       COMP := COMP AND EQUALBOUNDS(FSP1,FSP2);
      COMPTYPES := COMP
     END
    ELSE
     IF FSP2↑.FORM = SUBRANGE THEN
      BEGIN COMP := COMPTYPES(FSP1,FSP2↑.RANGETYPE,FPCKD);
       IF COMP AND FPCKD AND (FSP1 <> REALPTR) THEN
        COMP := COMP AND EQUALBOUNDS(FSP1,FSP2);
       COMPTYPES := COMP
      END
     ELSE COMPTYPES := FALSE
  ELSE COMPTYPES := TRUE
END (*COMPTYPES*) ;

FUNCTION STRING(FSP: STP) : BOOLEAN;
BEGIN STRING := FALSE;
 IF FSP <> NIL THEN
  WITH FSP↑ DO
   IF FORM = ARRAYS THEN
    IF COMPTYPES(AELTYPE,CHARPTR,FALSE) THEN STRING := PCKDARR
END (*STRING*) ;

PROCEDURE STRINGTYPE(VAR FSP: STP);
 (*ENTER TYPE OF STRINGCONST (PACKED ARRAY [1..LGTH] OF CHAR) INTO
  STRUCTURE TABLE*)
 VAR LSP,LSP1: STP;
BEGIN NEW(LSP,SUBRANGE);
 WITH LSP↑ DO
  BEGIN FORM := SUBRANGE; RANGETYPE := INTPTR;
   MIN.IVAL := 1; MAX.IVAL := LGTH ; FTYPE := FALSE;
```

```
    WITH SIZE DO
      BEGIN WORDS := 0; BITS := NROFBITS(LGTH) END
   END;
  NEW(LSP1,ARRAYS,TRUE,TRUE);
  WITH LSP1^ DO
   BEGIN FORM := ARRAYS;
    AELTYPE := CHARPTR; INXTYPE := LSP;
    PCKDARR := TRUE; PARTWORDELS := TRUE;
    ELSPERWORD := ALFALENG; FTYPE := FALSE;
    WITH SIZE DO
     BEGIN WORDS := LGTH DIV ALFALENG;
      BITS := LGTH MOD ALFALENG*6
     END
   END;
  FSP := LSP1
 END (*STRINGTYPE*) ;

 PROCEDURE CONSTANT(FSYS: SETOFSYS; VAR FSP: STP; VAR FVALU: VALU);
  VAR LSP: STP; LCP: CTP; SIGN: (NONE,POS,NEG); LCSP: CSP;
 BEGIN LSP := NIL; FVALU.IVAL := 0;
  TEST2(CONSTBEGSYS,50,FSYS);
  IF SY IN CONSTBEGSYS THEN
   BEGIN
    IF SY = CHARCONST THEN
     BEGIN LSP := CHARPTR; FVALU.IVAL := IVAL; INSYMBOL END
    ELSE
     IF SY = STRINGCONST THEN
      BEGIN STRINGTYPE(LSP);
       FVALU.VALP := CONSTP;
       INSYMBOL
      END
    ELSE
     BEGIN
      SIGN := NONE;
      IF OP IN [PLUS,MINUS] THEN
       BEGIN IF OP = PLUS THEN SIGN := POS ELSE SIGN := NEG;
        INSYMBOL
       END;
      IF SY = IDENT THEN
       BEGIN SEARCHID([KONST],LCP);
        WITH LCP^ DO
         BEGIN LSP := IDTYPE; FVALU := VALUES END;
        IF SIGN <> NONE THEN
         IF LSP = INTPTR THEN
          BEGIN IF SIGN = NEG THEN FVALU.IVAL := -FVALU.IVAL END
         ELSE
          IF LSP = REALPTR THEN
           BEGIN
            IF SIGN = NEG THEN FVALU.RVAL := -FVALU.RVAL
           END
         ELSE ERROR(105);
        INSYMBOL;
       END
      ELSE
       IF SY = INTCONST THEN
        BEGIN IF SIGN = NEG THEN IVAL := -IVAL;
         LSP := INTPTR; FVALU.IVAL := IVAL; INSYMBOL
        END
       ELSE
        IF SY = REALCONST THEN
         BEGIN IF SIGN = NEG THEN RVAL := -RVAL;
          LSP := REALPTR; FVALU.RVAL := RVAL; INSYMBOL
         END
        ELSE
         BEGIN ERROR(106); SKIP(FSYS) END
     END;
    TEST2(FSYS,6,[])
   END;
  FSP := LSP
 END (*CONSTANT*) ;


 PROCEDURE TYP(FSYS: SETOFSYS; VAR FSP: STP);
  VAR LSP,LSP1,LSP2: STP; OLDTOP: DISPRANGE; LCP: CTP;
    LSIZE,DISPL: WBSIZE; LMIN,LMAX,LRL: INTEGER;
     T,T1: INTEGER; PACKFLAG,SEGFLAG,LFILTYP,EXITLOOP: BOOLEAN;

  PROCEDURE SIMPLETYPE(FSYS: SETOFSYS; VAR FSP: STP);
   VAR LSP,LSP1: STP; LCP,LCP1: CTP; TTOP: DISPRANGE;
     LVAL: INTEGER; LVALU: VALU;
```

```
  PROCEDURE SUBRANGES(FSP: STP; FVALU: VALU);                                    COMP      113
   (*PROCESS SUBRANGE TYPE*)                                                     COMP      113
   VAR LOW,HIGH: INTEGER;                                                        COMP      113
  BEGIN NEW(LSP,SUBRANGE);                                                       COMP      113
   WITH LSP↑ DO                                                    000015        COMP      113
    BEGIN RANGETYPE := FSP; FORM := SUBRANGE;                                    COMP      113
     MIN := FVALU; FTYPE := FALSE                                                COMP      113
    END;                                                                         COMP      113
   TEST1(COLON,5);                                                               COMP      114
   CONSTANT(FSYS,LSP1,LVALU);                                                    COMP      114
   WITH LSP↑ DO                                                                  COMP      114
    BEGIN MAX := LVALU;                                                          COMP      114
     WITH SIZE DO                                                                COMP      114
      BEGIN WORDS := 1; BITS := 0 END;                                           COMP      114
     IF NOT COMPTYPES(FSP,LSP1,FALSE) THEN ERROR(107)                            COMP      114
     ELSE                                                                        COMP      114
      WITH SIZE DO                                                               COMP      114
       IF FSP = REALPTR THEN                                                     COMP      114
        BEGIN IF MIN.RVAL >MAX.RVAL THEN ERROR(102)                              COMP      115
        END                                                                      COMP      115
       ELSE                                                                      COMP      115
        IF STRING(FSP) THEN                                                      COMP      115
         BEGIN ERROR(148); RANGETYPE := NIL                                      COMP      115
         END                                                                     COMP      115
        ELSE                                                                     COMP      115
         BEGIN LOW := MIN.IVAL; HIGH := MAX.IVAL;                                COMP      115
          IF LOW > HIGH THEN ERROR(102);                                         COMP      115
          WORDS := 0;                                                            COMP      115
          IF ABS(LOW) < ABS(HIGH) THEN                                           COMP      116
           BITS := NROFBITS(ABS(HIGH))                                           COMP      116
          ELSE BITS := NROFBITS(ABS(LOW));                                       COMP      116
          IF LOW < 0 THEN BITS := BITS + 1                                       COMP      116
         END                                                                     COMP      116
    END                                                                          COMP      116
  END (*SUBRANGES*);                                                             COMP      116

BEGIN (*SIMPLETYPE*)                                                             COMP      116
 TEST2(SIMPTYPEBEGSYS,1,FSYS);                                                   COMP      116
 IF SY IN SIMPTYPEBEGSYS THEN                                                    COMP      117
  BEGIN                                                                          COMP      117
   IF SY = LPARENT THEN                                                          COMP      117
    BEGIN TTOP := TOP;    (*DECL. CONSTS LOCAL TO INNERMOST BLOCK*)              COMP      117
     WHILE DISPLAY[TOP].OCCUR <> BLCK DO TOP := TOP - 1;                         COMP      117
     NEW(LSP,SCALAR,DECLARED);                                                   COMP      117
     WITH LSP↑ DO                                                                COMP      117
      BEGIN FORM := SCALAR; SCALKIND := DECLARED; FTYPE := FALSE;                COMP      117
       FCONST := NIL                                                             COMP      117
      END;                                                                       COMP      117
     LCP1 := NIL; LVAL := -1;                                                    COMP      118
     REPEAT INSYMBOL;                                                            COMP      118
      IF SY = IDENT THEN                                                         COMP      118
       BEGIN NEW(LCP,KONST); LVAL := LVAL + 1;                                   COMP      118
        WITH LCP↑ DO                                                             COMP      118
         BEGIN NAME := ID; IDTYPE := LSP; NEXT := LCP1;                          COMP      118
          VALUES.IVAL := LVAL; KLASS := KONST                                    COMP      118
         END;                                                                    COMP      118
        ENTERID(LCP);                                                            COMP      118
        LCP1 := LCP; INSYMBOL                                                    COMP      118
       END                                                                       COMP      119
      ELSE ERROR(2);                                                             COMP      119
      TEST2(FSYS+[COMMA,RPARENT],6,[])                                           COMP      119
     UNTIL SY <> COMMA;                                                          COMP      119
     WITH LSP↑, SIZE DO                                                          COMP      119
      BEGIN FCONST := LCP1;                                                      COMP      119
       WORDS := 0; BITS := NROFBITS(LVAL)                                        COMP      119
      END;                                                                       COMP      119
     TOP := TTOP;                                                                COMP      119
     TEST1(RPARENT,4)                                                            COMP      119
    END                                                                          COMP      120
   ELSE                                                                          COMP      120
    BEGIN                                                                        COMP      120
     IF SY = IDENT THEN                                                          COMP      120
      BEGIN SEARCHID([TYPES,KONST],LCP);                                         COMP      120
       INSYMBOL;                                                                 COMP      120
       WITH LCP↑ DO                                                              COMP      120
        IF KLASS = KONST THEN SUBRANGES(IDTYPE,VALUES)                           COMP      120
        ELSE                                                                     COMP      120
         LSP := IDTYPE                                                           COMP      120
      END (*SY = IDENT*)                                                         COMP      121
     ELSE                                                                        COMP      121
      BEGIN CONSTANT(FSYS+[COLON],LSP1,LVALU);                                   COMP      121
       SUBRANGES(LSP1,LVALU)                                                     COMP      121
```

```
      END;                                                                  COMP     121
    END;                                                                    COMP     121
   FSP := LSP;                                                              COMP     121
   TEST2(FSYS,6,[])                                          000016         COMP     121
  END                                                                       COMP     121
   ELSE FSP := NIL                                                          COMP     121
END (*SIMPLETYPE*) ;                                                        COMP     122
                                                                            COMP     122
PROCEDURE FIELDLIST(FSYS: SETOFSYS; VAR FRECVAR: STP;                       COMP     122
          VAR FFSTFLD: CTP; VAR FTYP: BOOLEAN);                             COMP     122
 VAR LASTFLD, LCP,LCP1,NXT,NXT1: CTP; LSP,LSP1,LSP2,LSP3,LSP4: STP;         COMP     122
   MINSIZE,MAXSIZE: WBSIZE; LVALU: VALU;                                    COMP     122
   TAGFLAG,LFILTYP,EXITLOOP: BOOLEAN;                                       COMP     122
                                                                            COMP     122
 PROCEDURE FIELDADDRESS(FCP: CTP; FSIZE: WBSIZE);                           COMP     122
  (*COMPUTE ADDRESS OF FCP↑ ACCORDING TO ITS SIZE *)                        COMP     122
                                                                            COMP     123
  PROCEDURE ADJUST;                                                         COMP     123
   (*ADJUST LASTFLD*)                                                       COMP     123
  BEGIN                                                                     COMP     123
   IF NOT TAGFLAG THEN                                                      COMP     123
    WITH LASTFLD↑ DO                                                        COMP     123
     IF IDTYPE <> NIL THEN                                                  COMP     123
      IF IDTYPE↑.FORM <= POWER THEN                                         COMP     123
       IF BITADDR = 0 THEN PCKDFLD := FALSE                                 COMP     123
       ELSE                                                                 COMP     123
        BITADDR := WORDSIZE - IDTYPE↑.SIZE.BITS;                            COMP     124
   WITH DISPL DO                                                            COMP     124
    BEGIN WORDS := WORDS + 1; BITS := 0 END                                 COMP     124
  END (*ADJUST*);                                                           COMP     124
                                                                            COMP     124
 BEGIN (*FIELDADDRESS*)                                                     COMP     124
  WITH DISPL, FCP↑ DO                                                       COMP     124
   IF PACKFLAG AND (FSIZE.WORDS = 0) THEN                                   COMP     124
    BEGIN IF BITS + FSIZE.BITS > WORDSIZE THEN ADJUST;                      COMP     124
     FLDADDR := WORDS; PCKDFLD := TRUE;                                     COMP     124
     BITADDR := BITS;                                                       COMP     125
     IF BITS + FSIZE.BITS = WORDSIZE THEN                                   COMP     125
      BEGIN WORDS := WORDS + 1; BITS := 0 END                               COMP     125
     ELSE BITS := BITS + FSIZE.BITS                                         COMP     125
    END                                                                     COMP     125
   ELSE                                                                     COMP     125
    BEGIN IF BITS <> 0 THEN ADJUST;                                         COMP     125
     FLDADDR := WORDS; PCKDFLD := FALSE;                                    COMP     125
     WORDS := WORDS + FULLWORDS(FSIZE)                                      COMP     125
    END;                                                                    COMP     125
 END (*FIELDADDRESS*) ;                                                     COMP     126
                                                                            COMP     126
BEGIN (*FIELDLIST*) NXT1 := NIL; LSP := NIL;                                COMP     126
 TAGFLAG := TRUE; FTYP := FALSE;                                            COMP     126
 TEST2(FSYS+[IDENT,CASESY],19,[]);                                          COMP     126
 WHILE SY = IDENT DO                                                        COMP     126
  BEGIN NXT := NXT1;                                                        COMP     126
   (*LOOP UNTIL SY <> COMMA:*)                                              COMP     126
   REPEAT                                                                   COMP     126
    IF SY = IDENT THEN                                                      COMP     126
     BEGIN NEW(LCP,FIELD);                                                  COMP     127
      WITH LCP↑ DO                                                          COMP     127
       BEGIN NAME := ID; IDTYPE := NIL; NEXT := NXT;                        COMP     127
        KLASS := FIELD                                                      COMP     127
       END;                                                                 COMP     127
      NXT := LCP;                                                           COMP     127
      ENTERID(LCP);                                                         COMP     127
      INSYMBOL                                                              COMP     127
     END                                                                    COMP     127
    ELSE ERROR(2);                                                          COMP     127
    TEST2([COMMA,COLON],6,FSYS+[SEMICOLON,CASESY]);                         COMP     128
    EXITLOOP := SY <> COMMA;                                                COMP     128
    IF NOT EXITLOOP THEN INSYMBOL                                           COMP     128
   UNTIL EXITLOOP;                                                          COMP     128
   TEST1(COLON,5);                                                          COMP     128
   TYP(FSYS+[CASESY,SEMICOLON],LSP);                                        COMP     128
   WHILE NXT <> NXT1 DO                                                     COMP     128
    WITH NXT↑ DO                                                            COMP     128
     BEGIN IDTYPE := LSP;                                                   COMP     128
      IF LSP <> NIL THEN                                                    COMP     128
       BEGIN FTYP := FTYP OR LSP↑.FTYPE;                                    COMP     129
        FIELDADDRESS(NXT,LSP↑.SIZE)                                         COMP     129
       END                                                                  COMP     129
      ELSE                                                                  COMP     129
       BEGIN FLDADDR := DISPL.WORDS; PCKDFLD := FALSE END;                  COMP     129
```

```
      TAGFLAG := FALSE; LASTFLD := NXT; NXT := NEXT                          COMP     129
    END;                                                                     COMP     129
  NXT1 := LCP;                                                               COMP     129
  IF SY = SEMICOLON THEN                                    000017           COMP     129
    BEGIN INSYMBOL;                                                          COMP     129
     TEST2(FSYS+[IDENT,CASESY],19,[]);                                       COMP     130
    END                                                                      COMP     130
 END (*WHILE*);                                                              COMP     130
NXT := NIL;                                                                  COMP     130
WHILE NXT1 <> NIL DO                                                         COMP     130
 WITH NXT1^ DO                                                               COMP     130
  BEGIN LCP := NEXT; NEXT := NXT; NXT := NXT1; NXT1 := LCP END;              COMP     130
FFSTFLD := NXT;                                                              COMP     130
IF SY = CASESY THEN                                                          COMP     130
 BEGIN NEW(LSP,TAGFIELD);                                                    COMP     130
  WITH LSP^ DO                                                               COMP     131
   BEGIN TAGFIELDP := NIL; FSTVAR := NIL; FORM := TAGFIELD;                  COMP     131
    FTYPE := FALSE                                                           COMP     131
   END;                                                                      COMP     131
  FRECVAR := LSP;                                                            COMP     131
  INSYMBOL;                                                                  COMP     131
  IF SY = IDENT THEN                                                         COMP     131
   BEGIN PRTERR := FALSE; SEARCHID([TYPES],LCP1); PRTERR := TRUE;            COMP     131
    NEW(LCP,FIELD);                                                          COMP     131
    WITH LCP^ DO                                                             COMP     131
     BEGIN IDTYPE := NIL; KLASS := FIELD; NEXT := NIL END;                   COMP     132
    IF LCP1 = NIL THEN   (*EXPLICITE TAGFIELD*)                              COMP     132
     BEGIN LCP^.NAME := ID; ENTERID(LCP);                                    COMP     132
      INSYMBOL;                                                              COMP     132
      TEST1(COLON,5);                                                        COMP     132
      IF SY = IDENT THEN SEARCHID([TYPES],LCP1)                              COMP     132
      ELSE                                                                   COMP     132
       BEGIN ERROR(2); SKIP(FSYS+[OFSY,LPARENT]);                            COMP     132
        LCP1 := NIL                                                          COMP     132
       END                                                                   COMP     132
     END                                                                     COMP     133
    ELSE LCP^.NAME := Ξ           Ξ;                                        COMP     133
    IF LCP1 <> NIL THEN                                                      COMP     133
     BEGIN LSP1 := LCP1^.IDTYPE;                                             COMP     133
      IF LSP1 <> NIL THEN                                                    COMP     133
       BEGIN                                                                 COMP     133
        IF LSP1^.FORM <= SUBRANGE THEN                                       COMP     133
         BEGIN                                                               COMP     133
          IF COMPTYPES(REALPTR,LSP1,FALSE) THEN ERROR(109)                   COMP     133
          ELSE                                                               COMP     133
           BEGIN LSP^.TAGFIELDP := LCP;                                      COMP     134
            WITH LCP^ DO                                                     COMP     134
             BEGIN IDTYPE := LSP1;                                           COMP     134
              IF NAME <> Ξ           Ξ THEN                                 COMP     134
               FIELDADDRESS(LCP,LSP1^.SIZE)                                  COMP     134
             END                                                             COMP     134
           END                                                               COMP     134
         END                                                                 COMP     134
        ELSE ERROR(110)                                                      COMP     134
       END;                                                                  COMP     134
      INSYMBOL                                                               COMP     135
     END                                                                     COMP     135
   END                                                                       COMP     135
  ELSE                                                                       COMP     135
   BEGIN ERROR(2); SKIP(FSYS+[OFSY,LPARENT]) END;                            COMP     135
  LSP^.SIZE := DISPL;                                                        COMP     135
  TEST1(OFSY,8);                                                             COMP     135
  LSP1 := NIL; MINSIZE := DISPL; MAXSIZE := DISPL;                           COMP     135
  (*LOOP UNTIL SY <> SEMICOLON:*)                                            COMP     135
  REPEAT                                                                     COMP     135
   IF NOT (SY IN FSYS+[SEMICOLON]) THEN                                      COMP     136
   BEGIN LSP2 := NIL;                                                        COMP     136
   (*LOOP UNTIL SY <> SEMICOLON:*)                                           COMP     136
   REPEAT CONSTANT(FSYS+[COMMA,COLON,LPARENT],LSP3,LVALU);                   COMP     136
    IF LSP^.TAGFIELDP <> NIL THEN                                            COMP     136
     IF NOT COMPTYPES(LSP^.TAGFIELDP^.IDTYPE,LSP3,FALSE) THEN                COMP     136
      ERROR(111);                                                            COMP     136
    NEW(LSP3,VARIANT);                                                       COMP     136
    WITH LSP3^ DO                                                            COMP     136
     BEGIN NXTVAR := LSP1; SUBVAR := LSP2; VARVAL := LVALU;                  COMP     136
      FORM := VARIANT; FTYPE := FALSE                                        COMP     137
     END;                                                                    COMP     137
    LSP4:=LSP1;                                                              COMP     137
    WHILE LSP4 <> NIL DO                                                     COMP     137
     WITH LSP4^ DO                                                           COMP     137
      BEGIN IF VARVAL.IVAL = LVALU.IVAL THEN ERROR(178);                     COMP     137
```

```
                LSP4:=NXTVAR                                                    COMP      137
          END;                                                                  COMP      137
        LSP1 := LSP3; LSP2 := LSP3;                                             COMP      137
        EXITLOOP := SY <> COMMA;                          000018                COMP      137
        IF NOT EXITLOOP THEN INSYMBOL                                           COMP      138
      UNTIL EXITLOOP;                                                           COMP      138
      TEST1(COLON,5);                                                           COMP      138
      TEST1(LPARENT,9);                                                         COMP      138
      FIELDLIST(FSYS+[RPARENT,SEMICOLON],LSP2,LCP,LFILTYP);                     COMP      138
      IF LFILTYP THEN ERROR(108);                                               COMP      138
      FTYP := FTYP OR LFILTYP;                                                  COMP      138
      IF (DISPL.WORDS > MAXSIZE.WORDS)OR                                        COMP      138
       (DISPL.WORDS = MAXSIZE.WORDS)AND (DISPL.BITS > MAXSIZE.BITS)             COMP      138
       THEN MAXSIZE := DISPL;                                                   COMP      138
      WHILE LSP3 <> NIL DO                                                      COMP      139
       WITH LSP3↑ DO                                                            COMP      139
        BEGIN LSP4 := SUBVAR; SUBVAR := LSP2;                                   COMP      139
         SIZE := DISPL; FSTVARFLD := LCP;                                       COMP      139
         LSP3 := LSP4                                                           COMP      139
        END;                                                                    COMP      139
      IF SY = RPARENT THEN                                                      COMP      139
       BEGIN INSYMBOL;                                                          COMP      139
        TEST2(FSYS+[SEMICOLON],6,[])                                            COMP      139
       END                                                                      COMP      139
      ELSE ERROR(4);                                                            COMP      140
      END (*NOT (SY IN ...*) ;                                                  COMP      140
      EXITLOOP := SY <> SEMICOLON;                                              COMP      140
      IF NOT EXITLOOP THEN                                                      COMP      140
       BEGIN DISPL := MINSIZE; INSYMBOL END                                     COMP      140
    UNTIL EXITLOOP;                                                             COMP      140
    DISPL := MAXSIZE;                                                           COMP      140
    LSP↑.FSTVAR := LSP1;                                                        COMP      140
   END                                                                          COMP      140
  ELSE                                                                          COMP      140
   FRECVAR := NIL                                                               COMP      141
 END (*FIELDLIST*) ;                                                            COMP      141
                                                                                COMP      141
BEGIN (*TYP*) LSP := NIL;                                                       COMP      141
 TEST2(TYPEBEGSYS,10,FSYS);                                                     COMP      141
 IF SY IN TYPEBEGSYS THEN                                                       COMP      141
  BEGIN                                                                         COMP      141
   IF SY IN SIMPTYPEBEGSYS THEN SIMPLETYPE(FSYS,LSP)                            COMP      141
   ELSE                                                                         COMP      141
(*↑*)                                                                           COMP      141
    IF SY = ARROW THEN                                                          COMP      142
     BEGIN NEW(LSP,POINTER);                                                    COMP      142
      WITH LSP↑ DO                                                              COMP      142
       BEGIN ELTYPE := NIL; FORM := POINTER; FTYPE := FALSE;                    COMP      142
        WITH SIZE DO                                                            COMP      142
         BEGIN WORDS := 0; BITS := NROFBITS(MAXADDR) END                        COMP      142
       END;                                                                     COMP      142
      INSYMBOL;                                                                 COMP      142
      IF SY = IDENT THEN                                                        COMP      142
       BEGIN PRTERR := FALSE;   (*NO ERROR IF SEARCH NOT SUCCESSFUL*)           COMP      142
        SEARCHID([TYPES],LCP); PRTERR := TRUE;                                  COMP      143
        IF LCP = NIL THEN   (*FORWARD REFERENCED TYPE ID*)                      COMP      143
         BEGIN NEW(LCP,TYPES);                                                  COMP      143
          WITH LCP↑ DO                                                          COMP      143
           BEGIN NAME := ID; IDTYPE := LSP; KLASS := TYPES;                     COMP      143
            NEXT := FWPTR                                                       COMP      143
           END;                                                                 COMP      143
          FWPTR := LCP                                                          COMP      143
         END                                                                    COMP      143
        ELSE                                                                    COMP      143
         BEGIN                                                                  COMP      144
          IF LCP↑.IDTYPE <> NIL THEN                                            COMP      144
           IF LCP↑.IDTYPE↑.FTYPE THEN ERROR(108)                                COMP      144
           ELSE LSP↑.ELTYPE := LCP↑.IDTYPE                                      COMP      144
         END;                                                                   COMP      144
        INSYMBOL;                                                               COMP      144
       END                                                                      COMP      144
      ELSE ERROR(2);                                                            COMP      144
     END                                                                        COMP      144
    ELSE                                                                        COMP      144
     BEGIN                                                                      COMP      145
      IF SY = PACKEDSY THEN                                                     COMP      145
       BEGIN PACKFLAG := TRUE; INSYMBOL END                                     COMP      145
      ELSE PACKFLAG := FALSE;                                                   COMP      145
      IF SY = SEGMENTEDSY THEN                                                  COMP      145
       BEGIN SEGFLAG := TRUE; INSYMBOL END                                      COMP      145
      ELSE SEGFLAG := FALSE;                                                    COMP      145
```

```
      TEST2(TYPEDELS,10,FSYS);                                                  COMP     145
      IF (SY <> FILESY)AND SEGFLAG THEN ERROR(57);                              COMP     145
(*ARRAY*)                                                                       COMP     145
      IF SY = ARRAYSY THEN                                    000019            COMP     146
       BEGIN INSYMBOL;                                                          COMP     146
        TEST1(LBRACK,11);                                                       COMP     146
        LSP1 := NIL;                                                            COMP     146
        (*LOOP UNTIL SY <> COMMA:*)                                             COMP     146
        REPEAT NEW(LSP,ARRAYS);                                                 COMP     146
         WITH LSP↑ DO                                                           COMP     146
          BEGIN AELTYPE := LSP1; INXTYPE := NIL;                                COMP     146
           PCKDARR := PACKFLAG; FORM := ARRAYS;                                 COMP     146
           FTYPE := FALSE                                                       COMP     146
          END;                                                                  COMP     147
         LSP1 := LSP;                                                           COMP     147
         SIMPLETYPE(FSYS+[COMMA,RBRACK,OFSY],LSP2);                             COMP     147
         IF LSP2 <> NIL THEN                                                    COMP     147
          IF LSP2↑.FORM <= SUBRANGE THEN                                        COMP     147
           IF COMPTYPES(LSP2,REALPTR,FALSE) THEN ERROR(112)                     COMP     147
           ELSE IF LSP2 = INTPTR THEN ERROR(149)                                COMP     147
           ELSE LSP↑.INXTYPE := LSP2                                            COMP     147
          ELSE ERROR(113);                                                      COMP     147
         EXITLOOP := SY <> COMMA;                                               COMP     147
         IF NOT EXITLOOP THEN INSYMBOL                                          COMP     148
        UNTIL EXITLOOP;                                                         COMP     148
        TEST1(RBRACK,12);                                                       COMP     148
        TEST1(OFSY,8);                                                          COMP     148
        TYP(FSYS,LSP);                                                          COMP     148
        (*REVERSE POINTERS, COMPUTE SIZE, SET PARTWORDELS AND                   COMP     148
          ELSPERWORD*)                                                          COMP     148
        IF LSP <> NIL THEN                                                      COMP     148
         BEGIN LSIZE := LSP↑.SIZE;                                              COMP     148
          REPEAT                                                                COMP     148
           WITH LSP1↑ DO                                                        COMP     149
            BEGIN LSP2 := AELTYPE; AELTYPE := LSP;                              COMP     149
             FTYPE := LSP↑.FTYPE;                                               COMP     149
             IF INXTYPE <> NIL THEN                                             COMP     149
              BEGIN GETBOUNDS(INXTYPE,LMIN,LMAX);                               COMP     149
               IF (LSIZE.WORDS > 0)OR NOT PACKFLAG THEN                         COMP     149
                BEGIN LSIZE.WORDS := FULLWORDS(LSIZE)*                          COMP     149
                           (LMAX-LMIN+1);                                       COMP     149
                 LSIZE.BITS := 0; PARTWORDELS := FALSE                          COMP     149
                END                                                             COMP     149
               ELSE                                                             COMP     150
                BEGIN                                                           COMP     150
                 IF LSIZE.BITS > 0 THEN                                         COMP     150
                  T := WORDSIZE DIV LSIZE.BITS                                  COMP     150
                 ELSE T := 1;                                                   COMP     150
                 T1 := (LMAX - LMIN + 1) MOD T;                                 COMP     150
                 IF (T1 = 0)AND(T*LSIZE.BITS < WORDSIZE) THEN T1 := T;          COMP     150
                 LSIZE.WORDS := (LMAX - LMIN + 1 - T1) DIV T;                   COMP     150
                 LSIZE.BITS := T1*LSIZE.BITS;                                   COMP     150
                 IF T > 1 THEN                                                  COMP     150
                  BEGIN PARTWORDELS := TRUE;                                    COMP     151
                   ELSPERWORD := T                                              COMP     151
                  END                                                           COMP     151
                 ELSE PARTWORDELS := FALSE                                      COMP     151
                END                                                             COMP     151
              END;                                                              COMP     151
             SIZE := LSIZE                                                      COMP     151
            END (*WITH LSP1↑*) ;                                                COMP     151
           LSP := LSP1; LSP1 := LSP2                                            COMP     151
          UNTIL LSP1 = NIL                                                      COMP     151
         END (*LSP <> NIL*)                                                     COMP     152
       END                                                                      COMP     152
      ELSE                                                                      COMP     152
(*RECORD*)                                                                      COMP     152
       IF SY = RECORDSY THEN                                                    COMP     152
        BEGIN INSYMBOL;                                                         COMP     152
         OLDTOP := TOP;                                                         COMP     152
         IF TOP < DISPLIMIT THEN                                                COMP     152
          BEGIN TOP := TOP + 1;                                                 COMP     152
           WITH DISPLAY[TOP] DO                                                 COMP     152
            BEGIN FNAME := NIL; OCCUR := REC END                                COMP     153
          END                                                                   COMP     153
         ELSE ERROR(250);                                                       COMP     153
         WITH DISPL DO                                                          COMP     153
          BEGIN WORDS := 0; BITS := 0 END;                                      COMP     153
         FIELDLIST(FSYS-[SEMICOLON]+[ENDSY],LSP1,LCP,LFILTYP);                  COMP     153
         NEW(LSP,RECORDS);                                                      COMP     153
         WITH LSP↑ DO                                                           COMP     153
```

```
          BEGIN FIELDS := DISPLAY[TOP].FNAME; FTYPE := LFILTYP;                COMP     151
            FSTFLD := LCP; RECVAR := LSP1; SIZE := DISPL;                      COMP     153
            PCKDREC := PACKFLAG; FORM := RECORDS                               COMP     154
          END;                                                                 COMP     154
          TOP := OLDTOP;                                                       COMP     154
          TEST1(ENDSY,13)                                                      COMP     154
        END                                                    000020          COMP     154
      ELSE                                                                     COMP     154
(*SET*)                                                                        COMP     154
        IF SY = SETSY THEN                                                     COMP     154
          BEGIN INSYMBOL;                                                      COMP     154
            TEST1(OFSY,8);                                                     COMP     154
            SIMPLETYPE(FSYS,LSP1);                                             COMP     155
            IF LSP1 <> NIL THEN                                                COMP     155
              IF LSP1↑.FORM > SUBRANGE THEN                                    COMP     155
                BEGIN ERROR(115); LSP1 := NIL END                              COMP     155
              ELSE                                                             COMP     155
                IF LSP1 = REALPTR THEN ERROR(114)                              COMP     155
                ELSE                                                           COMP     155
                  IF LSP1 = INTPTR THEN                                        COMP     155
                    ERROR(169)                                                 COMP     155
                  ELSE                                                         COMP     155
                    BEGIN GETBOUNDS(LSP1,LMIN,LMAX);                           COMP     156
                      IF (LMIN < 0)OR (LMAX > 58) THEN ERROR(169);             COMP     156
                      (*IMPLEMENTATION RESTRICTION TO ONE WORD SETS*)           COMP     156
                      NEW(LSP,POWER);                                          COMP     156
                      WITH LSP↑, SIZE DO                                       COMP     156
                        BEGIN ELSET := LSP1; PCKDSET := PACKFLAG;              COMP     156
                          FORM := POWER; FTYPE := FALSE;                       COMP     156
                          IF LMAX >= 58 THEN                                   COMP     156
                            BEGIN WORDS := 1; BITS := 0 END                    COMP     156
                          ELSE                                                 COMP     156
                            BEGIN WORDS := 0; BITS := LMAX + 1 END             COMP     157
                        END                                                    COMP     157
                    END                                                        COMP     157
        END                                                                    COMP     157
      ELSE                                                                     COMP     157
(*FILE*)  IF SY = FILESY THEN                                                  COMP     157
          BEGIN INSYMBOL;                                                      COMP     157
            TEST1(OFSY,8);                                                     COMP     157
            TYP(FSYS,LSP1);                                                    COMP     157
            (*COMPUTE MACHINE AND IMPLEMENTATION DEPENDANT                     COMP     157
            FILE SIZE*)                                                        COMP     158
            IF LSP1 <> NIL THEN                                                COMP     158
              BEGIN LRL := FULLWORDS(LSP1↑.SIZE);                              COMP     158
                IF LRL <= 1 THEN LRL := 1                                      COMP     158
              END                                                              COMP     158
            ELSE LRL := 1;                                                     COMP     158
            T1 := ((128*BUFFAC + LRL - 1) DIV LRL + 1)*LRL;                    COMP     158
            NEW(LSP,FILES);                                                    COMP     158
            WITH LSP↑ DO                                                       COMP     158
              BEGIN FILTYPE := LSP1; FORM := FILES; FTYPE := TRUE;             COMP     158
                SEGFILE := SEGFLAG;                                            COMP     159
                TEXTFILE := (*PACKFLAG AND *)COMPTYPES(CHARPTR,LSP1,           COMP     159
                              FALSE);                                          COMP     159
                IF TEXTFILE THEN PCKDFIL := TRUE                               COMP     159
                ELSE PCKDFIL := PACKFLAG;                                      COMP     159
                WITH SIZE DO                                                   COMP     159
                  BEGIN                                                        COMP     159
                    IF TEXTFILE THEN WORDS := T1 + CHEFETSZ                    COMP     159
                    ELSE WORDS := T1 + EFETSZ;                                 COMP     159
                    BITS := 0                                                  COMP     159
                  END                                                          COMP     160
              END;                                                             COMP     160
            IF LSP1 <> NIL THEN                                                COMP     160
              IF LSP1↑.FTYPE THEN                                              COMP     160
                BEGIN ERROR(108); LSP↑.FILTYPE := NIL END;                     COMP     160
          END;                                                                 COMP     160
      END;                                                                     COMP     160
    TEST2(FSYS,6,[])                                                           COMP     160
  END;                                                                         COMP     160
  FSP := LSP                                                                   COMP     160
END (*TYP*) ;                                                                  COMP     161

PROCEDURE LABELDECLARATION;                                                    COMP     161
  LABEL 1;                                                                     COMP     161
  VAR LLP: LBP; EXITLOOP: BOOLEAN;                                             COMP     161
BEGIN                                                                          COMP     161
  (*LOOP UNTIL SY <> COMMA:*)                                                  COMP     161
  REPEAT                                                                       COMP     161
    IF SY = INTCONST THEN                                                      COMP     161
```

```
    BEGIN LLP := FSTLABP;                                                    COMP     161
     WHILE LLP <> FLABP DO                                                   COMP     162
      IF LLP↑.LABVAL = IVAL THEN                          000021             COMP     162
       BEGIN ERROR(166); GOTO 1 END                                          COMP     162
      ELSE LLP := LLP↑.NEXTLAB;                                              COMP     162
     NEW(LLP);                                                               COMP     162
     WITH LLP↑ DO                                                            COMP     162
      BEGIN LABVAL := IVAL; DEFINED := FALSE; NEXTLAB := FSTLABP;            COMP     162
       LABLEV := LEVEL; LCNT := 0; FSTOCC := NIL                             COMP     162
      END;                                                                   COMP     162
     FSTLABP := LLP;                                                         COMP     162
1:   INSYMBOL;                                                              COMP     163
     IF OP = MUL THEN                                                        COMP     163
      BEGIN IF TRAPSET THEN ERROR(178);                                      COMP     163
       TRAPSET := TRUE; TRAPLAB := LLP↑.LABVAL;                              COMP     163
       TRADD := LC; LC := LC + 1;                                            COMP     163
       INSYMBOL                                                              COMP     163
      END                                                                    COMP     163
    END                                                                      COMP     163
   ELSE ERROR(15);                                                           COMP     163
   TEST2(FSYS+[COMMA,SEMICOLON],6,[]);                                       COMP     163
   EXITLOOP := SY <> COMMA;                                                  COMP     164
   IF NOT EXITLOOP THEN INSYMBOL                                             COMP     164
  UNTIL EXITLOOP;                                                            COMP     164
  TEST1(SEMICOLON,14)                                                        COMP     164
END (*LABELDECLARATION*) ;                                                   COMP     164

PROCEDURE CONSTDECLARATION;                                                  COMP     164
 VAR LCP: CTP; LSP: STP; LVALU: VALU;                                        COMP     164
BEGIN                                                                        COMP     164
 IF SY <> IDENT THEN                                                         COMP     164
  BEGIN ERROR(2); SKIP(FSYS+[IDENT]) END;                                    COMP     165
 WHILE SY = IDENT DO                                                         COMP     165
  BEGIN NEW(LCP,KONST);                                                      COMP     165
   WITH LCP↑ DO                                                              COMP     165
    BEGIN NAME := ID; IDTYPE := NIL; NEXT := NIL;                            COMP     165
     KLASS := KONST                                                          COMP     165
    END;                                                                     COMP     165
   INSYMBOL;                                                                 COMP     165
   IF OP = EQOP THEN INSYMBOL ELSE ERROR(16);                                COMP     165
   CONSTANT(FSYS+[SEMICOLON],LSP,LVALU);                                     COMP     165
   ENTERID(LCP);                                                             COMP     166
   LCP↑.IDTYPE := LSP; LCP↑.VALUES := LVALU;                                 COMP     166
   IF SY = SEMICOLON THEN                                                    COMP     166
    BEGIN INSYMBOL;                                                          COMP     166
     TEST2(FSYS+[IDENT],6,[])                                                COMP     166
    END                                                                      COMP     166
   ELSE ERROR(14)                                                            COMP     166
  END                                                                        COMP     166
END (*CONSTDECLARATION*) ;                                                   COMP     166

PROCEDURE FORWTEST;                                                          COMP     167
BEGIN                                                                        COMP     167
 IF FWPTR <> NIL THEN                                                        COMP     167
  BEGIN ERROR(117); WRITELN;                                                 COMP     167
   REPEAT WRITELN('*** UNDEFINED TYPE: ',FWPTR↑.NAME);                       COMP     167
    FWPTR:=FWPTR↑.NEXT                                                       COMP     167
   UNTIL FWPTR = NIL;                                                        COMP     167
  SPACES                                                                     COMP     167
 END                                                                         COMP     167
END (* FORWTEST *);                                                          COMP     167

PROCEDURE TYPEDECLARATION;                                                   COMP     168
 VAR LCP,LCP1,LCP2: CTP; LSP: STP;                                           COMP     168
BEGIN                                                                        COMP     168
 IF SY <> IDENT THEN                                                         COMP     168
  BEGIN ERROR(2); SKIP(FSYS+[IDENT]) END;                                    COMP     168
 WHILE SY = IDENT DO                                                         COMP     168
  BEGIN NEW(LCP,TYPES);                                                      COMP     168
   WITH LCP↑ DO                                                              COMP     168
    BEGIN NAME := ID; IDTYPE := NIL; KLASS := TYPES END;                     COMP     168
   INSYMBOL;                                                                 COMP     169
   IF OP = EQOP THEN INSYMBOL ELSE ERROR(16);                                COMP     169
   TYP(FSYS+[SEMICOLON],LSP);                                                COMP     169
   ENTERID(LCP);                                                             COMP     169
   LCP↑.IDTYPE := LSP;                                                       COMP     169
   (*HAS ANY FORWARD REFERENCE BEEN SATISFIED:*)                             COMP     169
   LCP1 := FWPTR;                                                            COMP     169
   WHILE LCP1 <> NIL DO                                                      COMP     169
    BEGIN                                                                    COMP     169
     IF LCP1↑.NAME = LCP↑.NAME THEN                                          COMP     169
```

```
      BEGIN
       WITH LCP↑ DO
        BEGIN LCP1↑.IDTYPE↑.ELTYPE := IDTYPE;
         IF IDTYPE <> NIL THEN
          IF IDTYPE↑.FTYPE THEN ERROR(108)
        END;
       IF LCP1 <> FWPTR THEN
        LCP2↑.NEXT := LCP1↑.NEXT
       ELSE FWPTR := LCP1↑.NEXT;
      END;
     LCP2 := LCP1; LCP1 := LCP1↑.NEXT
    END;
   IF SY = SEMICOLON THEN
    BEGIN INSYMBOL;
     TEST2(FSYS+[IDENT],6,[])
    END
   ELSE ERROR(14)
  END;
 FORWTEST;
END (*TYPEDECLARATION*) ;

PROCEDURE VARDECLARATION;
 VAR LCP,NXT: CTP; LSP: STP; EXITLOOP: BOOLEAN;
BEGIN NXT := NIL;
 REPEAT
  (*LOOP UNTIL SY <> COMMA:*)
  REPEAT
   IF SY = IDENT THEN
    BEGIN NEW(LCP,VARS);
     WITH LCP↑ DO
      BEGIN NAME := ID; NEXT := NXT; KLASS := VARS;
       IDTYPE := NIL; VKIND := DRCT; VLEV := LEVEL
      END;
     ENTERID(LCP);
     NXT := LCP;
     INSYMBOL;
    END
   ELSE ERROR(2);
   TEST2(FSYS+[COMMA,COLON]+TYPEDELS,6,[SEMICOLON]);
   EXITLOOP := SY <> COMMA;
   IF NOT EXITLOOP THEN INSYMBOL
  UNTIL EXITLOOP;
  TEST1(COLON,5);
  TYP(FSYS+[SEMICOLON]+TYPEDELS,LSP);
  WHILE NXT <> NIL DO
   WITH  NXT↑ DO
    BEGIN IDTYPE := LSP; VADDR := LC;
     IF LSP <> NIL THEN LC := LC + FULLWORDS(LSP↑.SIZE);
     NXT := NEXT
    END;
  IF SY = SEMICOLON THEN
   BEGIN INSYMBOL;
    TEST2(FSYS+[IDENT],6,[])
   END
  ELSE ERROR(14)
 UNTIL (SY <> IDENT)AND NOT (SY IN TYPEDELS);
 FORWTEST;
END (*VARDECLARATION*) ;

PROCEDURE PROCEDUREDECLARATION(FSY: SYMBOL);
 VAR OLDLEV: 0..MAXLEVEL; LSY: SYMBOL; LCP,LCP1: CTP; LSP: STP;
   FORW: BOOLEAN; OLDTOP: DISPRANGE;
   LLC,LCM: ADDRRANGE; DLOCP: LOCOFREF;

  PROCEDURE PARAMETERLIST(FSY: SETOFSYS; VAR FPAR: CTP);
   VAR LCP,LCP1,LCP2,LCP3: CTP; LSP,LSP1,LSP2: STP; LKIND: DRCTINDRCT;
     DESCSZ,SZ: INTEGER;
     PACKFLAG,EXITLOOP: BOOLEAN;
  BEGIN LCP1 := NIL;
   TEST2(FSY+[LPARENT],7,FSYS);
   IF SY = LPARENT THEN
    BEGIN IF FORW THEN ERROR(119);
     INSYMBOL;
     IF NOT (SY IN [IDENT,VARSY,PROCEDURESY,FUNCTIONSY]) THEN
      BEGIN ERROR(7); SKIP(FSYS+[IDENT,RPARENT]) END;
     WHILE SY IN [IDENT,VARSY,PROCEDURESY,FUNCTIONSY] DO
      BEGIN
       IF SY = PROCEDURESY THEN
        BEGIN
         REPEAT INSYMBOL;
          IF SY = IDENT THEN
```

```
      BEGIN NEW(LCP,PROC,DECLARED,FORMAL);                                  COMP     178
        WITH LCP↑ DO                                                        COMP     178
         BEGIN NAME := ID; IDTYPE := NIL; NEXT := LCP1;                     COMP     178
          KLASS := PROC; PFDECKIND := DECLARED;                             COMP     178
          PFLEV := LEVEL; PFADDR := LC; PFKIND := FORMAL;                   COMP     178
          PFXOPT:=XPARMAX;                                                  COMP     178
         END;                                                               COMP     178
        ENTERID(LCP);                                                       COMP     178
        LCP1 := LCP; LC := LC + 1;                                          COMP     178
        INSYMBOL                                                            COMP     179
       END                                                        000023    COMP     179
      ELSE ERROR(2);                                                        COMP     179
      TEST2(FSYS+[COMMA,SEMICOLON,RPARENT],7,[])                            COMP     179
     UNTIL SY <> COMMA                                                      COMP     179
    END                                                                     COMP     179
   ELSE                                                                     COMP     179
    BEGIN LCP2 := LCP1; LSP := NIL;                                         COMP     179
     IF SY = FUNCTIONSY THEN                                                COMP     179
      BEGIN                                                                 COMP     179
       REPEAT INSYMBOL;                                                     COMP     180
        IF SY = IDENT THEN                                                  COMP     180
         BEGIN NEW(LCP,FUNC,DECLARED,FORMAL);                               COMP     180
          WITH LCP↑ DO                                                      COMP     180
           BEGIN NAME := ID; IDTYPE := NIL; NEXT := LCP1;                   COMP     180
            KLASS := FUNC; PFDECKIND := DECLARED;                           COMP     180
            PFLEV := LEVEL; PFADDR := LC; PFKIND := FORMAL                  COMP     180
           END;                                                             COMP     180
          ENTERID(LCP);                                                     COMP     180
          LCP1 := LCP; LC := LC + 1;                                        COMP     180
          INSYMBOL                                                          COMP     181
         END                                                                COMP     181
        ELSE ERROR(2);                                                      COMP     181
        IF NOT (SY IN [COMMA,COLON]+FSYS) THEN                              COMP     181
         BEGIN ERROR(7); SKIP(FSYS+[COMMA,SEMICOLON,RPARENT])               COMP     181
         END                                                                COMP     181
       UNTIL SY <> COMMA;                                                   COMP     181
       IF SY = COLON THEN                                                   COMP     181
        BEGIN INSYMBOL;                                                     COMP     181
         IF SY = IDENT THEN                                                 COMP     181
          BEGIN SEARCHID([TYPES],LCP);                                      COMP     182
           LSP := LCP↑.IDTYPE;                                              COMP     182
           IF LSP <> NIL THEN                                               COMP     182
            IF NOT (LSP↑.FORM IN [SCALAR,SUBRANGE,POINTER])                 COMP     182
             THEN BEGIN ERROR(120); LSP := NIL END;                         COMP     182
           INSYMBOL                                                         COMP     182
          END                                                               COMP     182
         ELSE ERROR(2);                                                     COMP     182
         TEST2(FSYS+[SEMICOLON,RPARENT],7,[])                               COMP     182
        END                                                                 COMP     182
       ELSE ERROR(5)                                                        COMP     183
      END                                                                   COMP     183
     ELSE                                                                   COMP     183
      BEGIN                                                                 COMP     183
       IF SY = VARSY THEN                                                   COMP     183
        BEGIN LKIND := INDRCT; INSYMBOL END                                 COMP     183
       ELSE LKIND := DRCT;                                                  COMP     183
       (*LOOP UNTIL SY <> COMMA*)                                           COMP     183
       REPEAT                                                               COMP     183
        IF SY = IDENT THEN                                                  COMP     183
         BEGIN NEW(LCP,VARS);                                               COMP     184
          WITH LCP↑ DO                                                      COMP     184
           BEGIN NAME := ID; IDTYPE := NIL; KLASS := VARS;                  COMP     184
            VKIND := LKIND; NEXT := LCP1; VLEV := LEVEL;                    COMP     184
            VADDR := LC                                                     COMP     184
           END;                                                             COMP     184
          ENTERID(LCP);                                                     COMP     184
          LCP1 := LCP; LC := LC + 1;                                        COMP     184
          INSYMBOL;                                                         COMP     184
         END                                                                COMP     184
        ELSE ERROR(2);                                                      COMP     185
        IF NOT (SY IN [COMMA,COLON]+FSYS) THEN                              COMP     185
         BEGIN ERROR(7); SKIP(FSYS+[COMMA,SEMICOLON,RPARENT])               COMP     185
         END;                                                               COMP     185
        EXITLOOP := SY <> COMMA;                                            COMP     185
        IF NOT EXITLOOP THEN INSYMBOL                                       COMP     185
       UNTIL EXITLOOP;                                                      COMP     185
       IF SY = COLON THEN                                                   COMP     185
        BEGIN INSYMBOL;                                                     COMP     185
          IF SY = IDENT THEN                                                COMP     185
           BEGIN SEARCHID([TYPES],LCP);                                     COMP     186
            LSP := LCP↑.IDTYPE;                                             COMP     186
```

```
              INSYMBOL
            END
           ELSE ERROR(2);
          IF LSP <> NIL THEN
           IF (LKIND = DRCT) AND LSF↑.FTYPE THEN
            ERROR(121);
          TEST2(FSYS+[SEMICOLON,RPARENT],7,[])
         END
        ELSE ERROR(5);
       END;
      LCP3 := LCP1;
      WHILE LCP3 <> LCP2 DO
       BEGIN LCP3↑.IDTYPE := LSP; LCP3 := LCP3↑.NEXT END
     END;
    IF SY = SEMICOLON THEN
     BEGIN INSYMBOL;
      IF NOT (SY IN FSYS+[IDENT,VARSY,PROCEDURESY,FUNCTIONSY]) THEN
       BEGIN ERROR(7); SKIP(FSYS+[IDENT,RPARENT]) END
     END
   END (*WHILE*) ;
  IF SY = RPARENT THEN
   BEGIN INSYMBOL;
    TEST2(FSY+FSYS,6,[])
   END
  ELSE ERROR(4);
  LCP3 := NIL;
  (*REVERSE POINTERS AND RESERVE LOCAL CELLS FOR COPIES OF
   EITHER MULTIPLE VALUES OR THEIR DESCRIPTORS*)
  WHILE LCP1 <> NIL DO
   WITH LCP1↑ DO
    BEGIN LCP2 := NEXT; NEXT := LCP3;
     IF KLASS = VARS THEN
      IF IDTYPE <> NIL THEN
      WITH IDTYPE↑ DO
         BEGIN SZ := FULLWORDS(SIZE);
          IF (VKIND = DRCT)AND (SZ <> 1) THEN
           BEGIN VADDR := LC; LC := LC + SZ
           END
         END;
     LCP3 := LCP1; LCP1 := LCP2
    END;
  FPAR := LCP3
 END
  ELSE FPAR := NIL
END (*PARAMETERLIST*) ;

BEGIN (*PROCEDUREDECLARATION*)
 LLC := LC; LC := 3; FORW := FALSE; DP := TRUE;
 IF SY = IDENT THEN
  BEGIN SEARCHS:CTION(DISPLAY[TOP].FNAME,LCP); (*DECIDE WHETHER FORW*)
   IF LCP <> NIL THEN
   BEGIN
    IF LCP↑.KLASS = PROC THEN
     FORW := (LCP↑.PFDECL = FORWDECL)AND (FSY = PROCEDURESY)
         AND (LCP↑.PFKIND = ACTUAL)
    ELSE
     IF LCP↑.KLASS = FUNC THEN
      FORW := (LCP↑.PFDECL = FORWDECL)AND (FSY = FUNCTIONSY)
          AND (LCP↑.PFKIND = ACTUAL)
     ELSE FORW := FALSE;
    IF NOT FORW THEN ERROR(160)
   END
   ELSE FORW := FALSE;
   IF NOT FORW THEN
    BEGIN
     IF FSY = PROCEDURESY THEN
      BEGIN NEW(LCP,PROC,DECLARED,ACTUAL); LCP↑.KLASS := PROC END
     ELSE
      BEGIN NEW(LCP,FUNC,DECLARED,ACTUAL); LCP↑.KLASS := FUNC END;
     WITH LCP↑ DO
      BEGIN NAME := ID; IDTYPE := NIL; NEXT := NIL;
       PFDECKIND := DECLARED; PFKIND := ACTUAL;
       PFDECL := DECL; PFLEV := LEVEL;
       PCNT := PCNT + 1;
       PFXOPT:=XPARMAX;
       IF EXTON THEN PFCNT := 0 ELSE PFCNT := PCNT
      END;
     ENTERID(LCP)
    END
   ELSE
    BEGIN LCP1 := LCP↑.NEXT;
```

```
    WHILE LCP1 <> NIL DO                                                    000025
     BEGIN
      WITH LCP1↑ DO
       IF KLASS = VARS THEN
       BEGIN
        IF IDTYPE <> NIL THEN
         BEGIN LCM := VADDR + FULLWORDS(IDTYPE↑.SIZE);
          IF LCM > LC THEN LC := LCM
         END;
       END
       ELSE
        BEGIN LCM := PFADDR + 1;
         IF LCM > LC THEN LC := LCM
        END;
       LCP1 := LCP1↑.NEXT
      END
     END;
   INSYMBOL
  END
 ELSE
  BEGIN ERROR(2); LCP := UFCTPTR END;
 OLDLEV := LEVEL; OLDTOP := TOP;
 IF LEVEL < MAXLEVEL THEN LEVEL := LEVEL + 1 ELSE ERROR(251);
 IF TOP < DISPLIMIT THEN
  BEGIN TOP := TOP + 1;
   WITH DISPLAY[TOP] DO
    BEGIN
     IF FORW THEN FNAME := LCP↑.NEXT
     ELSE FNAME := NIL;
     OCCUR := BLCK
    END
  END
 ELSE ERROR(250);
 IF FSY = PROCEDURESY THEN
  BEGIN PARAMETERLIST([SEMICOLON],LCP1);
   IF NOT FORW THEN LCP↑.NEXT := LCP1
  END
 ELSE
  BEGIN PARAMETERLIST([SEMICOLON,COLON],LCP1);
   IF NOT FORW THEN LCP↑.NEXT := LCP1;
   IF SY = COLON THEN
    BEGIN INSYMBOL;
     IF SY = IDENT THEN
      BEGIN IF FORW THEN ERROR(122);
       SEARCHID([TYPES],LCP1);
       LSP := LCP1↑.IDTYPE;
       LCP↑.IDTYPE := LSP;
       IF LSP <> NIL THEN
        IF NOT (LSP↑.FORM IN [SCALAR,SUBRANGE,POINTER]) THEN
         BEGIN ERROR(120); LCP↑.IDTYPE := NIL END;
       INSYMBOL
      END
     ELSE BEGIN ERROR(2); SKIP(FSYS+[SEMICOLON]) END
    END
   ELSE
    IF NOT FORW THEN ERROR(123)
  END;
 TEST1(SEMICOLON,14);
  IF (SY = IDENT)AND ((ID = ΞFORWARD   Ξ)OR (ID = ΞEXTERN    Ξ)
   OR (ID = ΞFORTRAN   Ξ)) THEN
   BEGIN IF FORW THEN ERROR(161);
    WITH LCP↑ DO
     BEGIN
      IF ID = ΞFORWARD   Ξ THEN
       BEGIN PFDECL := FORWDECL; LFORWCNT := LFORWCNT + 1 END
      ELSE
       IF ID = ΞEXTERN    Ξ THEN PFDECL := EXTDECL
       ELSE PFDECL := FTNDECL;
      IF PFDECL IN [EXTDECL,FTNDECL] THEN PFCNT := 0
     END;
    INSYMBOL;
    TEST1(SEMICOLON,14);
    TEST2(FSYS,6,[])
  END
 ELSE
   BEGIN LCP↑.PFDECL := DECL;
    IF FORW THEN LFORWCNT := LFORWCNT - 1;
   NEW(DLCP);
   REPEAT BLOCK(FSYS,SEMICOLON,LCP);
    IF SY = SEMICOLON THEN
     BEGIN INSYMBOL;
```

```
      IF NOT (SY IN [BEGINSY,PROCEDURESY,FUNCTIONSY]) THEN                    COMP  202
       BEGIN ERROR(6); SKIP(FSYS) END                                         COMP  202
     END                                                                      COMP  202
    ELSE ERROR(14)                                                            COMP  202
   UNTIL SY IN [BEGINSY,PROCEDURESY,FUNCTIONSY];                              COMP  202
   RELEASE(DLOCP);                                          000026            COMP  202
  END;                                                                        COMP  203
 LEVEL := OLDLEV; TOP := OLDTOP; LC := LLC;                                   COMP  203
END (*PROCEDUREDECLARATION*) ;                                                COMP  203
                                                                              COMP  203
PROCEDURE BODY(FSYS: SETOFSYS);                                               COMP  203
 TYPE OPRANGE = 0..63;                                                        COMP  203
    RCODERANGE = 0..RCODEMAX; RELRANGE= 0..2;                                 COMP  203
    CODEP = ↑CODESEGMENT;                                                     COMP  203
    CODESEGMENT = RECORD NXTSEG: CODEP;                                       COMP  203
                  RCODE: ARRAY [RCODERANGE] OF INTEGER;                       COMP  203
                  CODE: ARRAY [CODERANGE] OF INTEGER                          COMP  204
                 END;                                                         COMP  204
    CLOSEP = ↑ CLOSEREC;                                                      COMP  204
    CLOSEREC = PACKED RECORD NXTP: CLOSEP;                                    COMP  204
               EFETADDR: ADDRRANGE                                            COMP  204
             END;                                                             COMP  204
                                                                              COMP  204
                                                                              COMP  204
    CSTKIND = (NOP,PUREP,POSP,NEGP);                                          COMP  204
    CSTREC = PACKED RECORD                                                    COMP  204
         CASE CKIND : CSTKIND OF                                              COMP  205
          PUREP: (EXP: BITRANGE);                                             COMP  205
          POSP,                                                               COMP  205
          NEGP:  (EXP1,EXP2: BITRANGE)                                        COMP  205
         END;                                                                 COMP  205
                                                                              COMP  205
                                                                              COMP  205
 VAR                                                                          COMP  205
   BONUS: ARRAY [SHRTCST..INDVAR] OF INTEGER;                                 COMP  205
   CSEGP: CODEP;                                                              COMP  205
                                                                              COMP  206
                                                                              COMP  206
   LCP: CTP; LFSTOCC: LOCOFREF;                                               COMP  206
   FCLSP,CLSP: CLOSEP; LP: CTAILP;                                            COMP  206
   I: REGNR;                                                                  COMP  206
   LPL,LPL1,LPL2: PLACE;                                                      COMP  206
   LCMAX,LDISP,LSZ: ADDRRANGE;                                                COMP  206
 RCIX: RCODERANGE; RCP: 1..15;                                                COMP  206
   LASTOP: OPRANGE;   LASTI: REGNR;                                           COMP  206
   LOCP: LOCOFREF; LCSP: CSP;                                                 COMP  206
   LCD: INTEGER;                                                              COMP  207
   EXFILP: EXTFILEP; LOCFILES: BOOLEAN;                                       COMP  207
                                                                              COMP  207
                                                                              COMP  207
 PROCEDURE NOOP;                                                              COMP  207
 BEGIN LASTOP:= 46B;                                                          COMP  207
  WITH PC DO                                                                  COMP  207
  WHILE CP<4 DO BEGIN CBUF:= CBUF*100000B+46000B;                             COMP  207
        RBUF:= 2*RBUF; CP:= CP+1                                              COMP  207
       END                                                                    COMP  207
 END (*NOOP*);                                                                COMP  208
 PROCEDURE PUTREL(R: INTEGER);                                                COMP  208
  VAR SEGP: CODEP;                                                            COMP  208
 BEGIN CSEGP↑.RCODE[RCIX] := RBUF; RBUF := R; RCP := 1;                       COMP  208
  WITH PC DO                                                                  COMP  208
   IF CIX = CODEMAX THEN                                                      COMP  208
    BEGIN NEW(SEGP); SEGP↑.NXTSEG := CSEGP; CSEGP := SEGP;                    COMP  208
     CIX := 0; SIX := SIX + 1; RCIX := 0                                      COMP  208
    END;                                                                      COMP  208
   RCIX := RCIX + 1                                                           COMP  208
 END (*PUTREL*) ;                                                             COMP  209
                                                                              COMP  209
                                                                              COMP  209
 PROCEDURE GEN15(FOP: OPRANGE; FI,FJ,FK: REGNR);                              COMP  209
 BEGIN LASTOP:= FOP; LASTI:= FI;                                              COMP  209
 WITH PC DO                                                                   COMP  209
  IF CP<>4 THEN                                                               COMP  209
  BEGIN CP:= CP+1; CBUF:= CBUF*64+FOP; RBUF:= RBUF*2;                         COMP  209
  END ELSE                                                                    COMP  209
  BEGIN CSEGP↑.CODE[CIX] := CBUF; CBUF := FOP; CP := 1;                       COMP  209
  IF RCP = 15 THEN PUTREL(0)                                                  COMP  210
  ELSE BEGIN RBUF := 2*RBUF; RCP := RCP + 1 END;                              COMP  210
  CIX := CIX + 1; IC := IC + 1                                                COMP  210
  END;                                                                        COMP  210
  CBUF:= ((8*CBUF+FI)*8+FJ)*8+FK;                                             COMP  210
```

```
  END (*GEN15*) ;                                                                                                  COMP      210
                                                                                                                   COMP      210
  PROCEDURE GEN30(FOP: OPRANGE; FI,FJ: REGNR; FK: ADDRRANGE;                                                       COMP      210
   FR: RELRANGE);                                                                                                  COMP      210
   LABEL 1;                                                                                                        COMP      210
   VAR I,J,K: REGNR; MAXPR: CODERANGE;                                                                             COMP      211
       EXTRP: EXTREFP;                                                                          000027             COMP      211
  BEGIN                                                                                                            COMP      211
   WITH PC DO                                                                                                      COMP      211
   IF CP<3 THEN                                                                                                    COMP      211
   BEGIN CBUF:= 64*CBUF+FOP; RBUF:= 4*RBUF+FR; CP:= CP+2;                                                          COMP      211
   END ELSE                                                                                                        COMP      211
   BEGIN                                                                                                           COMP      211
    IF CP = 3 THEN (*REPLACE NOOP BY BXIXJ INSTRUCTION IF APPROPRIATE*)                                            COMP      211
     BEGIN                                                                                                         COMP      211
      FOR I := 0 TO 5 DO (*FIND AVAILABLE X-REGISTER*)                                                             COMP      212
       IF (XRGS[I].XCONT=AVAIL)AND (I<>FJ) THEN                                                                    COMP      212
        BEGIN MAXPR := 0;                                                                                          COMP      212
         FOR J := 6 TO 7 DO                                                                                        COMP      212
          WITH XRGS[J] DO                                                                                          COMP      212
           IF XCONT IN [SIMPVAR,INDVAR] THEN                                                                       COMP      212
            IF REFNR <> 0 THEN                                                                                     COMP      212
             BEGIN MAXPR := IC; K := J END                                                                         COMP      212
            ELSE                                                                                                   COMP      212
             IF LASTREF > MAXPR THEN                                                                               COMP      212
              BEGIN MAXPR := LASTREF; K := J END;                                                                  COMP      213
         IF MAXPR > 0 THEN (*COPY X-K INTO X-I*)                                                                   COMP      213
          BEGIN GEN15(10B,I,K,0); XRGS[I] := XRGS[K];                                                              COMP      213
           FOR J := 1 TO 7 DO                                                                                      COMP      213
            WITH ARGS[J] DO                                                                                        COMP      213
             IF ACONT = INDADDR THEN                                                                               COMP      213
              IF AREG = I THEN ACONT := UNSPECADDR;                                                                COMP      213
           WITH XRGS[K] DO                                                                                         COMP      213
            IF REFNR <> 0 THEN                                                                                     COMP      213
             BEGIN XCONT := OTHER;                                                                                 COMP      213
              WITH XRGS[I] DO                                                                                      COMP      214
               BEGIN REFNR := 0; LASTREF := IC END                                                                 COMP      214
             END                                                                                                   COMP      214
            ELSE XCONT := AVAIL                                                                                    COMP      214
          END                                                                                                      COMP      214
         ELSE NOOP;                                                                                                COMP      214
         GOTO 1                                                                                                    COMP      214
        END;                                                                                                       COMP      214
      NOOP;                                                                                                        COMP      214
1: END (*CP = 3*) ;                                                                                                COMP      214
    CSEGP↑.CODE[CIX] := CBUF; CBUF := FOP; CP := 2;                                                                COMP      215
    IF RCP = 15 THEN PUTREL(FR)                                                                                    COMP      215
    ELSE BEGIN RBUF := RBUF*4 + FR; RCP := RCP + 1 END;                                                            COMP      215
    CIX := CIX + 1; IC := IC + 1                                                                                   COMP      215
   END;                                                                                                            COMP      215
   LASTOP := FOP; LASTI := FI;                                                                                     COMP      215
   CBUF:= (8*CBUF+FI)*8+FJ;                                                                                        COMP      215
   IF FK>=0 THEN                                                                                                   COMP      215
   BEGIN CBUF:= CBUF*1000B*1000B+FK;                                                                               COMP      215
    IF EXT <> NIL THEN                                                                                             COMP      215
    BEGIN NEW(EXTRP); EXTRX := EXTRX + 1;                                                                          COMP      216
     WITH EXTRP↑,EXT↑ DO                                                                                           COMP      216
     BEGIN                                                                                                         COMP      216
      LINK := REF; REF := EXTRP;                                                                                   COMP      216
      LOC := ((8 - PC.CP)*1000B + 1)*1000000B + IC - 1;                                                            COMP      216
      EXT := NIL                                                                                                   COMP      216
     END                                                                                                           COMP      216
    END;                                                                                                           COMP      216
   END ELSE CBUF:= CBUF*1000B*1000B+777777B+FK;                                                                    COMP      216
  END (*GEN30*)  ;                                                                                                 COMP      216
                                                                                                                   COMP      217
  PROCEDURE INS(FIC: ADDRRANGE; FPL: PLACE);                                                                       COMP      217
   VAR SEGP: CODEP; I: INTEGER;                                                                                    COMP      217
  BEGIN IF FIC < 0 THEN FIC := 777777B + FIC;                                                                      COMP      217
   WITH FPL DO                                                                                                     COMP      217
    BEGIN IF (SIX=PC.SIX)AND(CIX=PC.CIX) THEN CP := 4 - PC.CP + CP;                                                COMP      217
     CASE CP OF                                                                                                    COMP      217
      1: FIC := FIC*1000000000000000B;                                                                             COMP      217
      2: FIC := FIC*100000000000B;                                                                                 COMP      217
     3: FIC := FIC*1000000B;                                                                                       COMP      217
     4:                                                                                                            COMP      218
     END;                                                                                                          COMP      218
     IF SIX = PC.SIX THEN                                                                                          COMP      218
      BEGIN IF CIX = PC.CIX THEN CBUF := CBUF + FIC                                                                COMP      218
            ELSE WITH CSEGP↑ DO CODE[CIX] := CODE[CIX] + FIC                                                       COMP      218
      END                                                                                                          COMP      218
```

000028

```
   ELSE
    BEGIN SEGP := CSEGP;
     FOR I := PC.SIX - 1 DOWNTO SIX DO SEGP := SEGP↑.NXTSEG;
     WITH SEGP↑ DO CODE[CIX] := CODE[CIX] + FIC
    END
  END
END (*INS*) ;

PROCEDURE LINKOCC(VAR FPTR: LOCOFREF);
 VAR LOCP: LOCOFREF;
BEGIN NEW(LOCP);
 WITH LOCP↑, PC DO
  BEGIN NXTREF := FPTR; FPTR := LOCP;
   LOC := PC
  END
END (*LINKOCC*) ;

PROCEDURE GEN60(FC:INTEGER);
VAR I:SHRTINT;
BEGIN NOOP;
WITH PC DO
BEGIN
 CSEGP↑.CODE[CIX] := CBUF; CBUF := FC;
 IF RCP = 15 THEN PUTREL(0)
 ELSE BEGIN RBUF := RBUF*16; RCP := RCP + 1 END;
 CIX := CIX + 1; IC := IC + 1
END
END (*GEN60*);

PROCEDURE PMDINFO(FCP: CTP);
 VAR I: INTEGER;
BEGIN
 IF FCP <> NIL THEN
  WITH FCP↑ DO
   BEGIN PMDINFO(LLINK); PMDINFO(RLINK);
    IF KLASS = VARS THEN
     IF IDTYPE <> NIL THEN
      IF IDTYPE↑.FORM <= POINTER THEN
       BEGIN ALFINT.A := NAME; GEN60(ALFINT.I);
        IF IDTYPE↑.FORM = POINTER THEN I := 14
        ELSE
         IF COMPTYPES(IDTYPE,INTPTR,FALSE) THEN I := 2
         ELSE
          IF COMPTYPES(IDTYPE,REALPTR,FALSE) THEN I := 4
          ELSE
           IF COMPTYPES(IDTYPE,CHARPTR,FALSE) THEN I := 6
           ELSE
            IF COMPTYPES(IDTYPE,BOOLPTR,FALSE) THEN I := 8
            ELSE I := 10;
        IF VKIND = INDRCT THEN I := I + 1;
        GEN60(I*1000000B + VADDR)
       END
   END
END (*PMDINFO*) ;

PROCEDURE SHORTNAME(FNAME: ALFA; VAR F1NAME: ALFA);
 VAR I: 1..ALFALENG;
BEGIN I := ALFALENG;
 IF FNAME <> ≡          ≡ THEN
  WHILE (FNAME[I] = ≡ ≡)OR (I > 7) DO
   BEGIN FNAME[I] := CHR(0); I := I - 1 END;
 F1NAME := FNAME
END (*SHORTNAME*);

PROCEDURE PFNAME(FI: INTEGER);
 (*CREATE INTERNAL NAME FOR PROC/FUNC FROM FI*)
 VAR K,L: INTEGER;
BEGIN
 FOR K := 7 DOWNTO 4 DO
  BEGIN L := FI DIV 8;
   PNAME[K] := CHR(ORD(≡0≡) + FI - 8*L);
   FI := L
  END
END (*PFNAME*);

PROCEDURE LGOHEAD(NAME: ALFA);
 TYPE STYPE= (WORD,ADRS,NAMS);
     B18 = 0..777777B;
 VAR PGNAME: INTEGER; I: INTEGER; LLP: LBP;
  STRUCTURES: RECORD CASE STYPE OF
          WORD: (CVAL: INTEGER);
```

```
        NAMS: (CNAM: ALFA);                                                    COMP      226
        ADRS: (IDW: PACKED RECORD CN: 0..63;                                   COMP      226
                    WC: B18; LR: B18; L: B18                    000029         COMP      226
              END)                                                             COMP      227
      END;                                                                     COMP      227
  LABERR : BOOLEAN;                                                            COMP      227
BEGIN                                                                          COMP      227
 WITH STRUCTURES DO                                                            COMP      227
 BEGIN CVAL := 0; IDW.CN := 77B; IDW.WC := 16B;          (*PREFIX*)            COMP      227
  LGO↑:= CVAL; PUT(LGO); SHORTNAME(NAME,CNAM); IDW.L:= 0;                      COMP      227
  PGNAME:= CVAL; LGO↑:= PGNAME; PUT(LGO);                                      COMP      227
  DATE(CNAM); LGO↑ := CVAL*100B; PUT(LGO);                                     COMP      227
  TIME(CNAM); LGO↑ := CVAL*100B; PUT(LGO);                                     COMP      227
  CNAM := ≡SCOPE 3.4 ≡; LGO↑ := CVAL; PUT(LGO);  (*OPER SYSTEM*)               COMP      228
  CNAM := ≡PASCAL 3.0≡; LGO↑ := CVAL; PUT(LGO);  (*COMPILER VERSION*)          MSUTITLE
  CNAM := ≡.01      ≡; LGO↑ := CVAL; PUT(LGO);  (*UPDATE LEVEL*)               MSUTITLE
  CNAM := ≡ I       ≡; LGO↑ := CVAL; PUT(LGO);  (*HARDWARE SPEC*)              COMP      228
  FOR I := 1 TO 7 DO BEGIN LGO↑ := 0; PUT(LGO) END;                            COMP      228
  CVAL := 0; IDW.CN := 70B; IDW.WC := 2;                (*LDSET*)             COMP      228
  LGO↑ := CVAL; PUT(LGO);                                                      COMP      228
  CVAL := 0; IDW.WC := 100001B; LGO↑ := CVAL; PUT(LGO);                        COMP      228
  SHORTNAME(≡PASCAL    ≡,CNAM); CNAM[7] := CHR(0);                             MSUPASCAL
   LGO↑ := CVAL; PUT(LGO);                                                     MSUPASCAL
  CVAL:= 0; IDW.CN:= 34B;                               (*PIDL*)             COMP      228
  IF LEVEL = 1 THEN IDW.WC := 2 ELSE IDW.WC := 1;                              COMP      229
  LGO↑:= CVAL; PUT(LGO); LGO↑:= PGNAME; PUT(LGO);                              COMP      229
  IF LEVEL = 1 THEN                                                            COMP      229
   BEGIN                                                                       COMP      229
    CNAM:=≡           ≡; IDW.L:= 0; LGO↑:= CVAL; PUT(LGO);                     COMP      229
   END;                                                                        COMP      229
  CVAL:= 0; IDW.CN:= 36B;                               (*ENTR*)             COMP      229
  I := 0; (*COUNT NUMBER OF ADDITIONAL ENTRY POINTS*)                          COMP      229
  LLP := FSTLABP;                                                              COMP      229
  WHILE LLP <> FLABP DO                                                        COMP      229
   WITH LLP↑ DO                                                                COMP      230
    BEGIN IF LCNT <> 0 THEN I := I + 1;                                        COMP      230
     LLP := NEXTLAB                                                            COMP      230
    END;                                                                       COMP      230
  IF (LEVEL = 1)AND (NAME <> ≡P.MAIN    ≡) THEN                                COMP      230
   BEGIN IDW.WC := 2*(I + 2); LGO↑ := CVAL; PUT(LGO);                          COMP      230
    LGO↑ := PGNAME; PUT(LGO); LGO↑ := 1000003B; PUT(LGO);                      COMP      230
    SHORTNAME(≡P.MAIN    ≡,ALFINT.A); LGO↑ := ALFINT.I; PUT(LGO);              COMP      230
    LGO↑ := 10000033B; PUT(LGO)                                                COMP      230
   END                                                                         COMP      230
  ELSE                                                                         COMP      231
   BEGIN IDW.WC := 2*(I + 1); LGO↑ := CVAL; PUT(LGO);                          COMP      231
    LGO↑ := PGNAME; PUT(LGO);                                                  COMP      231
    IF LEVEL = 1 THEN LGO↑ := 1000003B ELSE LGO↑ := 1000002B;                  COMP      231
    PUT(LGO)                                                                   COMP      231
   END;                                                                        COMP      231
   LABERR:=FALSE;                                                             COMP      231
  WHILE FSTLABP <> FLABP DO                                                    COMP      231
   WITH FSTLABP↑ DO                                                            COMP      231
    BEGIN                                                                      COMP      231
     IF LCNT <> 0 THEN                                                         COMP      232
      BEGIN PASCL[7] := CHR(LCNT);                                             COMP      232
       ALFINT.A := PASCL; LGO↑ := ALFINT.I; PUT(LGO);                          COMP      232
       IF DEFINED THEN                                                         COMP      232
        BEGIN CVAL := LABADDR; IDW.LR := 1; LGO↑ := CVAL;                      COMP      232
         PUT(LGO)                                                              COMP      232
        END                                                                    COMP      232
       ELSE LABERR:=TRUE                                                       COMP      232
      END                                                                      COMP      232
     ELSE IF NOT DEFINED AND (FSTOCC <> NIL) THEN LABERR:=TRUE;                COMP      232
     IF LABERR THEN                                                            COMP      232
      BEGIN ERROR(168); WRITELN;                                               COMP      232
       WRITELN(≡ ***     UNDEFINED LABEL: ≡,LABVAL);                           COMP      232
       LABERR:=FALSE                                                           COMP      232
      END;                                                                     COMP      232
     FSTLABP := NEXTLAB                                                        COMP      232
    END;                                                                       COMP      232
 END;                                                                          COMP      232
END (*LGOHEAD*);                                                               COMP      232
                                                                               COMP      233
PROCEDURE LGOTEXT;                                                             COMP      234
 TYPE B18 = 0..777777B;                                                        COMP      234
 VAR J,DISP: 0..15;                                                            COMP      234
   I,RCMAX: RCODERANGE; K: INTEGER; SEGP1,SEGP2: CODEP;                        COMP      234
   L,LCIX: CODERANGE;                                                          COMP      234
   STRUCTURES: RECORD CASE BOOLEAN OF                                          COMP      234
         TRUE: (CVAL: INTEGER);                                                COMP      234
```

```
          FALSE: (IDW: PACKED RECORD CN: 0..63;                                  COMP    234
                                WC: B18; LR: B18; L: B18                         COMP    234
                  END)                                                           COMP    234
        END;                                                          000030     COMP    235
BEGIN                                                                            COMP    235
 WITH STRUCTURES, PC DO                                                          COMP    235
  BEGIN NOOP;                                                                    COMP    235
  WHILE RCP < 15 DO                                                              COMP    235
    BEGIN RBUF := RBUF*16; RCP := RCP + 1 END;                                   COMP    235
   WITH CSEGP↑ DO                                                                COMP    235
    BEGIN CODE[CIX] := CBUF; RCODE[RCIX] := RBUF END;                            COMP    235
   (* REVERSE LIST OF CODE SEGMENTS *)                                           COMP    235
   SEGP1 := NIL;                                                                 COMP    235
   REPEAT                                                                        COMP    236
    WITH CSEGP↑ DO                                                               COMP    236
     BEGIN SEGP2 := NXTSEG; NXTSEG := SEGP1;                                     COMP    236
      SEGP1 := CSEGP; CSEGP := SEGP2                                             COMP    236
     END                                                                         COMP    236
   UNTIL CSEGP = NIL;                                                            COMP    236
   IDW.CN := 40B; IDW.LR := 1; IDW.WC := 20B;                                    COMP    236
   RCMAX := RCODEMAX; DISP := 15;                                                COMP    236
   WITH PC DO                                                                    COMP    236
   FOR K := 1 TO SIX DO                                                          COMP    236
    BEGIN LCIX := 1;                                                             COMP    237
     IF K = SIX THEN RCMAX := RCIX;                                              COMP    237
     FOR I := 1 TO RCMAX DO                                                      COMP    237
      BEGIN IDW.L := CADDR;                                                      COMP    237
       IF (K = SIX) AND (I = RCMAX) THEN                                         COMP    237
        BEGIN J := CIX MOD 15;                                                   COMP    237
         IF J <> 0 THEN                                                          COMP    237
          BEGIN DISP := J; IDW.WC := J + 1 END                                   COMP    237
        END;                                                                     COMP    237
       LGO↑ := CVAL; PUT(LGO);                                                   COMP    237
       WITH SEGP1↑ DO                                                            COMP    238
        BEGIN LGO↑ := RCODE[I]; PUT(LGO);                                        COMP    238
         FOR L := LCIX TO LCIX + DISP - 1 DO                                     COMP    238
          BEGIN LGO↑ := CODE[L]; PUT(LGO) END;                                   COMP    238
         CADDR := CADDR + DISP; LCIX := LCIX + 15                                COMP    238
        END                                                                      COMP    238
      END;                                                                       COMP    238
     SEGP1 := SEGP1↑.NXTSEG                                                      COMP    238
    END (*FOR K*)                                                                COMP    238
  END (*WITH*)                                                                   COMP    238
END (*LGOTEXT*) ;                                                                COMP    239

PROCEDURE LGOEND;                                                                COMP    239
 TYPE STYP = (WORD,ADRS,HLFS,NAMS);                                              COMP    239
    B18 = 0..777777B; B30 = 0..7777777777B;                                      COMP    239
    HALFS= PACKED RECORD LH: B30; RH: B30 END;                                   COMP    239
 VAR PAR: BOOLEAN;                                                               COMP    239
    STRUCTURES: RECORD CASE STYP OF                                              COMP    239
          WORD: (CVAL: INTEGER);                                                 COMP    239
          HLFS: (HS: HALFS);                                                     COMP    239
          ADRS: (IDW: PACKED RECORD CN: 0..63;                                   COMP    240
                                WC: B18; LR: B18; L: B18                         COMP    240
                  END);                                                          COMP    240
          NAMS: (CNAM: ALFA)                                                     COMP    240
         END;                                                                    COMP    240
    BUFF: RECORD CASE BOOLEAN OF                                                 COMP    240
        TRUE:  (BUF0: INTEGER);                                                  COMP    240
        FALSE: (BHS: HALFS)                                                      COMP    240
       END;                                                                      COMP    240
    WORDCNT: INTEGER;                                                            COMP    240

 PROCEDURE EXTTOLGO(PTR: EXTIDP);                                                COMP    241
  BEGIN (* PTR <> NIL *)                                                         COMP    241
   WITH PTR↑,BUFF,STRUCTURES DO                                                  COMP    241
    BEGIN                                                                        COMP    241
     IF L <> NIL THEN EXTTOLGO(L);                                               COMP    241
     IF R <> NIL THEN EXTTOLGO(R);                                               COMP    241
     CNAM := ID;                                                                 COMP    241
     IF PAR THEN LGO↑ := CVAL ELSE                                               COMP    241
      BEGIN BHS.RH:= HS.LH; LGO↑ := BUF0; BHS.LH := HS.RH END;                   COMP    241
     PUT(LGO);                                                                   COMP    242
     WHILE REF <> NIL DO WITH REF↑ DO                                            COMP    242
      BEGIN                                                                      COMP    242
      IF PAR THEN BHS.LH := LOC                                                  COMP    242
      ELSE BEGIN BHS.RH := LOC; LGO↑ := BUF0; PUT(LGO) END;                      COMP    242
      PAR := NOT PAR; REF := LINK                                                COMP    242
      END                                                                        COMP    242
    END                                                                          COMP    242
```

```
  END; (* EXTTOLGO *)                                                                  COMP  242
                                                                                       COMP  242
BEGIN CODEADDR:= CODEADDR + IC;                                                        COMP  243
 WITH STRUCTURES,BUFF DO                                          000031               COMP  243
 BEGIN                                                                                 COMP  243
  IF EXTROOT <> NIL THEN                                                               COMP  243
   BEGIN WORDCNT := EXTIDX + (EXTRX + 1) DIV 2;                                        COMP  243
   IF WORDCNT >= 10000B THEN ERROR(256)                                                COMP  243
   ELSE                                                                                COMP  243
    BEGIN CVAL := 0; IDW.CN := 44B; IDW.WC := WORDCNT;                                 COMP  243
     LGO↑ := CVAL; PUT(LGO);                                                           COMP  243
     PAR := TRUE;                                                                      COMP  243
     EXTTOLGO(EXTROOT);                                                                COMP  244
     IF NOT PAR THEN                                                                   COMP  244
      BEGIN BHS.RH := 0; LGO↑ := BUF0; PUT(LGO) END                                    COMP  244
    END                                                                                COMP  244
   END;                                                                                COMP  244
  CVAL:= 0; IDW.CN:= 46B; IDW.WC:= 1;                                (*XFER*)          COMP  244
  LGO↑:= CVAL; PUT(LGO);                                                               COMP  244
  IF LEVEL=1 THEN SHORTNAME(PROGNAME,CNAM)                                             COMP  244
  ELSE CNAM := ' ';                                                                    COMP  244
  IDW.L:= 0; LGO↑:= CVAL; PUT(LGO);                                                    COMP  244
  PUTSEG(LGO);                                                                         COMP  245
 END;                                                                                  COMP  245
END (*LGOEND*);                                                                        COMP  245
                                                                                       COMP  245
PROCEDURE SEARCHEXTID(FNAME: ALFA);                                                    COMP  245
 (* RETURNS POINTER TO FNAME-ENTRY IN EXT *)                                          COMP  245
                                                                                       COMP  245
 PROCEDURE ALLOCID;                                                                    COMP  245
  BEGIN NEW(EXT);                                                                      COMP  245
   WITH EXT↑ DO                                                                        COMP  246
    BEGIN                                                                              COMP  246
    L := NIL; R := NIL; REF := NIL; ID := FNAME;                                       COMP  246
     EXTIDX := EXTIDX + 1                                                              COMP  246
    END                                                                                COMP  246
  END;                                                                                 COMP  246
                                                                                       COMP  246
 BEGIN SHORTNAME(FNAME,FNAME);                                                         COMP  246
 IF EXTROOT = NIL THEN                                                                 COMP  246
  BEGIN ALLOCID; EXTROOT := EXT END                                                    COMP  246
 ELSE                                                                                  COMP  246
  BEGIN EXT := EXTROOT;                                                                COMP  247
   WHILE EXT↑.ID <> FNAME DO WITH EXT↑ DO                                              COMP  247
    IF ID < FNAME THEN                                                                 COMP  247
     IF R = NIL THEN BEGIN ALLOCID; R := EXT END ELSE EXT := R                         COMP  247
    ELSE                                                                               COMP  247
     IF L = NIL THEN BEGIN ALLOCID; L:= EXT END ELSE EXT := L                          COMP  247
  END                                                                                  COMP  247
 END;                                                                                  COMP  247
                                                                                       COMP  247
PROCEDURE CLEARREGS;                                                                   COMP  247
 VAR I: INTEGER;                                                                       COMP  248
 BEGIN                                                                                 COMP  248
 FOR I := 0 TO 7 DO                                                                    COMP  248
  BEGIN                                                                                COMP  248
  XRGS[I].XCONT := AVAIL; ARGS[I].ACONT := UNSPECADDR;                                 COMP  248
  IF I IN [3,7] THEN BRGS[I].BCONT := FREE                                             COMP  248
  ELSE BRGS[I].BCONT := SPECPURP                                                       COMP  248
  END;                                                                                 COMP  248
 LEVELS := [0,1,LEVEL]                                                                 COMP  248
 END; (* CLEARREGS *)                                                                 COMP  248
                                                                                       COMP  249
PROCEDURE RJTOEXT(FNAME: ALFA);                                                        COMP  249
 BEGIN                                                                                 COMP  249
  SEARCHEXTID(FNAME); CLEARREGS;                                                       COMP  249
  IF PC.CP = 3 THEN NOOP;                                                              COMP  249
  GEN30(018,0,0,0,0); NOOP                                                             COMP  249
 END; (* RJTOEXT *)                                                                   COMP  249
                                                                                       COMP  249
PROCEDURE EQTOEXT(FNAME:ALFA);                                                         COMP  249
 BEGIN                                                                                 COMP  249
  SEARCHEXTID(FNAME); GEN30(04B,0,0,0,0); NOOP;                                        COMP  250
 END; (* EQTOEXT *)                                                                   COMP  250
                                                                                       COMP  250
PROCEDURE ENTERCST(FCSTP: CTAILP);                                                     COMP  250
 (*ENTER CONST POINTED AT BY FCSTP INTO CONSTANT TABLE AND CHAIN                       COMP  250
 ACTUAL OCCURENCE IN CODE (AT <CIX,CP>) WITH EARLIER OCCURENCE*)                       COMP  250
 LABEL 1,2;                                                                            COMP  250
 VAR LCSP: CSP; P1,P2: CTAILP; LFSTOCC: LOCOFREF;                                      COMP  250
BEGIN LCSP := FSTCSP;                                                                  COMP  250
```

```
  WHILE LCSP <> NIL DO                                                                  COMP      250
   BEGIN P1 := LCSP↑.CSTP; P2 := FCSTP;                                                 COMP      251
    WHILE (P1 <> NIL)AND (P2 <> NIL) DO                                                 COMP      251
     BEGIN IF P1↑.CSVAL <> P2↑.CSVAL THEN GOTO 1;                    000032              COMP      251
      P1 := P1↑.NXTCSP; P2 := P2↑.NXTCSP                                                COMP      251
     END;                                                                               COMP      251
    IF P1 = P2 THEN GOTO 2;                                                             COMP      251
1: LCSP := LCSP↑.NXTCSP                                                                 COMP      251
   END;                                                                                 COMP      251
  (*NEW ENTRY:*)                                                                        COMP      251
  NEW(LCSP);                                                                            COMP      251
  WITH LCSP↑ DO                                                                         COMP      252
   BEGIN NXTCSP := FSTCSP; CSTP := FCSTP; CREF := NIL END;                              COMP      252
  FSTCSP := LCSP;                                                                       COMP      252
2:(*CHAIN OCCURENCES:*)                                                                 COMP      252
  LFSTOCC := LCSP↑.CREF; LINKOCC(LFSTOCC);                                              COMP      252
  LCSP↑.CREF := LFSTOCC                                                                 COMP      252
END (*ENTERCST*) ;                                                                      COMP      252
                                                                                        COMP      252
PROCEDURE OPENFILES(FCP: CTP);                                                          COMP      252
  LABEL 1;                                                                              COMP      252
  VAR EXTFILE: BOOLEAN; LCSP: CTAILP;                                                   COMP      253
                                                                                        COMP      253
  PROCEDURE OPENFL(FSP: STP; FADDR: ADDRRANGE);                                         COMP      253
    VAR I,LMIN,LMAX: INTEGER; LCP: CTP;                                                 COMP      253
  BEGIN                                                                                 COMP      253
    IF FSP <> NIL THEN                                                                  COMP      253
     WITH FSP↑ DO                                                                       COMP      253
      IF FTYPE THEN                                                                     COMP      253
      CASE FORM OF                                                                      COMP      253
       RECORDS:                                                                         COMP      253
        BEGIN LCP := FSTFLD;                                                            COMP      254
         WHILE LCP <> NIL DO                                                            COMP      254
          WITH LCP↑ DO                                                                  COMP      254
           BEGIN                                                                        COMP      254
            IF FILECNT < 36 THEN OPENFL(IDTYPE,FADDR+FLDADDR)                           COMP      254
            ELSE ERROR(258);                                                            COMP      254
            LCP:= NEXT                                                                  COMP      254
           END                                                                          COMP      254
        END;                                                                            COMP      254
       ARRAYS:                                                                          COMP      254
        IF INXTYPE <> NIL THEN                                                          COMP      255
         BEGIN GETBOUNDS(INXTYPE,LMIN,LMAX);                                            COMP      255
          IF FILECNT + LMAX - LMIN < 36 THEN                                            COMP      255
           FOR I := LMIN TO LMAX DO                                                     COMP      255
            BEGIN OPENFL(AELTYPE,FADDR);                                                COMP      255
             IF AELTYPE <> NIL THEN                                                     COMP      255
              FADDR := FADDR + AELTYPE↑.SIZE.WORDS                                      COMP      255
            END                                                                         COMP      255
          ELSE ERROR(258)                                                               COMP      255
         END;                                                                           COMP      255
       FILES:                                                                           COMP      256
        BEGIN LOCFILES := TRUE;                                                         COMP      256
         IF TEXTFILE THEN I := FADDR + CHEFET                                           COMP      256
         ELSE I := FADDR + BINEFET;                                                     COMP      256
         IF (FCP <> INPUTPTR)AND (FCP <> OUTPUTPTR) THEN                                COMP      256
          BEGIN NEW(CLSP); (* CLOSE LIST ENTRY *)                                      COMP      256
           WITH CLSP↑ DO                                                                COMP      256
            BEGIN NXTP := FCLSP; EFETADDR := I END;                                     COMP      256
           FCLSP := CLSP;                                                               COMP      256
          END;                                                                          COMP      256
         GEN30(51B,0,BRG[LEVEL],I,0); (*EFET ADDRESS*)                                  COMP      257
           IF TEXTFILE THEN I := 20B ELSE I := 0;                                       COMP      257
           IF SEGFILE THEN I := I + 100B;                                               COMP      257
           IF EXTFILE THEN                                                              COMP      257
            BEGIN                                                                       COMP      257
             IF EXFILP↑.READONLY THEN I := I + 1 ELSE I := I+3;                         COMP      257
             SHORTNAME(EXFILP↑.FILENAME, ALFINT.A);                                     COMP      257
            END                                                                         COMP      257
           ELSE                                                                         COMP      257
            IF FILECNT = 36 THEN ERROR(258)                                             COMP      257
            ELSE                                                                        COMP      258
             BEGIN FILECNT := FILECNT + 1;                                              COMP      258
              PASCF[7] := CHR(FILECNT);                                                 COMP      258
              ALFINT.A := PASCF                                                         COMP      258
             END;                                                                       COMP      258
         GEN30(71B,1,0,I,0); (*DISPOSITION CODE*)                                       COMP      258
         GEN30(51B,2,0,0,2);  (*(FORMAL) LFN*)                                          COMP      258
         NEW(LCSP);                                                                     COMP      258
         WITH LCSP↑ DO                                                                  COMP      258
          BEGIN NXTCSP := NIL; CSVAL := ALFINT.I END;                                   COMP      258
```

```
        ENTERCST(LCSP);                                                          COMP      25
         IF EXTFILE THEN                                                         COMP      25
          BEGIN                                                                  COMP      25
          GEN15(74B,6,0,1); GEN15(12B,6,6,2);                                    COMP      25
          GEN30(51B,5,0,EXFILP↑.SYSLOC,0);                                       COMP      25
          GEN15(43B,4,5,2); GEN15(11B,5,4,5); NOOP;                              COMP      25
          GEN15(54B,6,5,0); GEN30(03B,0,5,IC,2); GEN15(10B,2,5,5);               COMP      25
          END;                                                                   COMP      25
        IF TEXTFILE THEN I := SIZE.WORDS - CHEFETSZ                              COMP      25
        ELSE I := SIZE.WORDS - EFETSZ;                                           COMP      25
        GEN30(71B,3,0,I,0); (*BUFFER SIZE*)                                      COMP      26
        IF FILTYPE <> NIL THEN                                                   COMP      26
         GEN30(71B,6,0,FULLWORDS(FILTYPE↑.SIZE),0); (*LRL*)                      COMP      26
        IF TEXTFILE THEN I := FADDR + CHEFETSZ                                   COMP      26
        ELSE I := FADDR + EFETSZ;                                                COMP      26
        GEN30(71B,7,BRG[LEVEL],I,0); (*BUFFER ADDRESS*)                          COMP      26
        RJTOEXT(EP,OPEN    );                                                    COMP      26
       END (*FILES*)                                                             COMP      26
     END (*CASE*)                                                                COMP      26
 END (*OPENFL*) ;                                                                COMP      26
                                                                                 COMP      26
BEGIN (*OPENFILES*)                                                              COMP      26
 IF FCP <> NIL THEN                                                              COMP      26
  WITH FCP↑ DO                                                                   COMP      26
   BEGIN OPENFILES(LLINK); OPENFILES(RLINK);                                     COMP      26
    IF (KLASS = VARS)AND (VKIND = DRCT) THEN                                     COMP      26
     BEGIN EXTFILE := FALSE;                                                     COMP      26
      IF (LEVEL = 1)AND (IDTYPE <> NIL) THEN                                     COMP      26
       IF IDTYPE↑.FORM = FILES THEN                                              COMP      26
        BEGIN EXFILP := FEXFILP;                                                 COMP      26
          WHILE EXFILP <> NIL DO                                                 COMP      26
          WITH EXFILP↑ DO                                                        COMP      26
           BEGIN                                                                 COMP      26
            IF FILENAME = NAME THEN                                              COMP      26
             BEGIN EXTFILE := TRUE; DECLARED := TRUE;                            COMP      26
              GOTO 1                                                             COMP      26
             END;                                                                COMP      26
            EXFILP := NXTP                                                       COMP      26
           END;                                                                  COMP      26
 1:      END;                                                                    COMP      26
       OPENFL(IDTYPE,VADDR)                                                      COMP      26
     END                                                                         COMP      26
   END (*WITH*)                                                                  COMP      26
END (*OPENFILES*);                                                               COMP      26
                                                                                 COMP      26
PROCEDURE ROTATEX(FI: REGNR);                                                    COMP      26
 (*IF X-FI IS SHIFTED, SHIFT IT BACK*)                                           COMP      26
BEGIN                                                                            COMP      26
 WITH XRGS[FI] DO                                                                COMP      26
  IF XCONT IN [SIMPVAR,INDVAR] THEN                                              COMP      26
   IF SHFTCNT <> 0 THEN                                                          COMP      26
    BEGIN GEN15(20B,FI,0,WORDSIZE-SHFTCNT); SHFTCNT := 0                         COMP      26
    END                                                                          COMP      26
END (*ROTATEX*);                                                                 COMP      26
                                                                                 COMP      26
PROCEDURE DECREFX(FI: REGNR);                                                    COMP      26
 (*DECREASE NUMBER OF REFERENCES TO X-FI BY ONE*)                                 COMP      26
BEGIN                                                                            COMP      26
 WITH XRGS[FI] DO                                                                COMP      26
  CASE XCONT OF                                                                  COMP      26
   AVAIL: ;                                                                      COMP      26
   SHRTCST,LONGCST,                                                              COMP      26
   SIMPVAR,INDVAR:                                                               COMP      26
    IF REFNR > 0 THEN                                                            COMP      26
     BEGIN REFNR := REFNR - 1;                                                   COMP      26
      IF REFNR = 0 THEN LASTREF := IC                                            COMP      26
     END;                                                                        COMP      26
   OTHER:                                                                        COMP      26
    IF REFNR > 0 THEN                                                            COMP      26
     BEGIN REFNR := REFNR - 1;                                                   COMP      26
      IF REFNR = 0 THEN XCONT := AVAIL                                           COMP      26
     END                                                                         COMP      26
  END                                                                            COMP      26
END (*DECREFX*) ;                                                                COMP      26
                                                                                 COMP      26
PROCEDURE BX1XJ(FI,FJ: REGNR);                                                   COMP      26
 (*AVOID GENERATION OF B XI XJ INSTRUCTIONS WHENEVER APPROPRIATE BY               COMP      26
  ALTERING PREVIOUSLY GENERATED INSTRUCTION*)                                    COMP      26
 VAR I: REGNR;                                                                   COMP      26
BEGIN XRGS[FI] := XRGS[FJ];                                                      COMP      26
 IF FI <> FJ THEN                                                                COMP      26
```

```
  IF (LASTI = FJ)AND ((LASTOP >= 10B)AND (LASTOP <= 47B)AND
   NOT (LASTOP IN [20B,21B,43B,46B]) OR (LASTOP >= 70B))
    AND (XRGS[FJ].REFNR <= 1) THEN
    BEGIN
     IF (LASTOP < 70B)OR (LASTOP >= 73B) THEN
      CBUF := CBUF - (LASTI - FI)*100B
     ELSE
      CBUF := CBUF - (LASTI - FI)*10000B*1000B;
     LASTI := FI; XRGS[FJ].XCONT := AVAIL
    END
   ELSE
    BEGIN GEN15(10B,FI,FJ,0); DECREFX(FJ); XRGS[FI].REFNR := 1;
     WITH XRGS[FJ] DO
      IF XCONT = INDVAR THEN
       WITH XRGS[XREG] DO  REFNR := REFNR + 1;
    END;
END (*BXIXJ*) ;

PROCEDURE SAVEREFXRGS(VAR FXRGS: XRGSTATUS);
 VAR I,J,K: REGNR; LXRGS: XRGSTATUS;
BEGIN LXRGS:=XRGS; CLEARREGS;
 FOR I:=0 TO 7 DO
  WITH LXRGS[I] DO
   IF XCONT <> AVAIL THEN
    IF XCONT = INDVAR THEN
     BEGIN WITH LXRGS[XREG] DO
            BEGIN REFNR:=REFNR - 1;
             IF REFNR = 0 THEN XCONT:=AVAIL
            END;
           IF REFNR = 0 THEN XCONT:=AVAIL ELSE XCONT:=OTHER
     END
    ELSE IF REFNR = 0 THEN XCONT:=AVAIL;
 K:=0;
 FOR I:=6 TO 7 DO
  WITH LXRGS[I] DO
   IF XCONT <> AVAIL THEN
    BEGIN
     IF K=0 THEN GEN30(51B,I,BRG[LEVEL],LC,0)
            ELSE GEN15(54B,7,6,1);
     K:=K+1; J:=I
    END;
 FOR I:=0 TO 5 DO
  WITH LXRGS[I] DO
   IF XCONT <> AVAIL THEN
    BEGIN GEN15(10B,7,I,0);
     IF K=0 THEN GEN30(51B,7,BRG[LEVEL],LC,0)
            ELSE GEN15(54B,7,J,1);
     K:=K+1; J:=7
    END;
 LC := LC + K;
 IF LC > LCMAX THEN LCMAX := LC;
 FXRGS := LXRGS
END (*SAVEREFXRGS*) ;

PROCEDURE RELOADREFXRGS(VAR FXRGS: XRGSTATUS);
 VAR I,J,K,L,M: REGNR; LPL: PLACE;
BEGIN K := 0; M := 0;
 FOR I := 0 TO 7 DO
  BEGIN J := (I+6) MOD 8;
   WITH FXRGS[J] DO
    IF XCONT <> AVAIL THEN
     IF REFNR <> 0 THEN
      BEGIN IF I <= 2 THEN L := 5 ELSE L := J;
       IF K = 0 THEN
        BEGIN GEN30(51B,L,BRG[LEVEL],0,0); LPL := PC;
        END
       ELSE GEN15(54B,L,K,1);
       IF I <= 2 THEN GEN15(10B,J,5,0);
       XRGS[J] := FXRGS[J];
       K := L; M := M + 1
      END
  END;
 IF M <> 0 THEN
  BEGIN LC := LC - M; INS(LC,LPL) END;
END (*RELOADXRGS*) ;

PROCEDURE NEEDB(VAR FI: REGNR);
 (*RETURN INDEX OF AVAILABLE B-REGISTER*)
 (*ASSUMES BRGS[I].BCONT <> SPECPURP FOR SOME 1 <= I <= 7*)
 LABEL 1;
 VAR MAXLEV: LEVRANGE; I,NR: REGNR;
```

```
BEGIN MAXLEV := 0; NR := 0;
 FOR I := 1 TO 7 DO
  WITH BRGS[I] DO
   CASE BCONT OF
    FREE:
     BEGIN NR := I; GOTO 1 END;
    BASADDR:
     IF BLEV > MAXLEV THEN
      BEGIN MAXLEV := BLEV; NR := I END;
    SPECPURP:
   END;
 LEVELS := LEVELS-[MAXLEV];
1:BRGS[NR].BCONT := SPECPURP; FI := NR
END (*NEEDB*) ;

PROCEDURE NEEDX(FLOW,FHIGH: REGNR; VAR FI: REGNR);
 (*RETURN INDEX FI (FLOW <= FI <= FHIGH) OF AVAILABLE X-REG.;
  DON=T TOUCH ANY X-REG. CONTENTS*)
 LABEL 1;
 VAR I,NR: REGNR; PR,MAXPR: INTEGER; FIRSTTIME: BOOLEAN;
BEGIN MAXPR := 0; NR := FLOW; FIRSTTIME := TRUE;
 REPEAT
 FOR I := FLOW TO FHIGH DO
  WITH XRGS[I] DO
   IF XCONT = AVAIL THEN
    BEGIN NR := I; GOTO 1 END
   ELSE
    IF XCONT <> OTHER THEN
     IF REFNR = 0 THEN
      BEGIN PR := IC - LASTREF + BONUS[XCONT];
       IF PR > MAXPR THEN
        BEGIN MAXPR := PR; NR := I END
      END;
 IF MAXPR = 0 THEN
  IF FIRSTTIME THEN
   BEGIN
    FOR I := 0 TO 7 DO
     WITH XRGS[I] DO
      IF XCONT = INDVAR THEN
       BEGIN DECREFX(XREG);
        IF REFNR = 0 THEN XCONT := AVAIL
        ELSE
         BEGIN IF SHFTCNT <> 0 THEN GEN15(20B,I,0,WORDSIZE-SHFTCNT);
          XCONT := OTHER
         END
       END;
    FIRSTTIME := FALSE
   END
  ELSE
   BEGIN IF FLOW <> FHIGH THEN ERROR(259); MAXPR := 1 END
 UNTIL MAXPR > 0;
1:WITH XRGS[NR] DO
  BEGIN
   IF XCONT = INDVAR THEN DECREFX(XREG)
   ELSE
     FOR I := 1 TO 7 DO
      WITH ARGS[I] DO
       IF ACONT = INDADDR THEN
        IF AREG = NR THEN ACONT := UNSPECADDR;
    XCONT := OTHER; REFNR := 1
   END;
 FI := NR
END (*NEEDX*) ;

PROCEDURE LOADBASE(FLEV: LEVRANGE; FB: BOOLEAN; VAR FI: REGNR);
 (*LOAD BASE ADDRESS OF DATA SEGMENT WITH LEVEL FLEV INTO X-FI*)
 LABEL 1;
 VAR L: REGNR; I,J,K:LEVRANGE;
BEGIN NEEDX(1,5,I);
 FOR J := FLEV + 1 TO LEVEL DO
  IF J IN LEVELS THEN
   BEGIN K := J - 1;
    GEN15(56B,I,BRG[J],0);
    (*LOOP UNTIL K = FLEV:*)
    WHILE K <> FLEV DO
     BEGIN GEN15(53B,I,I,0);
     K := K - 1
     END;
    IF FB
     THEN BEGIN NEEDB(L);
      WITH BRGS[L] DO
```

```
          BEGIN BCONT := BASADDR;                                                    COMP     28
          BLEV := K                                                                  COMP     28
          END;                                                                       COMP     28
        GEN15(63B,L,I,0);                                          000036          COMP     28
        BRG[K] := L;                                                               COMP     28
        LEVELS := LEVELS + [K]                                                     COMP     28
        END;                                                                       COMP     28
      ARGS[I].ACONT := UNSPECADDR;FI := I;                                         COMP     28
      GOTO 1                                                                       COMP     28
    END;                                                                           COMP     28
1:END (*LOADBASE*) ;                                                               COMP     28
                                                                                   COMP     28
 PROCEDURE LOADADDRESS(VAR FATTR: ATTR; VAR FI: REGNR);                            COMP     28
  (*LOAD WORD-ADDRESS OF FATTR INTO X-FI*)                                         COMP     28
  LABEL 1,2;                                                                       COMP     28
  VAR I,J: REGNR;                                                                  COMP     28
 BEGIN                                                                             COMP     28
  WITH FATTR DO                                                                    COMP     28
   BEGIN                                                                           COMP     28
   IF TYPTR <> NIL THEN                                                            COMP     28
    CASE KIND OF                                                                   COMP     28
    CST:                                                                           COMP     28
      (*MUST BE A STRING CONSTANT*)                                                COMP     28
      BEGIN NEEDX(0,7,I); GEN30(71B,I,0,0,2);                                      COMP     28
       IF STRING(TYPTR) THEN ENTERCST(CVAL.VALP)                                   COMP     28
      END;                                                                         COMP     28
    VARBL:                                                                         COMP     28
    CASE WORDACC OF                                                                COMP     28
     DRCT:                                                                         COMP     28
      BEGIN                                                                        COMP     28
       FOR J := 1 TO 7 DO                                                          COMP     28
        WITH ARGS[J] DO                                                            COMP     28
         IF ACONT = SIMPADDR THEN                                                  COMP     28
          IF (ALEV = VLEVEL)AND (AADDR = CWDISPL) THEN                             COMP     28
           BEGIN NEEDX(0,7,I); GEN15(74B,I,J,0); GOTO 1 END;                       COMP     28
       IF VLEVEL IN LEVELS THEN                                                    COMP     28
        BEGIN NEEDX(0,7,I); GEN30(71B,I,BRG[VLEVEL],CWDISPL,0)                     COMP     28
        END                                                                        COMP     28
       ELSE                                                                        COMP     28
        BEGIN LOADBASE(VLEVEL,TRUE,I); GEN30(72B,I,I,CWDISPL,0)                    COMP     28
        END;                                                                       COMP     28
1:   END;                                                                          COMP     28
     INDRCT:                                                                       COMP     28
      BEGIN                                                                        COMP     28
       IF CWDISPL = 0 THEN I := VWDISPL                                            COMP     28
       ELSE                                                                        COMP     28
        BEGIN DECREFX(VWDISPL); NEEDX(0,7,I);                                      COMP     28
         FOR J := 1 TO 7 DO                                                        COMP     28
          WITH ARGS[J] DO                                                          COMP     28
           IF ACONT = INDADDR THEN                                                 COMP     28
            IF (AREG = VWDISPL)AND (ADISPL = CWDISPL) THEN                         COMP     28
             BEGIN GEN15(74B,I,J,0); GOTO 2 END;                                   COMP     28
        IF CWDISPL = 1 THEN GEN15(73B,I,VWDISPL,1)                                 COMP     28
        ELSE GEN30(72B,I,VWDISPL,CWDISPL,0);                                       COMP     28
       END;                                                                        COMP     28
2:   END;                                                                          COMP     28
     INXD:                                                                         COMP     28
      BEGIN DECREFX(VWDISPL);                                                      COMP     28
       IF VLEVEL IN LEVELS THEN                                                    COMP     28
         BEGIN NEEDX(0,7,I); GEN30(72B,I,VWDISPL,CWDISPL,0);                       COMP     28
          GEN15(73B,I,1,BRG[VLEVEL])                                               COMP     28
         END                                                                       COMP     28
       ELSE                                                                        COMP     28
        BEGIN NEEDB(J); GEN30(62B,J,VWDISPL,CWDISPL,0);                            COMP     28
         LOADBASE(VLEVEL,TRUE,I); GEN15(73B,I,I,J);                                COMP     28
         BRGS[J].BCONT := FREE                                                     COMP     28
        END                                                                        COMP     28
      END                                                                          COMP     29
    END (*CASE*);                                                                  COMP     29
    COND,EXPR:                                                                     COMP     29
    NEEDX(0,7,I);                                                                  COMP     29
    END (*CASE*)                                                                   COMP     29
   ELSE                                                                            COMP     29
    NEEDX(0,7,I);                                                                  COMP     29
   WORDACC := INDRCT; VWDISPL := I; CWDISPL := 0                                   COMP     29
  END (*WITH FATTR*) ;                                                             COMP     29
  FI := I                                                                          COMP     29
 END (*LOADADDRESS*) ;                                                             COMP     29
                                                                                   COMP     29
 PROCEDURE LOAD(VAR FATTR: ATTR; VAR FI: REGNR);                                   COMP     29
  (*LOAD FATTR INTO X-FI*)                                                         COMP     29
```

```
  LABEL 1,2,4,5,6;                                                                    COMP     29
  VAR I,J,K: REGNR; SHRT,SIMPIND: BOOLEAN;                                            COMP     29
    BITSZ,SHIFT,MASK: BITRANGE; LADDR: INTEGER;                          000037      COMP     29
    SVAL: SHRTINT; CSHFT: INTEGER; LCSP: CTAILP;                                      COMP     29
    LCST: INTEGER; LMODE: (USRADJ,SRADJ,USLADJ);                                      COMP     29
BEGIN                                                                               COMP     29
 WITH FATTR DO                                                                      COMP     29
  BEGIN                                                                             COMP     29
  IF TYPTR <> NIL THEN                                                              COMP     29
   CASE KIND OF                                                                     COMP     29
    CST:                                                                            COMP     29
     BEGIN SHRT := FALSE; SVAL := 0; LCSP := NIL;                                   COMP     29
      IF STRING(TYPTR) THEN LCSP := CVAL.VALP                                       COMP     29
      ELSE                                                                          COMP     29
       BEGIN LCST:=CVAL.IVAL; (* INTENAL VALUE OF CONSTANT *)                       COMP     29
        IF ABS(LCST) < TWOTO17 THEN                                                 COMP     29
         BEGIN SVAL := LCST; SHRT := TRUE END                                       COMP     29
        ELSE                                                                        COMP     29
         BEGIN NEW(LCSP);                                                           COMP     29
          WITH LCSP↑ DO                                                             COMP     29
           BEGIN NXTCSP := NIL; CSVAL := LCST END                                   COMP     29
         END                                                                        COMP     29
      END;                                                                          COMP     29
      IF SHRT THEN                                                                  COMP     29
       BEGIN                                                                        COMP     29
        FOR I := 0 TO 7 DO                                                          COMP     29
         WITH XRGS[I] DO                                                            COMP     29
          IF XCONT = SHRTCST THEN                                                   COMP     29
           IF CSTVAL = SVAL THEN                                                    COMP     29
            BEGIN REFNR := REFNR + 1; GOTO 1 END                                    COMP     29
       END                                                                          COMP     29
      ELSE                                                                          COMP     29
       FOR I := 0 TO 7 DO                                                           COMP     29
        WITH XRGS[I] DO                                                             COMP     29
         IF XCONT = LONGCST THEN                                                    COMP     29
          IF CPTR↑.CSVAL = LCSP↑.CSVAL THEN                                         COMP     29
           BEGIN REFNR := REFNR + 1; GOTO 1 END;                                    COMP     29
      IF SHRT THEN                                                                  COMP     29
       BEGIN NEEDX(0,7,I);                                                          COMP     29
        WITH XRGS[I] DO                                                             COMP     29
         BEGIN XCONT := SHRTCST; REFNR := 1; CSTVAL := SVAL END;                    COMP     29
        IF SVAL = 0 THEN GEN15(138,I,I,I)                                           COMP     29
        ELSE                                                                        COMP     29
         IF SVAL = 1 THEN GEN15(768,I,1,0)                                          COMP     29
         ELSE                                                                       COMP     29
          IF SVAL = 2 THEN GEN15(768,I,1,1)                                         COMP     29
          ELSE GEN30(718,I,0,SVAL,0)                                                COMP     29
       END                                                                          COMP     29
      ELSE                                                                          COMP     29
       BEGIN NEEDX(1,5,I);                                                          COMP     29
        WITH XRGS[I] DO                                                             COMP     29
         BEGIN XCONT := LONGCST; REFNR := 1; CPTR := LCSP END;                      COMP     29
        ARGS[I].ACONT := UNSPECADDR;                                                COMP     29
        GEN30(518,I,0,0,2);                                                         COMP     29
        ENTERCST(LCSP)                                                              COMP     29
       END;                                                                         COMP     29
1:  END;                                                                            COMP     29
    VARBL:                                                                          COMP     29
     BEGIN                                                                          COMP     29
      CASE WORDACC OF                                                               COMP     29
       DRCT:                                                                        COMP     29
        BEGIN                                                                       COMP     29
         FOR I := 0 TO 7 DO                                                         COMP     29
          WITH XRGS[I] DO                                                           COMP     29
           IF XCONT = SIMPVAR THEN                                                  COMP     29
            IF (XLEV = VLEVEL)AND (XADDR = CWDISPL) THEN                            COMP     29
             BEGIN REFNR := REFNR + 1; GOTO 4 END;                                  COMP     29
         FOR J := 1 TO 7 DO                                                         COMP     29
          WITH ARGS[J] DO                                                           COMP     29
           IF ACONT = SIMPADDR THEN                                                 COMP     29
            IF VLEVEL = ALEV THEN                                                   COMP     29
             BEGIN LADDR := CWDISPL - AADDR;                                        COMP     29
              IF ABS(LADDR) <= 1 THEN                                               COMP     29
               BEGIN NEEDX(1,5,I);                                                  COMP     29
                IF LADDR >= 0 THEN GEN15(548,I,J,LADDR)                             COMP     29
                ELSE GEN15(558,I,J,1);                                              COMP     29
                GOTO 2                                                              COMP     29
               END;                                                                 COMP     29
         IF VLEVEL IN LEVELS THEN                                                   COMP     29
          BEGIN NEEDX(1,5,I); GEN30(518,I,BRG[VLEVEL],CWDISPL,0)                    COMP     29
```

```
          END
        ELSE
          BEGIN LOADBASE(VLEVEL,TRUE,I);
           GEN30(52B,I,I,CWDISPL,0)
          END;
  2:    WITH ARGS[I] DO
          BEGIN ACONT := SIMPADDR; ALEV := VLEVEL;
           AADDR := CWDISPL
          END;
        WITH XRGS[I] DO
          BEGIN XCONT := SIMPVAR; REFNR := 1; VPADDR := FALSE;
           SHFTCNT := 0; XLEV := VLEVEL; XADDR := CWDISPL
          END;
  4:    END;
      INDRCT:
       BEGIN SIMPIND := XRGS[VWDISPL].XCONT = SIMPVAR;
        IF SIMPIND THEN
          BEGIN
           FOR I := 0 TO 7 DO
            WITH XRGS[I] DO
             IF XCONT = INDVAR THEN
              IF (XREG = VWDISPL)AND (XDISPL = CWDISPL) THEN
               BEGIN REFNR := REFNR + 1;
                DECREFX(VWDISPL); GOTO 6
               END;
           FOR J := 1 TO 7 DO
            WITH ARGS[J] DO
             IF ACONT = INDADDR THEN
              IF AREG = VWDISPL THEN
               BEGIN LADDR := CWDISPL - ADISPL;
                IF ABS(LADDR) <= 1 THEN
                 BEGIN NEEDX(1,5,I);
                  IF LADDR >= 0 THEN GEN15(54B,I,J,LADDR)
                  ELSE GEN15(55B,I,J,1);
                  GOTO 5
                 END
               END
          END
        ELSE
          DECREFX(VWDISPL);
        NEEDX(1,5,I);
        IF CWDISPL IN [0,1] THEN GEN15(53B,I,VWDISPL,CWDISPL)
        ELSE GEN30(52B,I,VWDISPL,CWDISPL,0);
  5:   IF SIMPIND THEN
          BEGIN
           WITH ARGS[I] DO
            BEGIN ACONT := INDADDR;
             AREG := VWDISPL; ADISPL := CWDISPL
            END;
           WITH XRGS[I] DO
            BEGIN XCONT := INDVAR; REFNR := 1;
             SHFTCNT := 0;
             XREG := VWDISPL; XDISPL := CWDISPL
            END
          END
        ELSE ARGS[I].ACONT := UNSPECADDR;
  6:   END;
      INXD:
       BEGIN DECREFX(VWDISPL);
        IF VLEVEL IN LEVELS THEN
          BEGIN NEEDX(1,5,I);
            IF PC.CP = 3 THEN
             BEGIN GEN15(73B,I,VWDISPL,BRG[VLEVEL]);
              GEN30(52B,I,I,CWDISPL,0)
             END
            ELSE
             BEGIN GEN30(72B,I,VWDISPL,CWDISPL,0);
              GEN15(53B,I,I,BRG[VLEVEL])
             END;
          END
        ELSE
          BEGIN NEEDB(J); GEN30(62B,J,VWDISPL,CWDISPL,0);
           LOADBASE(VLEVEL,TRUE,I); GEN15(53B,I,I,J);
           BRGS[J].BCONT := FREE
          END;
        ARGS[I].ACONT := UNSPECADDR
       END
     END (*CASE*) ;
     IF PCKD THEN
      BEGIN
       WITH TYPTR↑ DO
```

```
        BEGIN                                                                   COMP
          IF FORM = SUBRANGE THEN                                               COMP
            IF MIN.IVAL < 0 THEN LMODE := SRADJ                                 COMP
            ELSE LMODE := USRADJ                                                COMP
          ELSE                                                                  COMP
            IF FORM IN [ARRAYS,RECORDS] THEN LMODE := USLADJ                    COMP
            ELSE LMODE := USRADJ;                                               COMP
          BITSZ := SIZE.BITS                                                    COMP
        END;                                                                    COMP
       WITH XRGS[I] DO                                                          COMP
        IF XCONT IN [SIMPVAR,INDVAR] THEN SHIFT := SHFTCNT                      COMP
        ELSE SHIFT := 0;                                                        COMP
       IF LMODE = USLADJ THEN MASK := BITSZ                                     COMP
       ELSE MASK := WORDSIZE - BITSZ;                                           COMP
       CSHFT := CBDISPL - SHIFT;                                                COMP
       IF LMODE = USRADJ THEN CSHFT := CSHFT + BITSZ;                           COMP
       IF BITREG = XREG THEN                                                    COMP
        BEGIN                                                                   COMP
          IF SHIFT <> 0 THEN (*TO GUARANTEE 0 <= B-K <= 60*)                    COMP
           BEGIN GEN15(20B,I,0,WORDSIZE-SHIFT);                                 COMP
            XRGS[I].SHFTCNT := 0; CSHFT := CSHFT + SHIFT                        COMP
           END;                                                                 COMP
          NEEDB(K);                                                             COMP
          IF CSHFT IN [0,1] THEN GEN15(63B,K,VBDISPL,CSHFT)                     COMP
          ELSE GEN30(62B,K,VBDISPL,CSHFT,0);                                    COMP
          DECREFX(VBDISPL); DECREFX(I);                                         COMP
          NEEDX(0,7,J); GEN15(22B,J,K,I);                                       COMP
          BRGS[K].BCONT := FREE;                                                COMP
          IF LMODE = SRADJ THEN GEN15(21B,J,0,MASK)                             COMP
          ELSE                                                                  COMP
           BEGIN NEEDX(0,7,K); GEN15(43B,K,0,MASK);                             COMP
            IF LMODE = USRADJ THEN GEN15(15B,J,J,K)                             COMP
            ELSE GEN15(11B,J,J,K);                                              COMP
            DECREFX(K)                                                          COMP
           END;                                                                 COMP
          I := J                                                                COMP
        END                                                                     COMP
       ELSE                                                                     COMP
        BEGIN IF CSHFT < 0 THEN CSHFT := CSHFT + WORDSIZE                       COMP
          ELSE                                                                  COMP
           IF CSHFT = WORDSIZE THEN CSHFT := 0;                                 COMP
          WITH XRGS[I] DO                                                       COMP
           IF XCONT IN [SIMPVAR,INDVAR] THEN                                    COMP
            IF LMODE = SRADJ THEN                                               COMP
             BEGIN NEEDX(0,7,J); DECREFX(I);                                    COMP
              GEN15(10B,J,I,0); I := J                                          COMP
              END                                                               COMP
            ELSE                                                                COMP
             SHFTCNT := (SHFTCNT + CSHFT) MOD WORDSIZE;                         COMP
          IF CSHFT <> 0 THEN GEN15(20B,I,0,CSHFT);                              COMP
          IF LMODE = SRADJ THEN GEN15(21B,I,0,MASK)                             COMP
          ELSE                                                                  COMP
           BEGIN NEEDX(0,7,J); GEN15(43B,J,0,MASK);                             COMP
            IF LMODE = USRADJ THEN GEN15(15B,J,I,J)                             COMP
            ELSE GEN15(11B,J,I,J);                                              COMP
            DECREFX(I); I := J                                                  COMP
           END                                                                  COMP
        END                                                                     COMP
      END (*PCKD*)                                                              COMP
     ELSE ROTATEX(I);                                                           COMP
    END;                                                                        COMP
   COND:                                                                        COMP
    BEGIN NEEDX(0,7,I);                                                         COMP
     IF CONDCD IN [ZR,NZ] THEN                                                  COMP
      BEGIN GEN15(13B,I,I,I); GEN15(37B,I,I,CDR); NEECX(0,7,K);                 COMP
       IF CONDCD = ZR THEN GEN15(13B,K,CDR,I)                                   COMP
       ELSE GEN15(17B,K,I,CDR);                                                 COMP
       GEN15(43B,I,0,59); GEN15(15B,I,K,I); DECREFX(K)                          COMP
      END                                                                       COMP
     ELSE                                                                       COMP
      BEGIN GEN15(43B,I,0,1);                                                   COMP
       IF CONDCD = PL THEN GEN15(11B,I,CDR,I)                                   COMP
       ELSE GEN15(15B,I,I,CDR);                                                 COMP
       GEN15(20B,I,0,1)                                                         COMP
      END;                                                                      COMP
     DECREFX(CDR)                                                               COMP
    END;                                                                        COMP
   EXPR:                                                                        COMP
    I := EXPREG                                                                 COMP
  END (*CASE*)                                                                  COMP
ELSE NEEDX(0,7,I);                                                              COMP
```

```
    KIND := EXPR; EXPREG := I                                                    COMP      31
    END (*WITH FATTR*) ;                                                         COMP      31
  FI := I                                                                        COMP      31
END (*LOAD*) ;                                                                   COMP      31

PROCEDURE STORE(VAR FATTR: ATTR; FI: REGNR);                        000040       COMP      31
 (*STORE X-FI AT FATTR*)                                                         COMP      31
 (*ASSUMES FATTR.KIND = VARBL*)                                                  COMP      31
 LABEL 1,2;                                                                      COMP      31
 VAR I,J,K,LNR: REGNR; LATTR: ATTR; LXRG: XRGSTAT;                               COMP      31
   BITSZ,SHIFT,MASK: BITRANGE; CSHFT: INTEGER; LCST: SHRTINT;                    COMP      31
   LCP: POSRANGE; LADDR: INTEGER; LFTADJ,LBX,LXFICST: BOOLEAN;                   COMP      31

BEGIN                                                                            COMP      31
 WITH FATTR DO                                                                   COMP      31
  IF TYPTR <> NIL THEN                                                           COMP      31
   BEGIN                                                                         COMP      31
    IF PCKD THEN                                                                 COMP      31
     BEGIN LATTR := FATTR;                                                       COMP      31
      IF WORDACC <> DRCT THEN                                                    COMP      31
       WITH XRGS[VWDISPL] DO REFNR := REFNR + 1;                                 COMP      31
      LATTR.PCKD := FALSE; LOAD(LATTR,I);                                        COMP      31
      WITH TYPTR↑ DO                                                             COMP      31
       BEGIN LFTADJ := FORM IN [ARRAYS,RECORDS];                                 COMP      31
        BITSZ := SIZE.BITS                                                       COMP      31
       END;                                                                      COMP      31
      WITH XRGS[I] DO                                                            COMP      31
       IF XCONT IN [SIMPVAR,INDVAR] THEN SHIFT := SHFTCNT                        COMP      31
       ELSE SHIFT := 0;                                                          COMP      31
      IF LFTADJ THEN                                                             COMP      31
       BEGIN MASK := BITSZ; CSHFT := CBDISPL - SHIFT END                         COMP      31
      ELSE                                                                       COMP      31
       BEGIN MASK := WORDSIZE - BITSZ;                                           COMP      31
        CSHFT := CBDISPL - SHIFT + BITSZ                                         COMP      31
       END;                                                                      COMP      31
      IF BITREG = XREG THEN                                                      COMP      31
       BEGIN                                                                     COMP      31
        IF BITSZ < SHIFT THEN (*TO GUARANTEE 0 <= 8-K <= 60*)                    COMP      31
         BEGIN GEN15(20B,I,0,WORDSIZE - SHIFT);                                  COMP      31
          XRGS[I].SHFTCNT := 0; CSHFT := CSHFT + SHIFT                           COMP      31
         END;                                                                    COMP      31
        NEEDB(K);                                                                COMP      31
        IF CSHFT IN [0,1] THEN GEN15(63B,K,VBDISPL,CSHFT)                        COMP      31
        ELSE GEN30(62B,K,VBDISPL,CSHFT,0);                                       COMP      32
        DECREFX(VBDISPL); DECREFX(I); NEEDX(0,7,J);                              COMP      32
        GEN15(22B,J,K,I);                                                        COMP      32
       END                                                                       COMP      32
      ELSE                                                                       COMP      32
       BEGIN                                                                     COMP      32
        IF CSHFT < 0 THEN CSHFT := CSHFT + WORDSIZE                              COMP      32
        ELSE IF CSHFT = WORDSIZE THEN CSHFT := 0;                                COMP      32
        IF CSHFT <> 0 THEN GEN15(20B,I,0,CSHFT); J := I;                         COMP      32
        WITH XRGS[I] DO                                                          COMP      32
         IF XCONT IN [SIMPVAR,INDVAR] THEN                                       COMP      32
          BEGIN SHFTCNT := (SHFTCNT + CSHFT) MOD WORDSIZE;                       COMP      32
           CSHFT := 0                                                            COMP      32
          END                                                                    COMP      32
       END;                                                                      COMP      32
      NEEDX(0,7,I); GEN15(43B,I,0,MASK);                                         COMP      32
      IF LFTADJ THEN                                                             COMP      32
       BEGIN GEN15(15B,J,J,I); GEN15(11B,I,FI,I) END                             COMP      32
      ELSE                                                                       COMP      32
       BEGIN GEN15(11B,J,J,I); GEN15(15B,I,FI,I) END;                            COMP      32
      DECREFX(FI); GEN15(12B,J,J,I);                                             COMP      32
      IF BITREG = XREG THEN                                                      COMP      32
       BEGIN GEN30(61B,K,K,-WORDSIZE,0); GEN15(23B,J,K,J);                       COMP      32
        BRGS[K].BCONT := FREE                                                    COMP      32
       END                                                                       COMP      32
      ELSE IF CSHFT <> 0 THEN GEN15(20B,J,0,WORDSIZE-CSHFT);                     COMP      32
      DECREFX(I); FI := J                                                        COMP      32
     END (*PCKD*) ;                                                              COMP      32
    LCP := PC.CP; LNR := FI;                                                     COMP      32
    IF NOT (FI IN [6,7]) THEN                                                    COMP      32
     BEGIN NEEDX(6,7,I); BXIXJ(I,FI);                                            COMP      32
      FI := I                                                                    COMP      32
     END;                                                                        COMP      32
    LBX := LCP <> PC.CP;                                                         COMP      32
    WITH XRGS[FI] DO                                                             COMP      32
     BEGIN LXFICST := XCONT = SHRTCST;                                           COMP      32
      IF LXFICST THEN LCST := CSTVAL                                             COMP      32
      ELSE ROTATEX(FI);                                                          COMP      32
```

```
 END;                                                                    COMP     32
CASE WORDACC OF                                                          COMP     32
 DRCT:                                                                   COMP     32
  BEGIN                                                                  COMP     32
   FOR I := 0 TO 7 DO                                                    COMP     32
    IF I <> FI THEN                                                      COMP     32
     WITH XRGS[I] DO                                                     COMP     32
      IF XCONT = SIMPVAR THEN                                            COMP     32
       IF (XLEV = VLEVEL)AND (XADDR = CWDISPL) THEN                      COMP     32
        XCONT := AVAIL;                                                  COMP     32
   IF TYPTR↑.FORM = POINTER THEN                                         COMP     32
    BEGIN                                                                COMP     32
     FOR I := 0 TO 7 DO                                                  COMP     32
      WITH XRGS[I] DO                                                    COMP     32
       IF XCONT = INDVAR THEN                                            COMP     32
        IF XRGS[XREG].XCONT = AVAIL THEN XCONT := AVAIL;                 COMP     32
     FOR I := 1 TO 7 DO                                                  COMP     32
      WITH ARGS[I] DO                                                    COMP     32
       IF ACONT = INDADDR THEN                                           COMP     32
        IF XRGS[AREG].XCONT = AVAIL THEN                                 COMP     32
         ACONT := UNSPECADDR                                             COMP     32
    END;                                                                 COMP     32
   WITH LXRG DO                                                          COMP     32
    BEGIN XCONT := SIMPVAR; REFNR := 1; VPADDR := FALSE;                 COMP     32
     XLEV := VLEVEL; XADDR := CWDISPL; SHFTCNT := 0                      COMP     32
    END                                                                  COMP     32
  END;                                                                   COMP     32
 INDRCT:                                                                 COMP     32
  BEGIN                                                                  COMP     32
   FOR I := 0 TO 7 DO                                                    COMP     32
    IF I <> FI THEN                                                      COMP     32
     WITH XRGS[I] DO                                                     COMP     32
      IF XCONT = INDVAR THEN                                             COMP     32
       IF (XREG = VWDISPL)AND (XDISPL = CWDISPL) THEN                    COMP     32
        BEGIN DECREFX(VWDISPL); XCONT := AVAIL END;                      COMP     32
   IF XRGS[VWDISPL].XCONT = SIMPVAR THEN                                 COMP     32
    WITH LXRG DO                                                         COMP     32
     BEGIN XCONT := INDVAR; REFNR := 1;                                  COMP     32
      XREG := VWDISPL; XDISPL := CWDISPL; SHFTCNT := 0                   COMP     32
     END                                                                 COMP     32
   ELSE                                                                  COMP     32
    WITH LXRG DO                                                         COMP     32
     BEGIN XCONT := OTHER; REFNR := 1 END                                COMP     32
  END;                                                                   COMP     32
 INXD:                                                                   COMP     32
  WITH LXRG DO                                                           COMP     32
   BEGIN XCONT := OTHER; REFNR := 1 END                                  COMP     32
END (*CASE*);                                                            COMP     32
IF WORDACC <> DRCT THEN DECREFX(VWDISPL);                                COMP     32
IF LXRG.XCONT = OTHER THEN                                               COMP     32
 BEGIN                                                                   COMP     32
  IF NOT LXFICST AND LBX THEN                                            COMP     32
   WITH XRGS[FI] DO                                                      COMP     32
    BEGIN IF XCONT = INDVAR THEN DECREFX(XREG);                          COMP     32
     XCONT := OTHER                                                      COMP     32
    END                                                                  COMP     32
 END                                                                     COMP     32
ELSE                                                                     COMP     32
 BEGIN IF LBX THEN K := LNR ELSE K := FI;                                COMP     32
  IF (LXRG.XCONT <> INDVAR) OR (LXRG.XREG <> K) THEN                     COMP     32
   BEGIN                                                                 COMP     32
    WITH XRGS[K] DO                                                      COMP     32
     IF XCONT = INDVAR THEN DECREFX(XREG)                                COMP     33
     ELSE IF XCONT = SIMPVAR THEN                                        COMP     33
           FOR I := 0 TO 7 DO                                            COMP     33
            WITH XRGS[I] DO                                              COMP     33
             IF XCONT = INDVAR THEN                                      COMP     33
              IF XREG = K THEN                                           COMP     33
               IF REFNR = 0 THEN XCONT := AVAIL                          COMP     33
               ELSE XCONT := OTHER;                                      COMP     33
    IF K = LNR THEN                                                      COMP     33
     BEGIN LXRG.REFNR := XRGS[K].REFNR;                                  COMP     33
      IF LXRG.REFNR = 0 THEN LXRG.LASTREF := IC                          COMP     33
     END;                                                                COMP     33
    XRGS[K] := LXRG;                                                     COMP     33
    IF LXRG.XCONT = INDVAR THEN                                          COMP     33
     WITH XRGS[LXRG.XREG] DO REFNR := REFNR + 1;                         COMP     33
   END                                                                   COMP     33
 END;                                                                    COMP     33
CASE WORDACC OF                                                          COMP     33
 DRCT:                                                                   COMP     33
```

```
      BEGIN                                                                    COMP     331
       FOR I := 1 TO 7 DO                                                      COMP     332
        WITH ARGS[I] DO                                                        COMP     332
         IF ACONT = SIMPADDR THEN                                              COMP     332
          IF ALEV = VLEVEL THEN                          000042                 COMP     332
           BEGIN LADDR := CWDISPL - AADDR;                                     COMP     332
            IF ABS(LADDR) <= 1 THEN                                            COMP     332
             BEGIN                                                             COMP     332
              IF LADDR >= 0 THEN GEN15(548,FI,I,LADDR)                         COMP     332
              ELSE GEN15(558,FI,I,1);                                          COMP     332
              GOTO 1                                                           COMP     332
             END;                                                              COMP     333
           END;                                                                COMP     333
       IF VLEVEL IN LEVELS THEN GEN30(518,FI,BRG[VLEVEL],CWDISPL,0)            COMP     333
       ELSE                                                                    COMP     333
        BEGIN LOADBASE(VLEVEL,TRUE,I); GEN30(528,FI,I,CWDISPL,0);              COMP     333
         DECREFX(I)                                                            COMP     333
        END;                                                                   COMP     333
1:     WITH ARGS[FI] DO                                                        COMP     333
        BEGIN ACONT := SIMPADDR; ALEV := VLEVEL; AADDR := CWDISPL              COMP     333
        END                                                                    COMP     333
      END;                                                                     COMP     334
     INDRCT:                                                                   COMP     334
      BEGIN                                                                    COMP     334
       IF LXRG.XCONT = INDVAR THEN                                             COMP     334
        BEGIN                                                                  COMP     334
         FOR I := 1 TO 7 DO                                                    COMP     334
          WITH ARGS[I] DO                                                      COMP     334
           IF ACONT = INDADDR THEN                                             COMP     334
            IF (AREG = VWDISPL)AND (ADISPL = CWDISPL) THEN                     COMP     334
             BEGIN GEN15(548,FI,I,0); GOTO 2 END                               COMP     334
        END;                                                                   COMP     335
       IF CWDISPL IN [0,1] THEN GEN15(538,FI,VWDISPL,CWDISPL)                  COMP     335
       ELSE GEN30(528,FI,VWDISPL,CWDISPL,0);                                   COMP     335
2:     WITH ARGS[FI] DO                                                        COMP     335
        IF LXRG.XCONT = INDVAR THEN                                            COMP     335
         BEGIN ACONT := INDADDR; AREG := VWDISPL;                              COMP     335
          ADISPL := CWDISPL                                                    COMP     335
         END                                                                   COMP     335
        ELSE ACONT := UNSPECADDR                                               COMP     335
      END;                                                                     COMP     335
     INXD:                                                                     COMP     336
      BEGIN                                                                    COMP     336
       IF VLEVEL IN LEVELS THEN                                                COMP     336
         BEGIN NEEDB(I);                                                       COMP     336
          IF (PC.CP = 3) OR (I = BRG[VLEVEL]) THEN                             COMP     336
           BEGIN GEN15(638,I,VWDISPL,BRG[VLEVEL]);                             COMP     336
            GEN30(518,FI,I,CWDISPL,0)                                          COMP     336
           END                                                                 COMP     336
          ELSE                                                                 COMP     336
           BEGIN GEN30(628,I,VWDISPL,CWDISPL,0);                               COMP     336
            GEN15(568,FI,BRG[VLEVEL],I)                                        COMP     337
           END;                                                                COMP     337
          BRGS[I].BCONT := FREE                                                COMP     337
         END                                                                   COMP     337
       ELSE                                                                    COMP     337
        BEGIN NEEDB(J); GEN30(628,J,VWDISPL,CWDISPL,0);                        COMP     337
         LOADBASE(VLEVEL,TRUE,I); GEN15(538,FI,I,J);                           COMP     337
         BRGS[J].BCONT := FREE; DECREFX(I)                                     COMP     337
        END;                                                                   COMP     337
       ARGS[FI].ACONT := UNSPECADDR                                            COMP     337
      END                                                                      COMP     338
    END (*CASE*);                                                              COMP     338
    IF VLEVEL > 0 THEN                                                         COMP     338
    (*UPDATE OF REGISTER CONTENTS:*)                                           COMP     338
    IF WORDACC = DRCT THEN    (*SIMPLE VAR HAS GOT NEW VALUE. DISPOSE*)        COMP     338
     BEGIN                    (*X-REGS CONTAINING VAR PARAMS*)                 COMP     338
      FOR I := 0 TO 7 DO                                                       COMP     338
       IF I <> FI THEN                                                         COMP     338
        WITH XRGS[I] DO                                                        COMP     338
         IF XCONT = INDVAR THEN                                                COMP     338
          IF XRGS[XREG].VPADDR THEN                                            COMP     339
           IF XRGS[XREG].XLEV > VLEVEL THEN                                    COMP     339
            BEGIN DECREFX(XREG);                                               COMP     339
             IF REFNR > 0 THEN XCONT := OTHER ELSE XCONT := AVAIL              COMP     339
            END                                                                COMP     339
     END                                                                       COMP     339
    ELSE  (*ASSUME COINCIDANCE. DISPOSE X-REGS NOT CONT. SIMPLE VARS*)         COMP     339
     BEGIN                                                                     COMP     339
      FOR I := 0 TO 7 DO                                                       COMP     339
       IF I <> FI THEN                                                         COMP     339
```

```
        WITH XRGS[I] DO                                                    000043
         IF XCONT = INDVAR THEN
          BEGIN DECREFX(XREG);
           IF REFNR > 0 THEN XCONT := OTHER ELSE XCONT := AVAIL
          END;
       IF XRGS[VWDISPL].VPADDR THEN   (*DISPOSE X-REGS CONTAINING*)
        BEGIN                        (*SIMPLE VARS OF LEVEL < XLEV*)
         FOR I := 0 TO 7 DO
          IF I <> FI THEN
           WITH XRGS[I] DO
            IF XCONT = SIMPVAR THEN
             IF XLEV < XRGS[VWDISPL].XLEV THEN
              IF REFNR > 0 THEN XCONT := OTHER ELSE XCONT := AVAIL;
         FOR I := 1 TO 7 DO
          WITH ARGS[I] DO
           IF ACONT = INDADDR THEN
            IF XRGS[AREG].XCONT = AVAIL THEN
             ACONT := UNSPECADDR
        END
      END;
    END (*TYPTR <> NIL*);
  DECREFX(FI)
END (*STORE*) ;

PROCEDURE CHECKBNDS(FI: REGNR; FMIN,FMAX: INTEGER; FADDR: ADDRRANGE);
  (*TEST X-FI AGAINST BOUNDS FMIN AND FMAX.IF OUT OF BOUNDS JUMP
  TO FADDR*)
  VAR BOUND: ATTR; I,J,K: REGNR;
BEGIN GEN30(51B,0,0,IC,0);
  WITH BOUND DO
   BEGIN TYPTR := INTPTR; KIND := CST; CVAL.IVAL := FMIN END;
  IF FMIN <> 0 THEN
   BEGIN LOAD(BOUND,I); DECREFX(I); NEEDX(0,7,K);
    GEN15(37B,K,FI,I)
   END;
  WITH BOUND DO
   BEGIN TYPTR := INTPTR; KIND := CST; CVAL.IVAL := FMAX END;
  LOAD(BOUND,I);
  DECREFX(I); NEEDX(0,7,J); GEN15(37B,J,I,FI);
  IF FMIN <> 0 THEN
   BEGIN GEN15(12B,J,J,K); DECREFX(K) END
  ELSE GEN15(12B,J,J,FI);
  GEN30(03B,3,J,FADDR,0); DECREFX(J)
END (*CHECKBNDS*) ;

  PROCEDURE STATEMENT(FSYS: SETOFSYS);
   LABEL 1;
   VAR LCP: CTP; LLP: LBP; LOCP: LOCOFREF;
       LASTSY: SYMBOL;

   PROCEDURE PACKOFL(FI: REGNR);
    VAR K: REGNR;
   BEGIN GEN30(51B,0,0,IC,0); NEEDX(0,7,K); GEN15(10B,K,FI,0);
    GEN15(21B,K,0,48); GEN30(03B,1,K,OVLERR,0); DECREFX(K)
   END (*PACKOFL*) ;

   PROCEDURE PACKANDNORM(VAR FI: REGNR);
    VAR K: REGNR;
   BEGIN IF DEBUG THEN PACKOFL(FI); DECREFX(FI); NEEDX(0,7,K);
    GEN15(27B,K,0,FI); GEN15(24B,K,0,K);
    FI := K
   END (*PACKANDNORM*) ;

   PROCEDURE OPERATION(FOP: OPRANGE; VAR FK: REGNR; FI,FJ: REGNR);
   BEGIN DECREFX(FI); DECREFX(FJ); NEEDX(0,7,FK); GEN15(FOP,FK,FI,FJ)
   END (*OPERATION*) ;

   PROCEDURE EXPREP(FVAL: INTEGER;VAR FREC: CSTREC);
    (*RETURN EXPONENTIAL REPRESENTATION OF FVAL:
      CKIND = PUREP    IF    FVAL = 2**EXP,
      CKIND = POSP     IF    FVAL = 2**EXP1*(2**EXP2 + 1),
      CKIND = NEGP     IF    FVAL = 2**EXP1*(2**EXP2 - 1),
      CKIND = NOP      ELSE.*)
    VAR E1,E2: BITRANGE;
   BEGIN
    IF FVAL > 0 THEN
     BEGIN E1 := 0;
      WHILE NOT ODD(FVAL) DO
       BEGIN FVAL := FVAL DIV 2; E1 := E1 + 1 END;
      IF FVAL = 1 THEN
       WITH FREC DO
```

```
COMP  340
COMP  340
COMP  340
COMP  340
COMP  340
COMP  340
COMP  340
COMP  340
COMP  340
COMP  340
COMP  340
COMP  341
COMP  341
COMP  341
COMP  341
COMP  341
COMP  341
COMP  341
COMP  341
COMP  341
COMP  341
COMP  342
COMP  342
COMP  342
COMP  342
COMP  342
COMP  342
COMP  342
COMP  342
COMP  343
COMP  343
COMP  343
COMP  343
COMP  343
COMP  343
COMP  343
COMP  343
COMP  344
COMP  344
COMP  344
COMP  344
COMP  344
COMP  344
COMP  344
COMP  344
COMP  345
COMP  345
COMP  345
COMP  345
COMP  345
COMP  345
COMP  345
COMP  345
COMP  346
COMP  346
COMP  346
COMP  346
COMP  346
COMP  346
COMP  346
COMP  346
COMP  346
COMP  346
COMP  347
COMP  347
COMP  347
COMP  347
COMP  347
COMP  347
COMP  347
COMP  348
```

```
      BEGIN CKIND := PUREP; EXP := E1 END
    ELSE
      BEGIN FVAL := FVAL DIV 2; E2 := 1;
       IF ODD(FVAL) THEN
        BEGIN
         REPEAT FVAL := FVAL DIV 2; E2 := E2 + 1
         UNTIL NOT ODD(FVAL);
         IF FVAL > 0 THEN FREC.CKIND := NOP
         ELSE
          WITH FREC DO
           BEGIN CKIND := NEGP; EXP1 := E1; EXP2 := E2 END
        END
       ELSE
        BEGIN
         REPEAT FVAL := FVAL DIV 2; E2 := E2 + 1
         UNTIL ODD(FVAL);
         IF FVAL > 1 THEN FREC.CKIND := NOP
         ELSE
          WITH FREC DO
           BEGIN CKIND := POSP; EXP1 := E1; EXP2 := E2 END
        END
      END
    END
   ELSE FREC.CKIND := NOP
  END (*EXPREP*) ;

  PROCEDURE OPTMULT(FI: REGNR; FREC: CSTREC; FEQ: BOOLEAN;
  VAR FK: REGNR);
   (*GENERATE CODE FOR X-FK := X-FI*FREC.  FEQ <=> FI=FK IS ALLOWED*)
   VAR E: BITRANGE; I,K: REGNR; B: BOOLEAN;
  BEGIN B := FALSE;
   WITH FREC DO
    BEGIN IF CKIND = PUREP THEN E := EXP ELSE E := EXP1;
     IF E <> 0 THEN
      IF E = 1 THEN
       BEGIN NEEDX(0,7,K); GEN15(36B,K,FI,FI);
        IF FEQ OR (CKIND = PUREP) THEN DECREFX(FI)
        ELSE
         BEGIN B := TRUE; I := FI END;
        FI := K
       END
      ELSE
       IF FEQ AND (XRGS[FI].REFNR = 1) THEN (*DESTROY X-FI*)
        BEGIN GEN15(20B,FI,0,E);
         WITH XRGS[FI] DO
          BEGIN IF XCONT = INDVAR THEN DECREFX(XREG);
           XCONT := OTHER
          END
        END
       ELSE (*COPY X-FI*)
        BEGIN NEEDX(0,7,K); GEN15(10B,K,FI,0);
         GEN15(20B,K,0,E);
         IF FEQ OR (CKIND = PUREP) THEN DECREFX(FI)
         ELSE
          BEGIN B := TRUE; I := FI END;
         FI := K
        END;
     IF CKIND <> PUREP THEN
      BEGIN NEEDX(0,7,K);
       IF B THEN DECREFX(I);
       GEN15(10B,K,FI,0); GEN15(20B,K,0,EXP2);
       IF CKIND = POSP THEN GEN15(36B,K,K,FI)
       ELSE GEN15(37B,K,K,FI);
       DECREFX(FI); FK := K
      END
     ELSE FK := FI
    END
  END (*OPTMULT*) ;

  PROCEDURE EXPRESSION(FSYS: SETOFSYS); FORWARD;

 PROCEDURE ASSIGNTO(VAR LATTR: ATTR);
   VAR I,J,K,L,M: REGNR; FLOAT: BOOLEAN;
       LWORDS: ADDRRANGE; LMIN,LMAX: INTEGER;
 BEGIN
     IF (LATTR.TYPTR <> NIL)AND (GATTR.TYPTR <> NIL) THEN
      BEGIN
       IF COMPTYPES(REALPTR,LATTR.TYPTR,FALSE)
        AND COMPTYPES(GATTR.TYPTR,INTPTR,FALSE) THEN
        FLOAT:=TRUE
       ELSE FLOAT := FALSE;
```

```
      IF COMPTYPES(LATTR.TYPTR,GATTR.TYPTR,FALSE) OR FLOAT THEN          COMP     356
       CASE LATTR.TYPTR↑.FORM OF                                         COMP     356
        SCALAR,                                                          COMP     356
        SUBRANGE:                                                        COMP     356
         BEGIN                                                           COMP     356
          IF (LATTR.TYPTR=INTPTR)OR (COMPTYPES(LATTR.TYPTR,              COMP     356
           REALPTR,FALSE)) THEN LOAD(GATTR,I)                            COMP     356
          ELSE                                                           COMP     356
           BEGIN GETBOUNDS(LATTR.TYPTR,LMIN,LMAX);                       COMP     357
            IF GATTR.KIND = CST THEN                                     COMP     357
             BEGIN                                                       COMP     357
              IF (GATTR.CVAL.IVAL<LMIN)OR (GATTR.CVAL.IVAL               COMP     357
               >LMAX) THEN ERROR(303);                                   COMP     357
              LOAD(GATTR,I)                                              COMP     357
             END                                                         COMP     357
            ELSE                                                         COMP     357
             BEGIN LOAD(GATTR,I);                                        COMP     357
              IF DEBUG THEN CHECKBNDS(I,LMIN,LMAX,ASSERR)                COMP     357
             END                                                         COMP     358
           END;                                                          COMP     358
          IF FLOAT THEN PACKANDNORM(I);                                  COMP     358
          STORE(LATTR,I)                                                 COMP     358
         END;                                                            COMP     358
        POINTER,                                                         COMP     358
        POWER:                                                           COMP     358
         BEGIN LOAD(GATTR,I);                                            COMP     358
          STORE(LATTR,I)                                                 COMP     358
         END;                                                            COMP     358
        ARRAYS,                                                          COMP     359
        RECORDS:                                                         COMP     359
         IF LATTR.TYPTR↑.FTYPE THEN ERROR(146) ELSE                      COMP     359
         BEGIN LWORDS := FULLWORDS(LATTR.TYPTR↑.SIZE);                   COMP     359
          IF LWORDS = 1 THEN                                             COMP     359
           BEGIN LOAD(GATTR,I); STORE(LATTR,I) END                       COMP     359
          ELSE                                                           COMP     359
           BEGIN LOADADDRESS(GATTR,I); LOADADDRESS(LATTR,K);             COMP     359
            IF [I,K] = [6,7] THEN  (* GET X-K FREE *)                     COMP     359
             BEGIN NEEDX(0,5,L); BXIXJ(L,K); K:=L END;                   COMP     359
            NEEDX(1,5,L); NEEDX(6,7,M);                                  COMP     360
            ARGS[L].ACONT := UNSPECADDR;                                 COMP     360
            ARGS[M].ACONT := UNSPECADDR;                                 COMP     360
            IF LWORDS > 3 THEN                                           COMP     360
             BEGIN NEEDB(J); GEN30(61B,J,0,LWORDS-1,0); NOOP;            COMP     360
              GEN15(53B,L,I,J); GEN15(10B,M,L,0);                        COMP     360
              GEN15(53B,M,K,J); GEN15(67B,J,J,1);                        COMP     360
              GEN30(06B,J,0,IC-1,2); BRGS[J].BCONT := FREE               COMP     360
             END                                                         COMP     360
            ELSE (*EXPAND LOOP*)                                         COMP     360
             BEGIN GEN15(53B,L,I,0); GEN15(10B,M,L,0);                   COMP     361
              GEN15(53B,M,K,0);                                          COMP     361
              FOR J := 2 TO LWORDS DO                                    COMP     361
               BEGIN GEN15(54B,L,L,1); GEN15(10B,M,L,0);                 COMP     361
                GEN15(54B,M,M,1)                                         COMP     361
               END                                                       COMP     361
             END;                                                        COMP     361
            DECREFX(I); DECREFX(K); DECREFX(L); DECREFX(M);               COMP     361
            CLEARREGS; (*BE MORE SOPHISTICATED LATER*)                   COMP     361
           END                                                           COMP     361
         END;                                                            COMP     362
        FILES: ERROR(146)                                                COMP     362
       END                                                               COMP     362
      ELSE ERROR(129)                                                    COMP     362
     END                                                                 COMP     362
END (*ASSIGNTO*);                                                        COMP     362
                                                                         COMP     362
 PROCEDURE SELECTOR(FSYS: SETOFSYS; FCP: CTP);                           COMP     362
  VAR LATTR:ATTR; LCP:CTP; I,K : REGNR;                                  COMP     362
                                                                         COMP     362
  PROCEDURE IDADDRESS;                                                   COMP     363
   VAR I: REGNR;                                                         COMP     363
  BEGIN                                                                  COMP     363
   WITH FCP↑, LATTR DO                                                   COMP     363
    BEGIN TYPTR := IDTYPE; KIND := VARBL;                                COMP     363
     IF TYPTR <> NIL THEN                                                COMP     363
      CASE KLASS OF                                                      COMP     363
       VARS:                                                             COMP     363
        BEGIN                                                            COMP     363
         WORDACC := VKIND;                                               COMP     363
         VLEVEL := VLEV; CWDISPL := VADDR;                               COMP     364
         PCKD := FALSE;                                                  COMP     364
          IF WORDACC = INDRCT THEN                                       COMP     364
```

000045

```
          BEGIN WORDACC := DRCT; LOAD(LATTR,I);                                COMP     364
           KIND := VARBL; XRGS[I].VPADDR := TRUE;                              COMP     364
           WORDACC := INDRCT; CWDISPL := 0;                                    COMP     364
           VWDISPL := I; PCKD := FALSE                                         COMP     364
          END                                                                  COMP     364
       END;                                                                    COMP     364
      FIELD:                                                                   COMP     364
       WITH DISPLAY[DISX] DO                                                   COMP     365
        BEGIN WORDACC := DRCT; VLEVEL := LEV;                                  COMP     365
         IF WACC = DRCT THEN CWDISPL := CWDSPL + FLDADDR                       COMP     365
         ELSE                                                                  COMP     365
          BEGIN CWDISPL := CWDSPL; PCKD := FALSE;                              COMP     365
           LOAD(LATTR,I);                                                      COMP     365
           KIND := VARBL; WORDACC := INDRCT;                                   COMP     365
           CWDISPL := FLDADDR; VWDISPL := I                                    COMP     365
          END;                                                                 COMP     365
         IF PKD THEN (*IMPLIES (FLDADDR=0)AND PCKDFLD*)                        COMP     365
          BEGIN PCKD := TRUE;                                                  COMP     366
           IF BACC = DRCT THEN                                                 COMP     366
            BEGIN CBDISPL := BDSPL + BITADDR;                                  COMP     366
             BITREG := NONE                                                    COMP     366
            END                                                                COMP     366
           ELSE                                                                COMP     366
            BEGIN                                                              COMP     366
             WITH GATTR DO                                                     COMP     366
              BEGIN TYPTR := IDTYPE; KIND := VARBL;                            COMP     366
               WORDACC := DRCT; VLEVEL := LEVEL;                               COMP     366
               CWDISPL := BDSPL; PCKD := FALSE                                 COMP     367
              END;                                                             COMP     367
             LOAD(GATTR,I);                                                    COMP     367
             CBDISPL := BITADDR; BITREG := XREG;                               COMP     367
             VBDISPL := I                                                      COMP     367
            END                                                                COMP     367
          END (*PKD*)                                                          COMP     367
         ELSE                                                                  COMP     367
          IF PCKDFLD THEN                                                      COMP     367
           BEGIN PCKD := TRUE; CBDISPL := BITADDR;                             COMP     367
            BITREG := NONE                                                     COMP     368
           END                                                                 COMP     368
          ELSE PCKD := FALSE                                                   COMP     368
        END (*WITH*) ;                                                         COMP     368
      FUNC:                                                                    COMP     368
       IF PFDECKIND = STANDARD THEN                                            COMP     368
        BEGIN ERROR(150); TYPTR := NIL END                                     COMP     368
       ELSE                                                                    COMP     368
        BEGIN                                                                  COMP     368
        IF PFDECL IN [EXTDECL,FTNDECL] THEN ERROR(150)                         COMP     368
        ELSE                                                                   COMP     369
         IF PFKIND = FORMAL THEN ERROR(151)                                    COMP     369
         ELSE                                                                  COMP     369
          IF (PFLEV+1 <> LEVEL) OR (FPROCP <> FCP) THEN ERROR(177);            COMP     369
           WORDACC := DRCT; VLEVEL := PFLEV + 1;                               COMP     369
           CWDISPL := 2; (*IMPLICIT ADDRESS OF FCT RESULT*)                    COMP     369
           PCKD := FALSE                                                       COMP     369
          END                                                                  COMP     369
    END (*CASE*)                                                               COMP     369
  END (*WITH*)                                                                 COMP     369
END (*IDADDRESS*) ;                                                            COMP     370
                                                                               COMP     370
PROCEDURE INDEXCODE;                                                           COMP     370
 VAR LBREG: REGKIND; LBITS: BITRANGE; LWORDS: ADDRRANGE;                       COMP     370
   LOW,HIGH: INTEGER; LB: BOOLEAN; I,J,K,L: REGNR;                             COMP     370
   LACC: ACCESSKIND; LT: INTEGER; LREC: CSTREC;                                COMP     370
   LBOUND: ATTR;                                                               COMP     370
                                                                               COMP     370
 PROCEDURE POWEROF2(I: INTEGER; VAR FB: BOOLEAN;                               COMP     370
           VAR FEXP: INTEGER);                                                 COMP     370
  (*DECIDE WHETHER POSITIVE I IS A POWER OF TWO*)                               COMP     371
  VAR LEXP: INTEGER;                                                           COMP     371
 BEGIN LEXP := 0;                                                              COMP     371
  WHILE NOT ODD(I) DO                                                          COMP     371
   BEGIN I := I DIV 2; LEXP := LEXP + 1 END;                                   COMP     371
  FEXP := LEXP; FB := I = 1                                                    COMP     371
 END (*POWEROF2*) ;                                                            COMP     371
                                                                               COMP     371
BEGIN (*INDEXCODE*) LACC := DRCT; LBREG := NONE;                               COMP     371
 IF GATTR.KIND <> CST THEN LOAD(GATTR,I);                                      COMP     371
 WITH LATTR, TYPTR^ DO                                                         COMP     372
  BEGIN                                                                        COMP     372
    GETBOUNDS(INXTYPE,LOW,HIGH);                                               COMP     372
    IF GATTR.KIND = CST THEN                                                   COMP     372
```

```
    BEGIN IF (GATTR.CVAL.IVAL>HIGH)OR (GATTR.CVAL.IVAL<LOW)                COMP  372
     THEN ERROR(302)                                                       COMP  372
    END                                                                    COMP  372
   ELSE                                                                    COMP  372
    IF DEBUG THEN CHECKBNDS(I,LOW,HIGH,INXERR);                  000047    COMP  372
IF PCKDARR AND PARTWORDELS THEN (*PARTWORD ACCESS*)                        COMP  372
 BEGIN                                                                     COMP  373
  IF NOT PCKD THEN                                                         COMP  373
   BEGIN PCKD := TRUE; CBDISPL := 0; BITREG := NONE END;                   COMP  373
  LBITS := AELTYPE↑.SIZE.BITS;                                             COMP  373
  IF FULLWORDS(SIZE) = 1 THEN                                              COMP  373
   IF GATTR.KIND = CST THEN                                                COMP  373
    CBDISPL := CBDISPL + (GATTR.CVAL.IVAL - LOW)*LBITS                     COMP  373
   ELSE                                                                    COMP  373
    BEGIN CBDISPL := CBDISPL - LOW*LBITS;                                  COMP  373
     EXPREP(LBITS,LREC);                                                   COMP  373
     IF LREC.CKIND <> NOP THEN OPTMULT(I,LREC,TRUE,J)                      COMP  374
     ELSE                                                                  COMP  374
      BEGIN NEEDX(0,7,K); GEN30(71B,K,0,LBITS,0);                          COMP  374
       OPERATION(42B,J,I,K)                                                COMP  374
      END;                                                                 COMP  374
     LBREG := XREG                                                         COMP  374
    END                                                                    COMP  374
   ELSE                                                                    COMP  374
    IF GATTR.KIND = CST THEN                                               COMP  374
     BEGIN CWDISPL := CWDISPL + (GATTR.CVAL.IVAL - LOW)                    COMP  374
           DIV ELSPERWORD;                                                 COMP  375
      CBDISPL := CBDISPL + (GATTR.CVAL.IVAL - LOW)                         COMP  375
           MOD ELSPERWORD * LBITS                                          COMP  375
     END                                                                   COMP  375
    ELSE                                                                   COMP  375
     BEGIN                                                                 COMP  375
       IF LOW = 0 THEN J := I                                              COMP  375
       ELSE                                                                COMP  375
        BEGIN                                                              COMP  375
         WITH LBOUND DO                                                    COMP  375
          BEGIN TYPTR := INTPTR; KIND := CST;                              COMP  376
          CVAL.IVAL := LOW                                                 COMP  376
         END;                                                              COMP  376
         LOAD(LBOUND,J); OPERATION(37B,J,I,J)                              COMP  376
        END;                                                               COMP  376
      POWEROF2(ELSPERWORD,LB,LT);                                          COMP  376
      NEEDX(0,7,K);                                                        COMP  376
      IF LB THEN                                                           COMP  376
       BEGIN GEN15(10B,K,J,0); GEN15(21B,K,0,LT);                          COMP  376
       END                                                                 COMP  376
      ELSE                                                                 COMP  377
       BEGIN                                                               COMP  377
        IF ELSPERWORD IN [3,6,12] THEN LT := 87382                         COMP  377
        ELSE                                                               COMP  377
         IF ELSPERWORD IN [5,10,20] THEN LT := 52429                       COMP  377
         ELSE                                                              COMP  377
          IF ELSPERWORD = 7 THEN LT := 74899                               COMP  377
          ELSE LT := 69906;                                                COMP  377
        GEN30(71B,K,0,LT,0); GEN15(42B,K,J,K);                             COMP  377
        IF ELSPERWORD <= 5 THEN LT := 18                                   COMP  377
        ELSE                                                               COMP  378
         IF ELSPERWORD <= 10 THEN LT := 19                                 COMP  378
         ELSE                                                              COMP  378
          IF ELSPERWORD <= 20 THEN LT := 20                                COMP  378
          ELSE                                                             COMP  378
           IF ELSPERWORD = 30 THEN LT := 21                                COMP  378
           ELSE LT := 22;                                                  COMP  378
        GEN15(21B,K,0,LT)                                                  COMP  378
       END;                                                                COMP  378
      IF LB THEN                                                           COMP  378
       BEGIN NEEDX(0,7,L); GEN15(10B,L,K,0);                               COMP  379
        GEN15(20B,L,0,LT)                                                  COMP  379
       END                                                                 COMP  379
      ELSE                                                                 COMP  379
       BEGIN EXPREP(ELSPERWORD,LREC);                                      COMP  379
        IF LREC.CKIND <> NOP THEN                                          COMP  379
         BEGIN OPTMULT(K,LREC,FALSE,L);                                    COMP  379
          (*RESET REFERENCE:*)                                             COMP  379
          WITH XRGS[K] DO                                                  COMP  379
           BEGIN XCONT := OTHER; REFNR := 1 END;                           COMP  379
         END                                                               COMP  380
        ELSE                                                               COMP  380
         BEGIN NEEDX(0,7,L); GEN30(71B,L,0,ELSPERWORD,0);                  COMP  380
          GEN15(42B,L,K,L)                                                 COMP  380
         END                                                               COMP  380
```

```
              END;                                                                  COMP     380
             DECREFX(J); I := J; NEEDX(0,7,J); GEN15(37B,J,I,L);                    COMP     380
             EXPREP(LBITS,LREC);                                                    COMP     380
             IF LREC.CKIND <> NOP THEN OPTMULT(J,LREC,TRUE,J)                       COMP     380
             ELSE                                                                   COMP     380
              BEGIN GEN30(71B,L,0,LBITS,0); GEN15(42B,J,J,L) END;                   COMP     381
             DECREFX(L);                                                            COMP     381
             LACC := INXD; LBREG := XREG                                            COMP     381
            END                                                                     COMP     381
         END (*PCKDARR AND PARTWORDELS*)                                  000048    COMP     381
        ELSE                                                                        COMP     381
         BEGIN                                                                      COMP     381
           LWORDS := FULLWORDS(AELTYPE↑.SIZE);                                      COMP     381
           IF GATTR.KIND = CST THEN                                                 COMP     381
            CWDISPL := CWDISPL + (GATTR.CVAL.IVAL - LOW)*LWORDS                     COMP     381
           ELSE                                                                     COMP     382
            BEGIN CWDISPL := CWDISPL - LOW*LWORDS;                                  COMP     382
             IF ABS(CWDISPL) > MAXADDR THEN ERROR(181);                             COMP     382
             EXPREP(LWORDS,LREC);                                                   COMP     382
             IF LREC.CKIND <> NOP THEN OPTMULT(I,LREC,TRUE,K)                       COMP     382
             ELSE                                                                   COMP     382
              BEGIN NEEDX(0,7,J); GEN30(71B,J,0,LWORDS,0);                          COMP     382
               OPERATION(42B,K,I,J)                                                 COMP     382
              END;                                                                  COMP     382
             LACC := INXD                                                           COMP     382
            END;                                                                    COMP     383
         END;                                                                       COMP     383
        IF LACC <> DRCT THEN                                                        COMP     383
         IF WORDACC = DRCT THEN                                                     COMP     383
          BEGIN VWDISPL := K; WORDACC := INXD;                                      COMP     383
          END                                                                       COMP     383
         ELSE                                                                       COMP     383
          BEGIN OPERATION(36B,L,VWDISPL,K); VWDISPL := L END;                       COMP     383
        IF LBREG <> NONE THEN                                                       COMP     383
         IF BITREG = NONE THEN                                                      COMP     383
          BEGIN BITREG := XREG; VBDISPL := J END                                    COMP     384
         ELSE                                                                       COMP     384
          BEGIN OPERATION(36B,L,VBDISPL,J); VBDISPL := L END                        COMP     384
      END (*WITH LATTR*)                                                            COMP     384
    END (*INDEXCODE*) ;                                                             COMP     384
                                                                                    COMP     384
   BEGIN (*SELECTOR*)                                                               COMP     384
    IDADDRESS;                                                                      COMP     384
    TEST2(FSYS+SELECTSYS,59,[]);                                                    COMP     384
    WHILE SY IN SELECTSYS DO                                                        COMP     384
     BEGIN                                                                          COMP     385
(*[*)  IF SY = LBRACK THEN                                                          COMP     385
       BEGIN                                                                        COMP     385
        REPEAT                                                                      COMP     385
         WITH LATTR DO                                                              COMP     385
          IF TYPTR <> NIL THEN                                                      COMP     385
           IF TYPTR↑.FORM <> ARRAYS THEN                                            COMP     385
            BEGIN ERROR(138); TYPTR := NIL END;                                     COMP     385
         INSYMBOL; EXPRESSION(FSYS+[COMMA,RBRACK]);                                 COMP     385
         IF GATTR.TYPTR <> NIL THEN                                                 COMP     385
          IF GATTR.TYPTR↑.FORM > SUBRANGE THEN ERROR(113);                          COMP     386
         IF LATTR.TYPTR <> NIL THEN                                                 COMP     386
          WITH LATTR.TYPTR↑ DO                                                      COMP     386
           BEGIN                                                                    COMP     386
            IF COMPTYPES(INXTYPE,GATTR.TYPTR,FALSE) THEN                            COMP     386
             BEGIN                                                                  COMP     386
              IF (INXTYPE <> NIL)AND (AELTYPE <> NIL) THEN INDEXCODE                COMP     386
             END                                                                    COMP     386
            ELSE ERROR(139);                                                        COMP     386
            LATTR.TYPTR := AELTYPE                                                  COMP     386
           END                                                                      COMP     387
        UNTIL SY <> COMMA;                                                          COMP     387
        TEST1(RBRACK,12)                                                            COMP     387
       END (*IF SY = LBRACK*)                                                       COMP     387
      ELSE                                                                          COMP     387
(*.*)  IF SY = PERIOD THEN                                                          COMP     387
       BEGIN                                                                        COMP     387
        WITH LATTR DO                                                               COMP     387
         BEGIN                                                                      COMP     387
          IF TYPTR <> NIL THEN                                                      COMP     387
           IF TYPTR↑.FORM <> RECORDS THEN                                           COMP     388
            BEGIN ERROR(140); TYPTR := NIL END;                                     COMP     388
          INSYMBOL;                                                                 COMP     388
          IF SY = IDENT THEN                                                        COMP     388
           BEGIN                                                                    COMP     388
            IF TYPTR <> NIL THEN                                                    COMP     388
```

```
              BEGIN SEARCHSECTION(TYPTR↑.FIELDS,LCP);                                  COMP     388
               IF LCP = NIL THEN                                                       COMP     388
                BEGIN ERROR(152); TYPTR := NIL END                    000049            COMP     388
               ELSE                                                                    COMP     388
                WITH LCP↑ DO                                                           COMP     389
                 BEGIN TYPTR := IDTYPE;                                                COMP     389
                  IF PCKD THEN (*IMPLIES (FLDADDR=0)AND PCKDFLD*)                      COMP     389
                   CBDISPL := CBDISPL + BITADDR                                        COMP     389
                  ELSE                                                                 COMP     389
                   BEGIN CWDISPL := CWDISPL + FLDADDR;                                 COMP     389
                    IF PCKDFLD THEN                                                    COMP     389
                     BEGIN PCKD := TRUE; BITREG := NONE;                               COMP     389
                      CBDISPL := BITADDR                                               COMP     389
                     END                                                               COMP     389
                   END                                                                 COMP     390
                 END                                                                   COMP     390
              END;                                                                     COMP     390
              INSYMBOL                                                                 COMP     390
             END (*SY = IDENT*)                                                        COMP     390
            ELSE ERROR(2)                                                            COMP     390
           END (*WITH GATTR*)                                                        COMP     390
         END (*IF SY = PERIOD*)                                                      COMP     390
        ELSE                                                                         COMP     390
(***)    BEGIN                                                                       COMP     390
          IF LATTR.TYPTR <> NIL THEN                                                 COMP     391
           BEGIN                                                                     COMP     391
            WITH LATTR DO                                                            COMP     391
             BEGIN                                                                   COMP     391
              IF TYPTR↑.FORM = FILES THEN                                            COMP     391
               IF TYPTR↑.TEXTFILE THEN                                               COMP     391
                CWDISPL := CWDISPL + CHEFET - 1                                      COMP     391
               ELSE CWDISPL := CWDISPL + BINEFET - 1;                                COMP     391
              LOAD(LATTR,I);                                                         COMP     391
              WITH TYPTR↑ DO                                                         COMP     391
               IF FORM = POINTER THEN                                                COMP     392
                BEGIN TYPTR:=ELTYPE;                                                 COMP     392
                 IF DEBUG THEN                                                       COMP     392
                  BEGIN GEN30(51B,0,0,IC,0); NEEDB(K);                               COMP     392
                        GEN15(63B,K,I,0); GEN30(07B,K,4,PNTERR,0);                   COMP     392
                        BRGS[K].BCONT:=FREE                                          COMP     392
                  END                                                                COMP     392
                END                                                                  COMP     392
               ELSE                                                                  COMP     392
                IF FORM = FILES THEN TYPTR := FILTYPE                                COMP     392
                ELSE ERROR(141);                                                     COMP     392
              KIND := VARBL; WORDACC := INDRCT;                                      COMP     392
              CWDISPL := 0; VWDISPL := I;                                            COMP     392
              PCKD := FALSE                                                          COMP     392
             END                                                                     COMP     392
           END;                                                                      COMP     392
          INSYMBOL                                                                   COMP     393
         END;                                                                        COMP     393
       TEST2(FSYS+SELECTSYS,6,[])                                                    COMP     393
      END (*WHILE*) ;                                                              COMP     393
     GATTR := LATTR;                                                               COMP     394
    END (*SELECTOR*) ;                                                             COMP     394
                                                                                   COMP     394
    PROCEDURE CALL(FSYS: SETOFSYS; FCP: CTP);                                      COMP     394
     VAR LKEY: 1..25; I,J,K: REGNR;                                                COMP     394
                                                                                   COMP     394
     PROCEDURE VARIABLE(FSYS: SETOFSYS);                                           COMP     394
      VAR LCP: CTP;                                                                COMP     394
     BEGIN                                                                         COMP     394
      IF SY = IDENT THEN                                                           COMP     394
       BEGIN SEARCHID([VARS,FIELD],LCP); INSYMBOL END                              COMP     395
      ELSE BEGIN ERROR(2); LCP := UVARPTR END;                                     COMP     395
      SELECTOR(FSYS,LCP)                                                           COMP     395
     END (*VARIABLE*) ;                                                            COMP     395
                                                                                   COMP     395
     PROCEDURE STDFLPROCS;                                                         COMP     395
      VAR CHARFILE,SEGMFILE: BOOLEAN; NAME: ALFA; I: REGNR;                        COMP     395
          LDPLMT: ADDRRANGE;                                                       COMP     395
     BEGIN TEST1(LPARENT,9);                                                       COMP     395
      CLEARREGS; XRGS[1].XCONT := OTHER; (*RESERVE A1/X1*)                         COMP     395
      VARIABLE(FSYS+[COMMA,RPARENT]);                                              COMP     396
      CLEARREGS; (*TO PREVENT BX1... AND GUARANTEE SA1...*)                        COMP     396
      CHARFILE := FALSE; SEGMFILE := FALSE;  LDPLMT := BINEFET;                    COMP     396
      IF GATTR.TYPTR <> NIL THEN                                                   COMP     396
       BEGIN                                                                       COMP     396
        IF GATTR.WORDACC <> DRCT THEN                                              COMP     396
         XRGS[GATTR.VWDISPL].XCONT := OTHER;                                       COMP     396
```

```
    WITH GATTR.TYPTR↑ DO                                                    COMP    396
     IF FORM = FILES THEN                                                   COMP    396
      BEGIN CHARFILE := TEXTFILE; SEGMFILE := SEGFILE;                      COMP    396
       IF CHARFILE THEN LOPLMT := CHEFET                                    COMP    397
      END                                                                   COMP    397
     ELSE ERROR(116)                                                        COMP    397
   END                                                          000050      COMP    397
  ELSE GATTR.CWDISPL := 0;                                                  COMP    397
  CASE LKEY OF                                                              COMP    397
   (*GET*)      1: IF CHARFILE THEN                                         COMP    397
                    BEGIN NAME := ΞP.GETC    Ξ; LOPLMT := CHEFET-1        COMP    397
                    END                                                     COMP    397
                   ELSE NAME := ΞP.GETB    Ξ;                           COMP    397
   (*PUT*)      2: IF CHARFILE THEN                                         COMP    398
                    BEGIN NAME := ΞP.PUTC    Ξ; LOPLMT := CHEFET - 1      COMP    398
                    END                                                     COMP    398
                   ELSE NAME := ΞP.PUTB    Ξ;                           COMP    398
   (*RESET*)    3: NAME := ΞP.RESET   Ξ;                                   COMP    398
   (*REWRITE*)  4: NAME := ΞP.REWRT   Ξ;                                   COMP    398
   (*GETSEG*)   5,                                                         COMP    398
   (*PUTSEG*)   6: BEGIN IF NOT SEGMFILE THEN ERROR(116);                  COMP    398
                    IF LKEY = 5 THEN NAME := ΞP.GETS    Ξ                  COMP    398
                    ELSE NAME := ΞP.PUTS    Ξ                           COMP    398
                   END                                                     COMP    399
  END (*CASE*) ;                                                           COMP    399
  GATTR.CWDISPL := GATTR.CWDISPL + LOPLMT;                                 COMP    399
  LOAD(GATTR,I);                                                           COMP    399
  IF SY = COMMA THEN                                                       COMP    399
   BEGIN                                                                   COMP    399
    IF LKEY = 4 (*REWRITE*) THEN                                           COMP    399
     BEGIN IF NOT SEGMFILE THEN ERROR(126);                                COMP    399
      NAME := ΞP.RWRTS   Ξ                                                 COMP    399
     END                                                                   COMP    399
    ELSE                                                                   COMP    400
     IF LKEY <> 5 (*GETSEG*) THEN ERROR(126);                              COMP    400
    INSYMBOL;                                                              COMP    400
    EXPRESSION(FSYS+[RPARENT]);                                            COMP    400
    IF NOT COMPTYPES(GATTR.TYPTR,INTPTR,FALSE) THEN ERROR(127);            COMP    400
    LOAD(GATTR,I); BXIXJ(2,I)                                              COMP    400
   END                                                                     COMP    400
  ELSE                                                                     COMP    400
   IF LKEY = 5 THEN GEN15(76B,2,1,0);                                      COMP    400
  BJTOEXT(NAME);                                                           COMP    400
  TEST1(RPARENT,4)                                                         COMP    401
END (*STOFLPROCS*) ;                                                       COMP    401

PROCEDURE LINELIMIT;                                                       COMP    401
 VAR LATTR: ATTR;                                                          COMP    401
BEGIN TEST1(LPARENT,9);                                                    COMP    401
 VARIABLE(FSYS+[COMMA,RPARENT]);                                           COMP    401
 IF GATTR.TYPTR <> NIL THEN                                                COMP    401
  WITH GATTR.TYPTR↑ DO                                                     COMP    401
   IF FORM = FILES THEN                                                    COMP    401
    BEGIN IF NOT TEXTFILE THEN ERROR(116)                                  COMP    402
    END                                                                    COMP    402
   ELSE ERROR(116);                                                        COMP    402
 LATTR := GATTR;                                                           COMP    402
 IF SY = COMMA THEN                                                        COMP    402
  BEGIN INSYMBOL; EXPRESSION(FSYS+[RPARENT]);                              COMP    402
   IF NOT COMPTYPES(GATTR.TYPTR,INTPTR,FALSE) THEN ERROR(116);             COMP    402
   LOAD(GATTR,I); STORE(LATTR,I)                                           COMP    402
  END                                                                      COMP    402
 ELSE ERROR(20);                                                           COMP    402
 TEST1(RPARENT,4)                                                          COMP    403
END (*LINELIMIT*) ;                                                        COMP    403

PROCEDURE LOADPPTR(FATTR: ATTR; FORCT: BOOLEAN);                           COMP    403
 (*LOAD PPTR (ADDRESS) OF FILE DESCRIBED BY <FATTR,FORCT> INTO             COMP    403
   X1 (A1); UPON ENTRY, A1,X1 HAVE TO BE RESERVED*)                        COMP    403
 VAR I: REGNR; L: ADDRRANGE;                                               COMP    403
BEGIN                                                                      COMP    403
 WITH FATTR DO                                                             COMP    403
  BEGIN IF TYPTR↑.TEXTFILE THEN L := CHEFET-1 ELSE L := BINEFET-1;         COMP    403
   IF FORCT THEN                                                           COMP    404
    BEGIN CWDISPL := CWDISPL + L;                                          COMP    404
     IF VLEVEL IN LEVELS THEN GEN30(51B,1,BRG[VLEVEL],CWDISPL,0)           COMP    404
     ELSE BEGIN DECREFX(1);  (* TO FORCE LOADING OF A1/X1 *)               COMP    404
          LOADBASE(VLEVEL,TRUE,I); GEN30(52B,1,1,CWDISPL,0)                COMP    404
          END                                                              COMP    404
    END                                                                    COMP    404
   ELSE BEGIN LOAD(FATTR,I); DECREFX(I); GEN30(52B,1,I,L,0) END            COMP    404
```

```
    END;
  ARGS[1].ACONT := UNSPECADDR
END (*LOADPPTR*);                                                    000051

PROCEDURE READ;
  VAR PARAM,LATTR,FILATTR: ATTR; I,J: REGNR;
    LDRCT,GETIN,EXITLOOP: BOOLEAN;

BEGIN (*READ*)
  (*SET DEFAULT FILE ATTRIBUTES:*)
  WITH FILATTR DO
    BEGIN TYPTR := TEXTPTR; KIND := VARBL; WORDACC := DRCT;
      VLEVEL := 1; PCKD := FALSE;
      IF INPUTPTR = NIL THEN CWDISPL := 0
      ELSE CWDISPL := INPUTPTR↑.VADDR
    END;
  LDRCT := TRUE;
  (*SET PARAMETER ATTRIBUTES:*)
  WITH PARAM DO
    BEGIN TYPTR := TEXTPTR; KIND := VARBL; WORDACC := DRCT;
      VLEVEL := 0; CWDISPL := 3; PCKD := FALSE
    END;
  NEEDX(1,1,I);                    (*RESERVE A1/X1*)
  IF SY = LPARENT THEN
    BEGIN GETIN := TRUE;
      INSYMBOL; VARIABLE(FSYS+[COMMA,RPARENT]);
      IF GATTR.TYPTR <> NIL THEN
        IF GATTR.TYPTR↑.FORM = FILES THEN
          BEGIN IF NOT GATTR.TYPTR↑.TEXTFILE AND (LKEY = 9) (*READLN*)
            THEN ERROR(116);
            IF GATTR.WORDACC = DRCT THEN
              BEGIN FILATTR := GATTR; LDRCT := TRUE END
            ELSE
              BEGIN LOADADDRESS(GATTR,I);
                WITH FILATTR DO
                  BEGIN TYPTR := GATTR.TYPTR;
                    VLEVEL := LEVEL; CWDISPL := LC
                  END;
                STORE(FILATTR,I);
                LC := LC + 1;
                IF LC > LCMAX THEN LCMAX := LC;
                LDRCT := FALSE
              END;
          IF SY = RPARENT THEN
            BEGIN IF LKEY = 8 (*READ*) THEN ERROR(116);
              GETIN := FALSE
            END
          ELSE
            IF SY = COMMA THEN
              BEGIN INSYMBOL; VARIABLE(FSYS+[COMMA,RPARENT]) END
        END (*FORM = FILES*)
      ELSE IF INPUTPTR = NIL THEN ERROR(175);
      IF GETIN THEN
        (*LOOP UNTIL SY <> COMMA:*)
        REPEAT
          IF FILATTR.TYPTR↑.TEXTFILE AND NOT COMPTYPES(GATTR.TYPTR,
            CHARPTR,FALSE) THEN
            BEGIN DECREFX(1);
              IF COMPTYPES(GATTR.TYPTR,INTPTR,FALSE)OR COMPTYPES(
                GATTR.TYPTR,REALPTR,FALSE) THEN
                BEGIN
                  IF GATTR.PCKD THEN ERROR(127);
                  (*PASS FILE ADDRESS:*)
                  LATTR := FILATTR;
                  IF LDRCT THEN LOADADDRESS(LATTR,J) ELSE LOAD(LATTR,J);
                  PARAM.CWDISPL := 3; STORE(PARAM,J);
                  (*PASS ADDRESS OF VARIABLE TO BE READ:*)
                  LOADADDRESS(GATTR,I);
                  PARAM.CWDISPL := 4; STORE(PARAM,I);
                  CLEARREGS; GEN3(718,7,0,IC+1,2);
                  IF COMPTYPES(GATTR.TYPTR,INTPTR,FALSE) THEN
                    EQTOEXT(=RDI          =)
                  ELSE EQTOEXT(=RDR          =);
                END
              ELSE ERROR(116);
            END
          ELSE
            BEGIN LATTR := GATTR; LOADPPTR(FILATTR,LDRCT);
              WITH GATTR DO
                BEGIN TYPTR := FILATTR.TYPTR↑.FILTYPE; KIND := VARBL;
                  WORDACC := INDRCT; CWDISPL := 0; VWDISPL := 1;
```

```
COMP     404
COMP     404
COMP     405
COMP     405
COMP     405
COMP     405
COMP     405
COMP     405
COMP     405
COMP     405
COMP     405
COMP     406
COMP     406
COMP     406
COMP     406
COMP     406
COMP     406
COMP     406
COMP     406
COMP     406
COMP     407
COMP     407
COMP     407
COMP     407
COMP     407
COMP     407
COMP     407
COMP     407
COMP     408
COMP     408
COMP     408
COMP     408
COMP     408
COMP     408
COMP     408
COMP     408
COMP     409
COMP     409
COMP     409
COMP     409
COMP     409
COMP     409
COMP     409
COMP     410
COMP     410
COMP     410
COMP     410
COMP     410
COMP     410
COMP     410
COMP     410
COMP     411
COMP     411
COMP     411
COMP     411
COMP     411
COMP     411
COMP     411
COMP     411
COMP     412
COMP     412
COMP     412
COMP     412
COMP     412
COMP     412
COMP     412
```

```
          PCKD := FALSE                                                    COMP      412
          END;                                                             COMP      413
        XRGS[1].REFNR := 2;  (*TO PROTECT A1/X1*)                          COMP      413
        ASSIGNTO(LATTR);                                                   COMP      413
        NEEDX(1,1,I); (*RESET REFERENCE TO X1*)                            COMP      413
        IF FILATTR.TYPTR↑.TEXTFILE THEN                                    COMP      413
          RJTOEXT(=P.GETC    =)                                            COMP      413
        ELSE BEGIN GEN15(54B,1,1,1);                    000052             COMP      413
             RJTOEXT(=P.GETB    =)                                         COMP      413
           END;                                                            COMP      413
      END;                                                                 COMP      413
      EXITLOOP := SY <> COMMA;                                             COMP      413
      IF NOT EXITLOOP THEN                                                 COMP      414
       BEGIN NEEDX(1,1,I); (*RESERVE A1/X1*)                               COMP      414
        INSYMBOL; VARIABLE(FSYS+[COMMA,RPARENT])                           COMP      414
       END                                                                 COMP      414
    UNTIL EXITLOOP;                                                        COMP      414
  TEST1(RPARENT,4)                                                         COMP      414
  END (*SY = LPARENT*)                                                     COMP      414
 ELSE                                                                      COMP      414
  IF LKEY = 8 THEN (*READ*) ERROR(116)                                     COMP      414
  ELSE                                                                     COMP      415
   IF INPUTPTR = NIL THEN ERROR(175);                                      COMP      415
 IF LKEY = 9 (*READLN*) THEN                                               COMP      415
  BEGIN LOADPPTR(FILATTR,LDRCT);                                           COMP      415
   RJTOEXT(=P.GETLN   =)                                                   COMP      415
  END;                                                                     COMP      415
 IF NOT LDRCT THEN LC := LC - 1                                            COMP      415
END (*READ*) ;                                                             COMP      415
                                                                           COMP      415
PROCEDURE WRITE;                                                           COMP      415
 VAR PARAM,LATTR,FILATTR: ATTR;                                            COMP      416
  LSP: STP; I,J: REGNR; LDEF,LDRCT,PUTOUT,EXITLOOP: BOOLEAN;               COMP      416
                                                                           COMP      416
 PROCEDURE DEFLJMP(FL: INTEGER; FNAME: ALFA);                              COMP      416
  (*IF DEFAULF PASS DEFAULF FIELDLENGTH; JUMP TO EXTERNAL FNAME*)          COMP      416
  VAR I: REGNR;                                                            COMP      416
 BEGIN                                                                     COMP      416
  IF LDEF THEN                                                             COMP      416
   BEGIN NEEDX(6,7,I); GEN30(71B,I,0,FL,0);                                COMP      416
    PARAM.CWDISPL := 5; STORE(PARAM,I)                                     COMP      416
   END;                                                                    COMP      417
  CLEARREGS; GEN30(71B,7,0,IC+1,2);                                        COMP      417
   EQTOEXT(FNAME)                                                          COMP      417
 END (*DEFLJMP*) ;                                                         COMP      417
                                                                           COMP      417
BEGIN (*WRITE*)                                                            COMP      417
 (*SET DEFAULT FILE ATTRIBUTES:*)                                          COMP      417
 WITH FILATTR DO                                                           COMP      417
  BEGIN TYPTR := TEXTPTR; KIND := VARBL; WORDACC := DRCT;                  COMP      417
   VLEVEL := 1; CWDISPL := OUTPUTPTR↑.VADDR; PCKD := FALSE                 COMP      417
  END;                                                                     COMP      418
 LDRCT := TRUE;                                                            COMP      418
 (*SET PARAMETER ATTRIBUTES:*)                                             COMP      418
 WITH PARAM DO                                                             COMP      418
  BEGIN TYPTR := TEXTPTR; KIND := VARBL; WORDACC := DRCT;                  COMP      418
   VLEVEL := 0; CWDISPL := 3; PCKD := FALSE                                COMP      418
  END;                                                                     COMP      418
 NEEDX(1,1,I);                  (*RESERVE A1/X1*)                          COMP      418
 IF SY = LPARENT THEN                                                      COMP      418
  BEGIN PUTOUT := TRUE;                                                    COMP      418
   INSYMBOL; EXPRESSION(FSYS+[COMMA,COLON,RPARENT,IDENT]);                 COMP      419
   IF GATTR.TYPTR <> NIL THEN                                              COMP      419
    IF GATTR.TYPTR↑.FORM = FILES THEN                                      COMP      419
     BEGIN IF NOT GATTR.TYPTR↑.TEXTFILE AND (LKEY = 11) (*WRITELN*)        COMP      419
       THEN ERROR(116);                                                    COMP      419
      IF SY = RPARENT THEN                                                 COMP      419
       BEGIN IF LKEY = 10 (*WRITE*) THEN ERROR(116);                       COMP      419
        PUTOUT := FALSE                                                    COMP      419
       END                                                                 COMP      419
      ELSE                                                                 COMP      419
       IF SY <> COMMA THEN                                                 COMP      420
        BEGIN ERROR(116); SKIP(FSYS+[COMMA,RPARENT]) END;                  COMP      420
      IF GATTR.WORDACC = DRCT THEN                                         COMP      420
       BEGIN FILATTR := GATTR; LDRCT := TRUE END                           COMP      420
      ELSE                                                                 COMP      420
       BEGIN LOADADDRESS(GATTR,I);                                         COMP      420
        WITH FILATTR DO                                                    COMP      420
         BEGIN TYPTR := GATTR.TYPTR;                                       COMP      420
          VLEVEL := LEVEL; CWDISPL := LC                                   COMP      420
         END;                                                              COMP      420
```

```
      STORE(FILATTR,I);                                                  COMP    421
      LC := LC + 1;                                                      COMP    421
      IF LC > LCMAX THEN LCMAX := LC;                                    COMP    421
      LDRCT := FALSE                                                     COMP    421
    END;                                                                 COMP    421
   IF SY = COMMA THEN                                                    COMP    421
    BEGIN INSYMBOL;                                                      COMP    421
     EXPRESSION(FSYS+[COMMA,COLON,RPARENT,IDENT])                        COMP    421
    END                                                                  COMP    421
  END (*FORM = FILES*) ;                                                 COMP    421
IF PUTOUT THEN                                                           COMP    422
 (*LOOP UNTIL SY <> COMMA*)                                              COMP    422
 REPEAT                                                                  COMP    422
  IF FILATTR.TYPTR↑.TEXTFILE AND (NOT COMPTYPES(GATTR.TYPTR,             COMP    422
   CHARPTR,FALSE) OR (SY = COLON)) THEN                                  COMP    422
   BEGIN LSP := GATTR.TYPTR; DECREFX(1);                                 COMP    422
    (*PASS FILEADDRESS:*)                                                COMP    422
    LATTR := FILATTR;                                                    COMP    422
    IF LDRCT THEN LOADADDRESS(LATTR,J)                                   COMP    422
    ELSE LOAD(LATTR,J);                                                  COMP    422
    PARAM.CWDISPL := 3; STORE(PARAM,J);                                  COMP    423
    (*PASS VALUE TO BE OUTPUT:*)                                         COMP    423
    IF STRING(LSP) THEN                                                  COMP    423
     IF FULLWORDS(LSP↑.SIZE) = 1 THEN LOAD(GATTR,I)                      COMP    423
     ELSE LOADADDRESS(GATTR,I)                                           COMP    423
    ELSE LOAD(GATTR,I);                                                  COMP    423
    PARAM.CWDISPL := 4; STORE(PARAM,I);                                  COMP    423
    IF SY = COLON THEN                                                   COMP    423
     BEGIN INSYMBOL; B6DPL := 5;                                         COMP    423
      EXPRESSION(FSYS+[COMMA,COLON,RPARENT,IDENT]);                      COMP    423
      IF NOT COMPTYPES(GATTR.TYPTR,INTPTR,FALSE) THEN                    COMP    424
       ERROR(116);                                                       COMP    424
      (*PASS FILELENGTH:*)                                               COMP    424
      LOAD(GATTR,I);                                                     COMP    424
      PARAM.CWDISPL := 5; STORE(PARAM,I);                                COMP    424
      LDEF := FALSE                                                      COMP    424
     END                                                                 COMP    424
    ELSE LDEF := TRUE;                                                   COMP    424
    IF SY = COLON THEN                                                   COMP    424
     BEGIN INSYMBOL; B6DPL := 6;                                         COMP    424
      EXPRESSION(FSYS+[COMMA,RPARENT]);                                  COMP    425
      IF NOT COMPTYPES(GATTR.TYPTR,INTPTR,FALSE) THEN                    COMP    425
       ERROR(116);                                                       COMP    425
      IF NOT COMPTYPES(LSP,REALPTR,FALSE) THEN ERROR(124);               COMP    425
      (*PASS FRACTION LENGTH:*)                                          COMP    425
      LOAD(GATTR,I);                                                     COMP    425
      PARAM.CWDISPL := 6; STORE(PARAM,I);                                COMP    425
      DEFLJMP(20,=WRF       =)                                           COMP    425
     END                                                                 COMP    425
    ELSE                                                                 COMP    425
     IF SY = IDENT THEN                                                  COMP    426
      BEGIN                                                              COMP    426
       IF ID = =OCT       = THEN                                        COMP    426
        BEGIN                                                            COMP    426
         IF LSP <> NIL THEN                                              COMP    426
          IF LSP↑.FORM > POWER THEN ERROR(116);                          COMP    426
         INSYMBOL                                                        COMP    426
        END;                                                             COMP    426
       DEFLJMP(20,=WRO       =);                                         COMP    426
       TEST2([COMMA,RPARENT],6,FSYS)                                     COMP    426
      END (*SY = IDENT*)                                                 COMP    427
     ELSE                                                                COMP    427
      IF COMPTYPES(LSP,INTPTR,FALSE) THEN                                COMP    427
       DEFLJMP(10,=WRI       =)                                          COMP    427
      ELSE                                                               COMP    427
       IF COMPTYPES(LSP,REALPTR,FALSE) THEN                              COMP    427
        DEFLJMP(22,=WRE       =)                                         COMP    427
       ELSE                                                              COMP    427
        IF COMPTYPES(LSP,CHARPTR,FALSE) THEN                             COMP    427
         DEFLJMP(1,=WRC       =)                                         COMP    427
        ELSE                                                             COMP    428
         IF COMPTYPES(LSP,BOOLPTR,FALSE) THEN                            COMP    428
          DEFLJMP(10,=WRB       =)                                       COMP    428
         ELSE                                                            COMP    428
          IF LSP <> NIL THEN                                             COMP    428
           IF STRING(LSP) THEN                                           COMP    428
            WITH LSP↑ DO                                                 COMP    428
             BEGIN LSZ := ALFALENG*SIZE.WORDS+                           COMP    428
                SIZE.BITS DIV 6;                                         COMP    428
                (*PASS STRING LENGTH:*)                                  COMP    428
                NEEDX(6,7,I);                                            COMP    429
```

000053

```
                    GEN30(71B,I,0,LSZ,0);                                                    COMP     429
                    PARAM.CWDISPL := 6;                                                      COMP     429
                    STORE(PARAM,I);                                                          COMP     429
                    DEFLJMP(LSZ,=WRSN      =)                                                COMP     429
                  END                                                                        COMP     429
                ELSE ERROR(116);                                          000054           COMP     429
        END                                                                                  COMP     429
       ELSE                                                                                  COMP     429
        BEGIN LOADPPTR(FILATTR,LDRCT);                                                       COMP     429
         WITH LATTR DO                                                                       COMP     430
          BEGIN TYPTR := FILATTR.TYPTR↑.FILTYPE; KIND := VARBL;                              COMP     430
           WORDACC := INDRCT; CWDISPL := 0; VWDISPL := 1;                                    COMP     430
           PCKD := FALSE                                                                     COMP     430
          END;                                                                               COMP     430
         ASSIGNTC(LATTR);                                                                    COMP     430
         NEEDX(1,1,I); (*RESET REFERENCE TO X1*)                                             COMP     430
         IF FILATTR.TYPTR↑.TEXTFILE THEN                                                     COMP     430
          RJTOEXT(=P.PUTC     =)                                                             COMP     430
         ELSE BEGIN GEN15(54B,1,1,1);                                                        COMP     430
              RJTOEXT(=P.PUTB     =)                                                         COMP     430
             END;                                                                            COMP     431
        END;                                                                                 COMP     431
       B6DPL := 3;                                                                           COMP     431
       EXITLOOP := SY <> COMMA;                                                              COMP     431
       IF NOT EXITLOOP THEN                                                                  COMP     431
        BEGIN INSYMBOL; NEEDX(1,1,I); (*RESERVE A1/X1*)                                      COMP     431
         EXPRESSION(FSYS+[COMMA,COLON,RPARENT,IDENT])                                        COMP     431
        END;                                                                                 COMP     431
      UNTIL EXITLOOP;                                                                        COMP     431
     TEST1(RPARENT,4)                                                                        COMP     431
    END (*SY = LPARENT*)                                                                     COMP     432
   ELSE IF LKEY = 10 THEN (*WRITE*) ERROR(116);                                              COMP     432
   IF LKEY = 11 THEN (*WRITELN*)                                                             COMP     432
    BEGIN LOADPPTR(FILATTR,LDRCT);                                                           COMP     432
     RJTOEXT(=P.PUTLN    =);                                                                 COMP     432
    END;                                                                                     COMP     432
   IF NOT LDRCT THEN LC := LC - 1                                                            COMP     432
 END (*WRITE*) ;                                                                             COMP     432
                                                                                             COMP     432
 PROCEDURE MESSAGE;                                                                          COMP     433
  VAR I: REGNR;                                                                              COMP     433
 BEGIN TEST1(LPARENT,9);                                                                     COMP     433
  EXPRESSION(FSYS+[RPARENT]);                                                                COMP     433
  LOADADDRESS(GATTR,I); BXIXJ(1,I);                                                          COMP     433
  IF GATTR.TYPTR <> NIL THEN                                                                 COMP     433
   IF STRING(GATTR.TYPTR) THEN                                                               COMP     433
    GEN30(71B,2,0,GATTR.TYPTR↑.SIZE.WORDS*ALFALENG                                           COMP     433
      +GATTR.TYPTR↑.SIZE.BITS DIV 6,0)                                                       COMP     433
   ELSE ERROR(116);                                                                          COMP     433
  RJTOEXT(=P.MSG      =);                                                                    COMP     434
  TEST1(RPARENT,4)                                                                           COMP     434
 END (*MESSAGE*) ;                                                                           COMP     434
                                                                                             COMP     434
 PROCEDURE PAGE;                                                                             COMP     434
  VAR I: REGNR;                                                                              COMP     434
 BEGIN TEST1(LPARENT,9);                                                                     COMP     434
  CLEARREGS; XRGS[1].XCONT := OTHER; (*RESERVES A1/X1*)                                      COMP     434
  VARIABLE(FSYS+[RPARENT]);                                                                  COMP     434
  CLEARREGS; (*TO PREVENT BX1... AND GUARANTEE SA1...*)                                      COMP     434
  IF GATTR.TYPTR <> NIL THEN                                                                 COMP     435
   BEGIN                                                                                     COMP     435
    IF GATTR.WORDACC <> DRCT THEN                                                            COMP     435
     XRGS[GATTR.VWDISPL].XCONT := OTHER;                                                     COMP     435
    WITH GATTR.TYPTR↑ DO                                                                     COMP     435
     IF FORM = FILES THEN                                                                    COMP     435
      BEGIN IF NOT TEXTFILE THEN ERROR(116);                                                 COMP     435
       GATTR.CWDISPL := GATTR.CWDISPL + CHEFET - 1                                           COMP     435
      END                                                                                    COMP     435
     ELSE ERROR(116)                                                                         COMP     435
   END;                                                                                      COMP     435
  LOAD(GATTR,I); (* I = 1 IS GUARANTEED *)                                                   COMP     436
  (*BECAUSE PUTLN AND PUTC GUARANTEE TO LEAVE A1 UNCHANGED, IT NEED                          COMP     436
  NOT BE SAVED*)                                                                             COMP     436
  RJTOEXT(=P.PUTLN    =);                                                                    COMP     436
  GEN30(71B,6,0,ORD(=1=),0); GEN15(54B,1,1,0); GEN15(53B,6,1,0);                             COMP     436
  RJTOEXT(=P.PUTC     =);                                                                    COMP     436
  GEN15(54B,1,1,0);                                                                          COMP     436
  RJTOEXT(=P.PUTLN    =);                                                                    COMP     436
  TEST1(RPARENT,4)                                                                           COMP     436
 END (*PAGE*) ;                                                                              COMP     437
                                                                                             COMP     437
```

```
  PROCEDURE TIMEDATE;                                                    COMP      437
   VAR I: REGNR;                                                         COMP      437
  BEGIN TEST1(LPARENT,9);                                                COMP      437
   VARIABLE(FSYS+[RPARENT]);                                             COMP      437
   IF NOT COMPTYPES(GATTR.TYPTR,ALFAPTR,FALSE) THEN ERROR(116);          COMP      437
   LOADADDRESS(GATTR,I); BXIXJ(1,I);                                     COMP      437
   IF LKEY = 14 THEN (*TIME*) RJTOEXT(=P.TIME    =)                      COMP      437
   ELSE (*DATE*) RJTOEXT(=P.DATE    =);                                  COMP      437
   TEST1(RPARENT,4)                                                      COMP      438
  END (*TIMEDATE*) ;                                                     COMP      438
                                                                  000055 COMP      438
  PROCEDURE HALT;                                                        COMP      438
  BEGIN CLEARREGS;                                                       COMP      438
   GEN30(51B,0,0,IC,0); GEN30(04B,0,0,HALTERR,0)                         COMP      438
  END (*HALT*) ;                                                         COMP      438
                                                                         COMP      438
PROCEDURE PACK;                                                          COMP      438
  VAR BITS: BITRANGE; FW,LADDR: ADDRRANGE;                               COMP      438
    I,J,K,R,S,T: REGNR; LSP,LSP1: STP; LATTR: ATTR;                      COMP      439
    PW,EPW,LOW,HIGH,LMIN,LMAX: INTEGER; LEFTADJ: BOOLEAN;                COMP      439
                                                                         COMP      439
  PROCEDURE PACKWORD(NROFELS: BITRANGE);                                 COMP      439
   VAR LADDR: ADDRRANGE; SHFT: BITRANGE;                                 COMP      439
  BEGIN NEEDB(S); GEN30(61B,S,R,NROFELS,0);                              COMP      439
   GEN15(13B,J,J,J); NOOP; LADDR := IC;                                  COMP      439
   GEN15(56B,K,R,0);                                                     COMP      439
   IF LEFTADJ THEN                                                       COMP      439
    BEGIN GEN15(11B,K,I,K); GEN15(12B,J,J,K);                            COMP      439
     GEN15(20B,J,0,BITS)                                                 COMP      440
    END                                                                  COMP      440
   ELSE                                                                  COMP      440
    BEGIN GEN15(15B,K,K,I); GEN15(20B,J,0,BITS);                         COMP      440
     GEN15(12B,J,J,K)                                                    COMP      440
    END;                                                                 COMP      440
   GEN15(66B,R,R,1); GEN30(05B,R,S,LADDR,2);                             COMP      440
   BRGS[S].BCONT := FREE;                                                COMP      440
   SHFT := WORDSIZE - NROFELS*BITS;                                      COMP      440
   IF SHFT > 0 THEN GEN15(20B,J,0,SHFT)                                  COMP      440
  END (*PACKWORD*) ;                                                     COMP      441
                                                                         COMP      441
BEGIN (*PACK*)                                                           COMP      441
  TEST1(LPARENT,9);                                                      COMP      441
  VARIABLE(FSYS+[COMMA,RPARENT]); LATTR := GATTR;                        COMP      441
  LSP := NIL; LSP1 := NIL; LOW := 0; HIGH := 0; LEFTADJ := FALSE;        COMP      441
  IF GATTR.TYPTR <> NIL THEN                                             COMP      441
   WITH GATTR.TYPTR↑ DO                                                  COMP      441
    IF FORM = ARRAYS THEN                                                COMP      441
     IF NOT PCKDARR THEN                                                 COMP      441
      BEGIN LSP := INXTYPE; LSP1 := AELTYPE;                             COMP      442
       IF LSP <> NIL THEN GETBOUNDS(LSP,LOW,HIGH);                       COMP      442
       IF LSP1 <> NIL THEN LEFTADJ := LSP1↑.FORM IN [ARRAYS,RECORDS]     COMP      442
      END                                                                COMP      442
     ELSE ERROR(116)                                                     COMP      442
    ELSE ERROR(116);                                                     COMP      442
  TEST1(COMMA,20);                                                       COMP      442
  EXPRESSION(FSYS+[COMMA,RPARENT]);                                      COMP      442
  IF NOT COMPTYPES(GATTR.TYPTR,LSP,FALSE) THEN ERROR(116);               COMP      442
  (*LOAD SOURCE ADDRESS (LATTR[GATTR]) INTO B-R:*)                        COMP      442
  NEEDB(R);                                                              COMP      443
  IF (LATTR.TYPTR <> NIL) AND (GATTR.TYPTR <> NIL) THEN                  COMP      443
   BEGIN LATTR.CWDISPL := LATTR.CWDISPL - LOW;                           COMP      443
    IF GATTR.KIND = CST THEN                                             COMP      443
     BEGIN LATTR.CWDISPL := LATTR.CWDISPL + GATTR.CVAL.IVAL;             COMP      443
      LOADADDRESS(LATTR,I)                                               COMP      443
     END                                                                 COMP      443
    ELSE                                                                 COMP      443
     BEGIN LOADADDRESS(LATTR,I); LOAD(GATTR,J);                          COMP      443
      OPERATION(36B,K,I,J); I := K                                       COMP      443
     END;                                                                COMP      444
    GEN15(63B,R,I,0); DECREFX(I)                                         COMP      444
   END;                                                                  COMP      444
  TEST1(COMMA,20);                                                       COMP      444
  NEEDX(6,7,J); ARGS[J].ACONT := UNSPECADDR;                             COMP      444
  VARIABLE(FSYS+[RPARENT]);                                              COMP      444
  FW := 1; PW := 0; EPW := 60; BITS := 1;                                COMP      444
  IF GATTR.TYPTR <> NIL THEN                                             COMP      444
   WITH GATTR.TYPTR↑ DO                                                  COMP      444
    IF FORM = ARRAYS THEN                                                COMP      444
     IF PCKDARR AND PARTWORDELS AND                                      COMP      445
      COMPTYPES(AELTYPE,LSP1,FALSE) AND COMPTYPES(INXTYPE,               COMP      445
      LSP,FALSE) THEN                                                    COMP      445
```

```
      BEGIN LMIN := 0; LMAX := 0;                                          COMP    445
       IF INXTYPE <> NIL THEN GETBOUNDS(INXTYPE,LMIN,LMAX);                COMP    445
       IF LMAX - LMIN > HIGH - LOW THEN ERROR(116);                        COMP    445
       IF AELTYPE <> NIL THEN BITS := AELTYPE↑.SIZE.BITS;                  COMP    445
       EPW := ELSPERWORD;                                                  COMP    445
       FW := (LMAX-LMIN+1) DIV EPW;                                        COMP    445
       PW := (LMAX-LMIN+1) - FW*EPW                                        COMP    445
      END                                                                  COMP    446
     ELSE ERROR(116)                                          000056       COMP    446
    ELSE ERROR(116);                                                       COMP    446
  NEEDX(1,5,K); ARGS[K].ACONT := UNSPECADDR;                               COMP    446
  NEEDX(0,7,I);                                                            COMP    446
  IF LEFTADJ THEN GEN15(43B,I,0,BITS)                                      COMP    446
  ELSE GEN15(43B,I,0,WORDSIZE-BITS);                                       COMP    446
  IF FW > 0 THEN                                                           COMP    446
   BEGIN                                                                   COMP    446
    IF FW <> 1 THEN                                                        COMP    446
     BEGIN (*SAVE B4/B6:*)                                                 COMP    447
      NEEDX(0,7,T); GEN15(76B,T,4,0);                                      COMP    447
      GEN15(20B,T,0,18); GEN15(76B,J,6,0);                                 COMP    447
      GEN15(12B,T,T,J); LOADADDRESS(GATTR,S); DECREFX(S);                  COMP    447
      GEN15(63B,4,S,0); GEN30(61B,6,4,FW,0); NOOP; LADDR := IC             COMP    447
     END;                                                                  COMP    447
    PACKWORD(EPW);                                                         COMP    447
    IF FW = 1 THEN STORE(GATTR,J)                                          COMP    447
    ELSE                                                                   COMP    447
     BEGIN GEN15(56B,J,4,0); DECREFX(J); GEN15(66B,4,4,1);                 COMP    447
      GEN30(05B,4,6,LADDR,2); GEN15(63B,6,T,0); GEN15(20B,T,0,42);         COMP    448
      GEN15(63B,4,T,0); DECREFX(T)                                         COMP    448
     END                                                                   COMP    448
   END (*FW > 0*) ;                                                        COMP    448
  IF PW > 0 THEN                                                           COMP    448
   BEGIN PACKWORD(PW);                                                     COMP    448
    IF FW > 0 THEN GEN15(54B,J,J,1) ELSE STORE(GATTR,J);                   COMP    448
   END;                                                                    COMP    448
  CLEARREGS;                                                               COMP    448
  TEST1(RPARENT,4)                                                         COMP    448
END (*PACK*) ;                                                             COMP    449
                                                                           COMP    449
PROCEDURE UNPACK;                                                          COMP    449
 VAR BITS: BITRANGE; FW,LADDR: ADDRRANGE;                                  COMP    449
   I,J,K,R,T: REGNR; LSP,LSP1: STP; SOURCE,DEST: ATTR;                     COMP    449
   EPW,PW,LOW,HIGH,LMIN,LMAX: INTEGER; LMODE: (USRADJ,SRADJ,USLADJ);       COMP    449
                                                                           COMP    449
 PROCEDURE UNPACKWORD(NROFELS: BITRANGE);                                  COMP    449
  VAR LADDR: ADDRRANGE; S: REGNR;                                          COMP    449
 BEGIN NEEDB(S); GEN30(61B,S,R,NROFELS,0);                                 COMP    449
  NOOP; LADDR := IC;                                                       COMP    450
  XRGS[K].XCONT := OTHER;                                                  COMP    450
  CASE LMODE OF                                                            COMP    450
   USRADJ:                                                                 COMP    450
    BEGIN GEN15(20B,K,0,BITS); GEN15(15B,J,K,I) END;                       COMP    450
   SRADJ:                                                                  COMP    450
    BEGIN GEN15(10B,J,K,0); GEN15(20B,K,0,BITS);                           COMP    450
     GEN15(21B,J,0,WORDSIZE-BITS)                                          COMP    450
    END;                                                                   COMP    450
   USLADJ:                                                                 COMP    450
    BEGIN GEN15(11B,J,I,K); GEN15(20B,K,0,BITS) END                        COMP    451
  END;                                                                     COMP    451
  GEN15(56B,J,R,0); GEN15(66B,R,R,1);                                      COMP    451
  GEN30(05B,R,S,LADDR,2); BRGS[S].BCONT := FREE                            COMP    451
 END (*UNPACKWORD*);                                                       COMP    451
                                                                           COMP    451
BEGIN (*UNPACK*)                                                           COMP    451
 TEST1(LPARENT,9);                                                         COMP    451
 EXPRESSION(FSYS+[COMMA,RPARENT]); SOURCE := GATTR;                        COMP    451
 LSP := NIL; LSP1 := NIL; LMIN := 0; LMAX := 0;                            COMP    451
 FW := 1; PW := 0; EPW := 60; BITS := 1; LMODE := USRADJ;                  COMP    452
 IF GATTR.TYPTR <> NIL THEN                                                COMP    452
  WITH GATTR.TYPTR↑ DO                                                     COMP    452
   IF FORM = ARRAYS THEN                                                   COMP    452
    IF PCKDARR AND PARTWORDELS THEN                                        COMP    452
     BEGIN LSP := INXTYPE; LSP1 := AELTYPE;                                COMP    452
      IF LSP <> NIL THEN GETBOUNDS(LSP,LMIN,LMAX);                         COMP    452
      IF LSP1 <> NIL THEN                                                  COMP    452
       WITH LSP1↑ DO                                                       COMP    452
        BEGIN BITS := SIZE.BITS;                                           COMP    452
         IF FORM = SUBRANGE THEN                                           COMP    453
          BEGIN IF MIN.IVAL < 0 THEN LMODE := SRADJ                        COMP    453
          END                                                              COMP    453
         ELSE                                                              COMP    453
```

```
          IF FORM IN [ARRAYS,RECORDS] THEN LMODE := USLADJ                    COMP    453
        END;                                                                  COMP    453
       EPW := ELSPERWORD;                                                     COMP    453
       FW := (LMAX - LMIN + 1) DIV EPW;                                       COMP    453
       PW := (LMAX - LMIN + 1) - FW*EPW                                       COMP    453
      END                                                          000057     COMP    453
     ELSE ERROR(116)                                                         COMP    454
    ELSE ERROR(116);                                                         COMP    454
  TEST1(COMMA,20);                                                           COMP    454
  VARIABLE(FSYS+[COMMA,RPARENT]); DEST := GATTR;                             COMP    454
  IF GATTR.TYPTR <> NIL THEN                                                 COMP    454
   WITH GATTR.TYPTR^ DO                                                      COMP    454
    IF FORM = ARRAYS THEN                                                    COMP    454
     IF NOT PCKDARR AND COMPTYPES(AELTYPE,LSP1,FALSE)                        COMP    454
      AND COMPTYPES(INXTYPE,LSP,FALSE) THEN                                  COMP    454
      BEGIN LOW := 0; HIGH := 0;                                             COMP    454
       IF INXTYPE <> NIL THEN GETBOUNDS(INXTYPE,LOW,HIGH);                   COMP    455
       IF LMAX - LMIN > HIGH - LOW THEN ERROR(116);                          COMP    455
      END                                                                    COMP    455
     ELSE ERROR(116)                                                         COMP    455
    ELSE ERROR(116);                                                         COMP    455
  TEST1(COMMA,20);                                                           COMP    455
  EXPRESSION(FSYS+[RPARENT]);                                                COMP    455
  IF NOT COMPTYPES(GATTR.TYPTR,LSP,FALSE) THEN ERROR(116);                   COMP    455
  (*LOAD DESTINATION ADDRESS (DEST[GATTR]) INTO B-R:*)                        COMP    455
  NEEDB(R);                                                                  COMP    455
  IF (DEST.TYPTR <> NIL)AND (GATTR.TYPTR <> NIL) THEN                        COMP    456
   BEGIN DEST.CWDISPL := DEST.CWDISPL - LOW;                                 COMP    456
    IF GATTR.KIND = CST THEN                                                 COMP    456
     BEGIN DEST.CWDISPL := DEST.CWDISPL + GATTR.CVAL.IVAL;                   COMP    456
      LOADADDRESS(DEST,I)                                                    COMP    456
     END                                                                     COMP    456
    ELSE                                                                     COMP    456
     BEGIN LOADADDRESS(DEST,I); LOAD(GATTR,J);                               COMP    456
      OPERATION(36B,K,I,J); I := K                                           COMP    456
     END;                                                                    COMP    456
    GEN15(63B,R,I,0); DECREFX(I)                                             COMP    457
   END;                                                                      COMP    457
  NEEDX(6,7,J); ARGS[J].ACONT := UNSPECADDR; NEEDX(0,7,I);                   COMP    457
  IF LMODE = USRADJ THEN GEN15(43B,I,0,WORDSIZE-BITS)                        COMP    457
  ELSE                                                                       COMP    457
   IF LMODE = USLADJ THEN GEN15(43B,I,0,BITS);                               COMP    457
  IF FW > 0 THEN                                                             COMP    457
   BEGIN                                                                     COMP    457
    IF (FW > 1)OR (PW > 0) THEN                                              COMP    457
     BEGIN NEEDX(0,7,T); GEN15(76B,T,4,0);                                   COMP    457
      LOADADDRESS(SOURCE,K); GEN15(63B,4,K,0);                               COMP    458
      DECREFX(K)                                                             COMP    458
     END;                                                                    COMP    458
    IF FW > 1 THEN                                                           COMP    458
     BEGIN GEN15(20B,T,0,18); GEN15(76B,J,6,0);                              COMP    458
      GEN15(12B,T,T,J); GEN30(61B,6,4,FW,0);                                 COMP    458
      NOOP; LADDR := IC;                                                     COMP    458
     END;                                                                    COMP    458
    IF (FW = 1)AND (PW = 0) THEN                                             COMP    458
     LOAD(SOURCE,K)                                                          COMP    458
    ELSE                                                                     COMP    459
     BEGIN NEEDX(1,5,K); ARGS[K].ACONT := UNSPECADDR;                        COMP    459
      GEN15(56B,K,4,0)                                                       COMP    459
     END;                                                                    COMP    459
    UNPACKWORD(EPW);                                                         COMP    459
    IF FW > 1 THEN                                                           COMP    459
     BEGIN GEN15(66B,4,4,1); GEN30(05B,4,6,LADDR,2);                         COMP    459
      GEN15(63B,6,T,0); GEN15(20B,T,0,42)                                    COMP    459
     END                                                                     COMP    459
   END;                                                                      COMP    459
  IF PW > 0 THEN                                                             COMP    460
   BEGIN                                                                     COMP    460
    IF FW > 0 THEN                                                           COMP    460
     BEGIN IF FW = 1 THEN GEN15(56B,K,4,1) ELSE GEN15(56B,K,4,0);            COMP    460
      GEN15(63B,4,T,0); DECREFX(T)                                           COMP    460
     END                                                                     COMP    460
    ELSE                                                                     COMP    460
     LOAD(SOURCE,K);                                                         COMP    460
    UNPACKWORD(PW);                                                          COMP    460
   END                                                                       COMP    460
  ELSE                                                                       COMP    461
   IF FW > 1 THEN                                                            COMP    461
    BEGIN GEN15(63B,4,T,0); DECREFX(T) END;                                  COMP    461
  CLEARREGS;                                                                 COMP    461
  TEST1(RPARENT,4)                                                           COMP    461
```

```
END (*UNPACK*);

 PROCEDURE NEWDISPOSE;
  LABEL 1;
  VAR LSP,LSP1: STP; LVAL: VALU; LSIZE: ADDRRANGE; I: REGNR;
    LNAME: ALFA;
 BEGIN TEST1(LPARENT,9);
  VARIABLE(FSYS+[COMMA,RPARENT]);
  LSP := NIL; LSIZE := 0;
  IF GATTR.TYPTR <> NIL THEN
   WITH GATTR.TYPTR↑ DO
    IF FORM = POINTER THEN
     BEGIN  IF GATTR.PCKD THEN ERROR(127);
      IF ELTYPE <> NIL THEN
       BEGIN LSIZE := FULLWORDS(ELTYPE↑.SIZE);
        IF ELTYPE↑.FORM = RECORDS THEN LSP := ELTYPE↑.RECVAR
       END
     END
    ELSE ERROR(116);
  WHILE SY = COMMA DO
   BEGIN INSYMBOL; CONSTANT(FSYS+[COMMA,RPARENT],LSP1,LVAL);
    IF LSP = NIL THEN ERROR(158)
    ELSE
      IF LSP↑.TAGFIELDP <> NIL THEN
       IF STRING(LSP1)OR (LSP1 = REALPTR) THEN ERROR(159)
       ELSE
        IF COMPTYPES(LSP↑.TAGFIELDP↑.IDTYPE,LSP1,FALSE) THEN
         BEGIN
          LSP1 := LSP↑.FSTVAR;
          WHILE LSP1 <> NIL DO
           WITH LSP1↑ DO
            IF VARVAL.IVAL = LVAL.IVAL THEN
             BEGIN LSP := SUBVAR; LSIZE := FULLWORDS(SIZE);
              GOTO 1
             END
            ELSE LSP1 := NXTVAR;
          LSIZE := FULLWORDS(LSP↑.SIZE); LSP := NIL
         END
        ELSE ERROR(116);
 1:  END (*WHILE*) ;
  LOADADDRESS(GATTR,I); BXIXJ(1,I);
  GEN30(61B,7,0,LSIZE,0);
  IF LKEY = 19 THEN LNAME := ≡P.NEW     ≡
  ELSE LNAME := ≡P.DISP    ≡;
  RJTOEXT(LNAME);
  TEST1(RPARENT,4)
 END (*NEWDISPOSE*) ;

 PROCEDURE RELEASE;
  VAR I: REGNR;
 BEGIN TEST1(LPARENT,9);
  VARIABLE(FSYS+[COMMA,RPARENT]);
  IF GATTR.TYPTR <> NIL THEN
   IF GATTR.TYPTR↑.FORM = POINTER THEN
    BEGIN LOAD(GATTR,I); GEN15(63B,4,I,0); DECREFX(I) END
   ELSE ERROR(116);
  TEST1(RPARENT,4)
 END (*RELEASE*) ;


 PROCEDURE STDFLFUNCS;
 BEGIN
  IF SY = LPARENT THEN
   BEGIN INSYMBOL; VARIABLE(FSYS+[RPARENT]);
    TEST1(RPARENT,4)
   END
  ELSE
   WITH GATTR DO
    BEGIN TYPTR := TEXTPTR; KIND := VARBL; WORDACC := DRCT;
     VLEVEL := 1; PCKD := FALSE;
     IF INPUTPTR = NIL THEN
      BEGIN ERROR(175); CWDISPL := 0 END
     ELSE WITH INPUTPTR↑ DO
      BEGIN CWDISPL:=VADDR; TYPTR:=IDTYPE END;
    END;
  IF GATTR.TYPTR <> NIL THEN
   WITH GATTR, TYPTR↑ DO
    IF FORM = FILES THEN
     BEGIN
      IF LKEY = 3 (*EOLN*) THEN
       BEGIN IF NOT TEXTFILE THEN ERROR(125);
```

```
      CWDISPL := CWDISPL + CHEFET -1 (*P-PTR*)                                  COMP    469
     END                                                                        COMP    469
    ELSE                                                                        COMP    469
     IF TEXTFILE THEN CWDISPL := CWDISPL + CHEFET                               COMP    469
     ELSE CWDISPL := CWDISPL + BINEFET; (*EFET*)          000059                COMP    470
     LOAD(GATTR,I);                                                             COMP    470
    IF LKEY = 2 (*EOS*) THEN                                                    COMP    470
     BEGIN IF NOT SEGFILE THEN ERROR(125) END                                   COMP    470
    ELSE                                                                        COMP    470
     IF LKEY = 1 (*EOF*) THEN                                                   COMP    470
      IF SEGFILE THEN                                                           COMP    470
       BEGIN NEEDX(0,7,J); GEN15(228,J,1,I);                                    COMP    470
        DECREFX(I); I := J                                                      COMP    470
       END;                                                                     COMP    470
    TYPTR := BOOLPTR; KIND := COND; CONDCD := PL; CDR := I;                     COMP    471
   END                                                                          COMP    471
  ELSE ERROR(125);                                                              COMP    471
END (*STDFLFUNCS*);                                                             COMP    471

PROCEDURE STDINLINEFUNCS;                                                       COMP    471
 VAR I,J,K: REGNR;                                                              COMP    471
     LSYS: SETOFSYS;                                                            COMP    471
BEGIN TEST1(LPARENT,9);                                                         COMP    471
 LSYS:=[RPARENT];                                                               COMP    471
 IF LKEY = 7 THEN LSYS:=[COMMA,RPARENT]; (*TRUNC MAY HAVE 2 ARG *)              COMP    472
 EXPRESSION(FSYS+LSYS); LOAD(GATTR,I);                                          COMP    472
 IF LKEY IN [4..10,13..15] THEN                                                 COMP    472
  NEEDX(0,7,K); (*FUNCTION NEEDING ANOTHER X REGISTER*)                         COMP    472
 IF LKEY IN [5..8] THEN (*FUNCTION ONLY ALLOWING REAL ARGS*)                    COMP    472
  IF NOT COMPTYPES(GATTR.TYPTR,REALPTR,FALSE) THEN ERROR(125);                  COMP    472
 IF LKEY IN [4,12] THEN  (*FUNCTION ONLY ALLOWING INTEGER ARGS*)                COMP    472
  IF NOT COMPTYPES(GATTR.TYPTR,INTPTR,FALSE) THEN ERROR(125);                   COMP    472
 CASE LKEY OF                                                                   COMP    472
  4: (*ODD*)                                                                    COMP    472
     BEGIN GEN15(10B,K,I,I);                                                    COMP    473
     GEN15(20B,K,0,59(*WORDSIZE-1*));                                           COMP    473
     GEN15(13B,K,I,K); DECREFX(I);                                              COMP    473
     WITH GATTR DO                                                              COMP    473
      BEGIN TYPTR := BOOLPTR; KIND := COND; CONDCD := PL;                       COMP    473
       CDR := K                                                                 COMP    473
      END                                                                       COMP    473
     END;                                                                       COMP    473
  5: (*UNDEFINED*)                                                              COMP    473
     BEGIN GEN15(76B,K,1,0);                                                    COMP    474
     IF PC.CP >= 3 THEN GEN30(03B,7,I,IC+2,2)                                   COMP    474
      ELSE GEN30(03B,7,I,IC+1,2);                                               COMP    474
     GEN30(03B,5,I,IC+1,2);                                                     COMP    474
     GEN15(13B,K,K,K); NOOP;                                                    COMP    474
     GATTR.TYPTR := BOOLPTR;                                                    COMP    474
     END;                                                                       COMP    474
  6: (*ROUND*)                                                                  COMP    474
     BEGIN NEEDX(0,7,J);                                                        COMP    474
     GEN15(13B,K,K,K); GEN15(27B,K,0,K); GEN15(30B,J,I,K);                      COMP    474
     GEN15(32B,K,I,K); GEN15(34B,K,J,K); GEN15(26B,K,0,K);                      COMP    474
     GEN15(13B,J,J,J); GEN15(36B,K,K,J); DECREFX(J)                             COMP    475
     END;                                                                       COMP    475
  7: (*TRUNC*)                                                                  COMP    475
     BEGIN                                                                      COMP    475
      NEEDB(J); DECREFX(I); GEN15(26B,K,J,I);                                   COMP    475
      IF SY = COMMA  THEN                                                       COMP    475
       BEGIN INSYMBOL; EXPRESSION(FSYS+[RPARENT]);                              COMP    475
        IF GATTR.TYPTR <> NIL THEN                                              COMP    475
         IF COMPTYPES(GATTR.TYPTR,INTPTR,FALSE) THEN                            COMP    475
          IF GATTR.KIND = CST THEN                                              COMP    475
           BEGIN GEN30(61B,J,J,GATTR.CVAL.IVAL,0);                              COMP    476
                 GATTR.KIND:=EXPR                                               COMP    476
           END                                                                  COMP    476
          ELSE BEGIN LOAD(GATTR,I); GEN15(63B,J,I,J); DECREFX(I)                COMP    476
               END                                                              COMP    476
         ELSE ERROR(125)                                                        COMP    476
       END;                                                                     COMP    476
      GEN15(22B,K,J,K); BRGS[J].BCONT:=FREE; NEEDX(0,7,I);                      COMP    476
      GEN15(13B,I,I,I); GEN15(36B,K,K,I)                                        COMP    476
     END;                                                                       COMP    476
  8: (*EXPO*)                                                                   COMP    477
     BEGIN NEEDB(J);                                                            COMP    477
     GEN15(26B,K,J,I); GEN30(71B,K,J,47,0);                                     COMP    477
     BRGS[J].BCONT := FREE;                                                     COMP    477
     END;                                                                       COMP    477
  9: (*ABS*)                                                                    COMP    477
     BEGIN IF COMPTYPES(GATTR.TYPTR,INTPTR,FALSE) OR                            COMP    477
```

```
              COMPTYPES(GATTR.TYPTR,REALPTR,FALSE) THEN                    COMP    477
       BEGIN GEN15(10B,K,I,I); GEN15(21B,K,0,59(*WORDSIZE-1*));            COMP    477
        GEN15(13B,K,K,I)                                                   COMP    477
       END                                                                 COMP    478
      ELSE ERROR(125)                                                      COMP    478
      END;                                                      000060     COMP    478
  10: (*SQR*)                                                              COMP    478
      BEGIN IF COMPTYPES(GATTR.TYPTR,INTPTR,FALSE) THEN                    COMP    478
       GEN15(42B,K,I,I)                                                    COMP    478
      ELSE                                                                 COMP    478
       IF COMPTYPES(GATTR.TYPTR,REALPTR,FALSE) THEN                        COMP    478
        GEN15(41B,K,I,I)                                                   COMP    478
       ELSE ERROR(125)                                                     COMP    478
      END;                                                                 COMP    479
  11: (*ORD*)                                                              COMP    479
      BEGIN IF GATTR.TYPTR <> NIL THEN                                     COMP    479
       IF GATTR.TYPTR↑.FORM > POWER THEN ERROR(125)                        COMP    479
      END;                                                                 COMP    479
  12: (*CHR*)                                                              COMP    479
      GATTR.TYPTR := CHARPTR;                                              COMP    479
  13,14: (*PRED,SUCC*)                                                     COMP    479
      BEGIN IF GATTR.TYPTR <> NIL THEN                                     COMP    479
       IF (GATTR.TYPTR↑.FORM > SUBRANGE) OR                                COMP    479
          COMPTYPES(GATTR.TYPTR,REALPTR,FALSE) THEN ERROR(125);            COMP    480
      GEN15(76B,K,1,0);                                                    COMP    480
      GEN15(54B+LKEY(*37B FOR PRED, 36B FOR SUCC*),K,I,K)                  COMP    480
      END;                                                                 COMP    480
  15: (*CARD*)                                                             COMP    480
      BEGIN IF GATTR.TYPTR <> NIL THEN                                     COMP    480
       IF GATTR.TYPTR↑.FORM <> POWER THEN ERROR(125);                      COMP    480
      GEN15(47B,K,0,I)                                                     COMP    480
      END                                                                  COMP    480
   END (*CASE*);                                                           COMP    480
  IF LKEY IN [5..10,13..15] THEN (*FUNCTIONS RETURNING RESULT              COMP    481
                                    IN K REGNR*)                           COMP    481
   BEGIN DECREFX(I); GATTR.EXPREG := K END;                                COMP    481
  IF LKEY IN [6..8,11,15] THEN (*FUNCTIONS FORCING INTEGER RESULT*)        COMP    481
   GATTR.TYPTR := INTPTR;                                                  COMP    481
  TEST1(RPARENT,4)                                                         COMP    481
 END (*STDINLINEFUNCS*);                                                   COMP    481
                                                                           COMP    481
PROCEDURE CLOCKF;                                                          COMP    481
 VAR LSP: STP; LXRGS: XRGSTATUS; I: REGNR;                                 COMP    481
BEGIN SAVEREFXRGS(LXRGS); RJTOEXT(ΞP.CLOCK    Ξ);                          COMP    482
 XRGS := LXRGS;                                                            COMP    482
 IF XRGS[6].XCONT <> AVAIL THEN                                            COMP    482
  BEGIN NEEDX(0,7,I); GEN15(10B,I,6,0) END                                 COMP    482
 ELSE I := 6;                                                              COMP    482
 CLEARREGS;                                                                COMP    482
 WITH XRGS[I] DO                                                           COMP    482
  BEGIN XCONT := OTHER; REFNR := 1 END;                                    COMP    482
 RELOADREFXRGS(LXRGS);                                                     COMP    482
 WITH GATTR DO                                                             COMP    482
  BEGIN TYPTR := INTPTR;                                                   COMP    483
   KIND := EXPR; EXPREG := I                                               COMP    483
  END                                                                      COMP    483
END (*CLOCKF*) ;                                                           COMP    483
                                                                           COMP    483
PROCEDURE STDARITHFUNCS;                                                   COMP    483
 VAR LXRGS: XRGSTATUS; I, K: REGNR; LNAME: ALFA;                           COMP    483
BEGIN TEST1(LPARENT,9);                                                    COMP    483
 SAVEREFXRGS(LXRGS); CLEARREGS;                                            COMP    483
 EXPRESSION(FSYS+[RPARENT]);                                               COMP    483
 LOAD(GATTR,I);                                                            COMP    484
 IF COMPTYPES(GATTR.TYPTR,INTPTR,FALSE) THEN                               COMP    484
  BEGIN PACKANDNORM(I); GATTR.TYPTR := REALPTR END;                        COMP    484
 IF I <> 1 THEN                                                            COMP    484
  BEGIN NEEDX(1,1,K); BXIXJ(1,I) END;                                      COMP    484
 IF NOT COMPTYPES(GATTR.TYPTR,REALPTR,FALSE) THEN ERROR(125);              COMP    484
 IF LKEY = 19 THEN GEN15(66B,3,0,0) (*SIN*)                                COMP    484
 ELSE IF LKEY = 20 THEN GEN15(66B,3,1,0); (*COS*)                          COMP    484
 (*SET NEG. REL. ADDRESS TO INDICATE A CALL OF A STD ARITH. FUNC:*)        COMP    484
 GEN30(71B,6,0,-IC,0); GEN15(56B,6,2,1);                                   COMP    484
 CASE LKEY OF                                                              COMP    485
  (*SIN*) 19,                                                              COMP    485
  (*COS*)     20: LNAME := ΞP.SINCO    Ξ;                                  COMP    485
  (*EXP*)     21: LNAME := ΞP.EXP      Ξ;                                  COMP    485
  (*SQRT*)    22: LNAME := ΞP.SQRT     Ξ;                                  COMP    485
  (*LN*)      23: LNAME := ΞP.LN       Ξ;                                  COMP    485
  (*ARCTAN*)  24: LNAME := ΞP.ATAN     Ξ;                                  MSURAND
  (*RANDOM*)  25: LNAME := ΞP.RAND     Ξ                                   MSURAND
```

```
 END;                                                                    COMP    485
 RJTOEXT(LNAME);                                                         COMP    485
 (*CLEAR INDICATION WORD:*)                                              COMP    485
 GEN15(13B,7,7,7); GEN15(56B,7,2,1);                    000061           COMP    486
 XRGS := LXRGS;                                                          COMP    486
 IF XRGS[6].XCONT <> AVAIL THEN                                          COMP    486
  BEGIN NEEDX(0,7,I); GEN15(10B,I,6,0) END                               COMP    486
 ELSE I := 6;                                                            COMP    486
 CLEARREGS;                                                              COMP    486
 WITH XRGS[I] DO                                                         COMP    486
  BEGIN XCONT := OTHER; REFNR := 1 END;                                  COMP    486
 RELOADREFXRGS(LXRGS);                                                   COMP    486
 WITH GATTR DO                                                           COMP    486
  BEGIN TYPTR := REALPTR; KIND := EXPR; EXPREG := I END;                 COMP    487
 TEST1(RPARENT,4)                                                        COMP    487
END (*STDARITHFUNCS*) ;                                                  COMP    487
                                                                         COMP    487
PROCEDURE CALLNONSTANDARD;                                               COMP    487
 VAR NXT,LCP: CTP; LSP: STP; LKIND: IDKIND; LB,FTN: BOOLEAN;             COMP    487
    L,M: LEVRANGE; I,K,LXPAR: REGNR; PARAM: ATTR;                        COMP    487
    PVDISP,LDSP,LB6DPL: ADDRRANGE; LXRGS: XRGSTATUS;                     COMP    487
    LMIN,LMAX: INTEGER;                                                  COMP    487
    PASS: (VAL,VARADDR,PROCDESC);                                        COMP    487
BEGIN                                                                    COMP    488
 LB6DPL := B6DPL;                                                        COMP    488
 WITH FCP↑ DO                                                            COMP    488
  BEGIN NXT := NEXT; LKIND := PFKIND;                                    COMP    488
   FTN := FALSE;                                                         COMP    488
   IF LKIND = ACTUAL THEN FTN := PFDECL = FTNDECL;                       COMP    488
   IF KLASS = FUNC THEN                                                  COMP    488
    BEGIN SAVEREFXRGS(LXRGS);                                            COMP    488
     IF B6DPL <> 3 THEN                                                  COMP    488
      BEGIN                                                              COMP    488
       FOR I := 0 TO 7 DO                                                COMP    489
        BEGIN                                                            COMP    489
         WITH ARGS[I] DO                                                 COMP    489
          IF ACONT = SIMPADDR THEN                                       COMP    489
           IF ALEV = 0 THEN ACONT := UNSPECADDR;                         COMP    489
         WITH XRGS[I] DO                                                 COMP    489
          IF XCONT = SIMPVAR THEN                                        COMP    489
           IF XLEV = 0 THEN XCONT := AVAIL                               COMP    489
        END;                                                             COMP    489
       GEN30(61B,6,6,B6DPL,0);                                           COMP    489
       GEN30(51B,0,0,IC,0); GEN30(06B,6,4,53B,0)                         COMP    490
      END;                                                               COMP    490
    END                                                                  COMP    490
  END;                                                                   COMP    490
 LXPAR := 0; LDSP := 3; PVDISP := LC;                                    COMP    490
 IF SY = LPARENT THEN                                                    COMP    490
  BEGIN                                                                  COMP    490
   REPEAT PASS := VAL;                                                   COMP    490
    IF LKIND = ACTUAL THEN                                               COMP    490
     IF NXT = NIL THEN ERROR(126)                                        COMP    490
     ELSE                                                                COMP    491
      IF NXT↑.KLASS IN [PROC,FUNC] THEN PASS := PROCDESC;                COMP    491
    (*NOTE THAT IN THIS IMPLEMENTATION FORMAL PROC/FUNC MUST             COMP    491
    ONLY HAVE VALUE PARAMETERS*)                                         COMP    491
    INSYMBOL;                                                            COMP    491
    IF PASS = PROCDESC THEN                                              COMP    491
     BEGIN                                                               COMP    491
      IF SY <> IDENT THEN                                                COMP    491
       BEGIN ERROR(2); SKIP(FSYS+[COMMA,RPARENT]); NEEDX(0,7,I) END      COMP    491
      ELSE                                                               COMP    491
       BEGIN                                                             COMP    492
        IF NXT↑.KLASS = PROC THEN SEARCHID([PROC],LCP)                   COMP    492
        ELSE                                                             COMP    492
         BEGIN SEARCHID([FUNC],LCP);                                     COMP    492
          IF NOT COMPTYPES(LCP↑.IDTYPE,NXT↑.IDTYPE,FALSE)                COMP    492
           THEN ERROR(128)                                               COMP    492
         END;                                                            COMP    492
        IF LCP↑.PFDECKIND = STANDARD THEN                                COMP    492
         BEGIN ERROR(164); NEEDX(0,7,I) END                              COMP    492
        ELSE                                                             COMP    492
        BEGIN IF LCP↑.PFXOPT <> NXT↑.PFXOPT THEN ERROR(179);             COMP    493
         IF LCP↑.PFKIND = ACTUAL THEN                                    COMP    493
          BEGIN (*SET UP DESCRIPTOR:*)                                   COMP    493
           WITH LCP↑ DO                                                  COMP    493
            BEGIN                                                        COMP    493
             IF FTN AND (PFDECL<>FTNDECL) THEN ERROR(173)                COMP    493
             ELSE                                                        COMP    493
              IF NOT FTN AND (PFDECL=FTNDECL) THEN ERROR(174);           COMP    493
```

```
          NEEDX(6,7,I);                                                    COMP    493
          IF PFCNT = 0 THEN SEARCHEXTID(NAME)                              COMP    493
          ELSE                                                             COMP    494
            BEGIN PFNAME(PFCNT);                                           COMP    494
             SEARCHEXTID(PNAME)                                            COMP    494
            END;                                                           COMP    494
          GEN30(71B,I,0,0,0);                                              COMP    494
          IF (PFLEV <> 1)AND NOT FTN THEN (*ADD SURR. BASE ADR*)           COMP    494
            BEGIN                                                          COMP    494
             IF PFLEV IN LEVELS THEN                                       COMP    494
              BEGIN NEEDX(0,7,K);                                          COMP    494
               GEN15(76B,K,BRG[PFLEV],0)                                   COMP    494
              END                                                          COMP    495
              ELSE LOADBASE(PFLEV,TRUE,K);                                 COMP    495
              GEN15(20B,K,0,18); GEN15(12B,I,I,K);                         COMP    495
              DECREFX(K)                                                   COMP    495
            END                                                            COMP    495
         END;                                                              COMP    495
        (*MAKE SURE THAT PROC/FUNC HAS ONLY VALUE                          COMP    495
        PARAMS OF FIXED SIZE:*)                                            COMP    495
        LCP := LCP↑.NEXT;                                                  COMP    495
        WHILE LCP <> NIL DO                                                COMP    495
         WITH LCP↑ DO                                                      COMP    496
          BEGIN IF KLASS <> VARS THEN ERROR(170)                           COMP    496
           ELSE                                                            COMP    496
            IF VKIND = INDRCT THEN ERROR(170);                             COMP    496
           LCP := NEXT                                                     COMP    496
          END                                                              COMP    496
       END (*LCP↑.PFKIND = ACTUAL*)                                        COMP    496
      ELSE                                                                 COMP    496
       BEGIN (*LOAD DESCRITOR:*)                                           COMP    496
        WITH LCP↑, PARAM DO                                                COMP    496
         BEGIN TYPTR := INTPTR; KIND := VARBL;                             COMP    497
          WORDACC := DRCT; VLEVEL := PFLEV;                                COMP    497
          CWDISPL := PFADDR; PCKD := FALSE                                 COMP    497
         END;                                                              COMP    497
        LOAD(PARAM,I)                                                      COMP    497
       END;                                                                COMP    497
     END;                                                                  COMP    497
     END (*SY = IDENT*);                                                   COMP    497
    GATTR.TYPTR := INTPTR;                                                 COMP    497
    INSYMBOL;                                                              COMP    497
    TEST2(FSYS+[COMMA,RPARENT],6,[])                                       COMP    498
  END (*PASS = PROCDESC*)                                                  COMP    498
ELSE                                                                       COMP    498
  BEGIN                                                                    COMP    498
    EXPRESSION(FSYS+[COMMA,RPARENT]); LSP := NIL;                          COMP    498
    IF GATTR.TYPTR <> NIL THEN                                             COMP    498
     BEGIN                                                                 COMP    498
      IF LKIND = ACTUAL THEN                                               COMP    498
       BEGIN                                                               COMP    498
        IF NXT <> NIL THEN                                                 COMP    498
         BEGIN LSP := NXT↑.IDTYPE;                                         COMP    499
          IF LSP <> NIL THEN                                               COMP    499
           BEGIN                                                           COMP    499
             IF (NXT↑.VKIND<>DRCT)OR (FULLWORDS                            COMP    499
              (LSP↑.SIZE)<>1) THEN                                         COMP    499
              PASS := VARADDR                                              COMP    499
           END                                                             COMP    499
         END                                                               COMP    499
       END                                                                 COMP    499
      ELSE                                                                 COMP    499
       IF FULLWORDS(GATTR.TYPTR↑.SIZE) <> 1 THEN                           COMP    500
        PASS := VARADDR;                                                   COMP    500
      IF PASS = VAL THEN                                                   COMP    500
       BEGIN LOAD(GATTR,I);                                                COMP    500
        IF (LKIND = ACTUAL)AND (NXT <> NIL) THEN                           COMP    500
         IF COMPTYPES(LSP,REALPTR,FALSE) THEN                              COMP    500
          BEGIN                                                            COMP    500
            IF COMPTYPES(GATTR.TYPTR,INTPTR,FALSE) THEN                    COMP    500
             BEGIN PACKANDNORM(I);                                         COMP    500
               GATTR.TYPTR := REALPTR                                      COMP    500
             END                                                           COMP    501
          END                                                              COMP    501
         ELSE                                                              COMP    501
          IF LSP <> NIL THEN                                               COMP    501
           IF DEBUG AND (LSP<>INTPTR)AND (LSP↑.FORM<=SUBRANGE)             COMP    501
            THEN                                                           COMP    501
            BEGIN GETBOUNDS(LSP,LMIN,LMAX);                                COMP    501
             CHECKBNDS(I,LMIN,LMAX,ASSERR)                                 COMP    501
            END                                                            COMP    501
```

```
        END
       ELSE
        BEGIN
         IF (LKIND = ACTUAL)AND (NXT <> NIL) THEN
          IF NXT↑.VKIND = INDRCT THEN
           IF GATTR.KIND <> VARBL THEN
            BEGIN ERROR(142); GATTR.TYPTR := NIL END
           ELSE
            IF GATTR.PCKD THEN ERROR(127);
         LOADADDRESS(GATTR,I)
        END;
       IF (LKIND = ACTUAL)AND (NXT <> NIL) THEN
        IF NOT COMPTYPES(NXT↑.IDTYPE,GATTR.TYPTR,FALSE) THEN
         ERROR(142)
      END (*GATTR.TYPTR <> NIL*)
     ELSE NEEDX(0,7,I);
    END (*PASS <> PROCDESC*) ;
   IF (LXPAR < FCP↑.PFXOPT) AND NOT FTN THEN
    BEGIN
     IF I <> LXPAR THEN
      BEGIN NEEDX(LXPAR,LXPAR,LXPAR); BXIXJ(LXPAR,I);
      END;
     LXPAR := LXPAR + 1
    END
   ELSE
    BEGIN
     WITH PARAM DO
      BEGIN TYPTR := GATTR.TYPTR; KIND := VARBL;
       WORDACC := DRCT; PCKD := FALSE;
       IF FTN AND (PASS IN [VARADDR,PROCDESC]) THEN
        BEGIN VLEVEL := LEVEL; CWDISPL := LC END
       ELSE
        BEGIN VLEVEL := 0; CWDISPL := LDSP END;
      END;
     STORE(PARAM,I);
     IF FTN THEN
      BEGIN
       IF PASS = VAL THEN
        BEGIN NEEDX(6,7,I); GEN30(71B,I,6,LDSP,0);
         WITH PARAM DO
          BEGIN VLEVEL := LEVEL; CWDISPL := LC END;
         STORE(PARAM,I)
        END;
       LC := LC + 1
      END;
     B6DPL := LDSP + 1
    END;
   LDSP := LDSP + 1;
   IF (LKIND = ACTUAL)AND (NXT <> NIL) THEN NXT := NXT↑.NEXT
  UNTIL SY <> COMMA;
  TEST1(RPARENT,4)
 END (*IF LPARENT*);
 FOR I := 0 TO LXPAR - 1 DO ROTATEX(I);
 IF LKIND = ACTUAL THEN
  IF FTN THEN
   BEGIN CLEARREGS;
    (*SET SENTINEL FOR PARAMETER VECTOR:*)
    GEN15(13B,6,6,6); GEN30(51B,6,5,LC,0);
    LC := LC + 1;
    IF LC > LCMAX THEN LCMAX := LC;
    (*SAVE B4/B5/B6:*)
    GEN15(76B,6,4,0);
    FOR I := 5 TO 6 DO
     BEGIN GEN15(20B,6,0,18); GEN15(76B,0,I,0); GEN15(12B,6,6,0)
     END;
    GEN15(56B,6,2,0);
    (*SET FTN INDICATOR FOR PMD:*)
    GEN30(71B,6,0,IC,0); GEN15(56B,6,2,1);
    (*LOAD 1. WORD OF PARAMETER VECTOR AND RJ:*)
    GEN30(51B,1,5,PVDISP,0);
    RJTOEXT(FCP↑.NAME);
    (*RELOAD B-REGISTERS:*)
    GEN30(61B,1,0,1,0); GEN30(51B,1,0,65B,0);
    GEN15(63B,2,1,0); GEN15(56B,1,2,0);
    FOR I := 6 DOWNTO 5 DO
     BEGIN GEN15(63B,I,1,0); GEN15(21B,1,0,18) END;
    GEN15(63B,4,1,0);
    (*CLEAR FTN INDICATOR:*)
    GEN15(13B,7,7,7); GEN15(56B,7,2,1)
   END
  ELSE
```

```
    BEGIN IF NXT <> NIL THEN ERROR(126);                                    COMP  510
     WITH FCP↑ DO                                                           COMP  510
      BEGIN                                                                 COMP  510
       IF PFLEV <> 1 THEN (*LOAD STATIC LINK INTO X6*)                      COMP  510
        IF PFLEV IN LEVELS THEN GEN15(76B,6,BRG[PFLEV],0)                   COMP  510
        ELSE                                                                COMP  510
         BEGIN LOADBASE(PFLEV,FALSE,I);                                     COMP  510
          GEN15(10B,6,I,0); DECREFX(I)                                      COMP  510
         END;                                                               COMP  510
       GEN30(71B,7,0,IC+1,2);                                               COMP  510
        IF PFCNT = 0 THEN (*TAKE USER DECLARED NAME*)                       COMP  511
         EQTOEXT(NAME)                                                      COMP  511
        ELSE                                                                COMP  511
         BEGIN PFNAME(PFCNT);                                               COMP  511
          EQTOEXT(PNAME)                                                    COMP  511
         END;                                                               COMP  511
      END                                                                   COMP  511
    END                                                                     COMP  511
   ELSE                                                                     COMP  511
    BEGIN                                                                   COMP  511
     WITH FCP↑, PARAM DO                                                    COMP  512
      BEGIN TYPTR := INTPTR; KIND := VARBL; WORDACC := DRCT;                COMP  512
       VLEVEL := PFLEV; CWDISPL := PFADDR; PCKD := FALSE                    COMP  512
      END;                                                                  COMP  512
     LOAD(PARAM,I);                                                         COMP  512
     IF I IN [6,7] THEN                                                     COMP  512
      BEGIN K := I; NEEDX(0,5,I); BXIXJ(I,K) END;                           COMP  512
     GEN15(63B,7,I,0); GEN15(73B,6,I,0); GEN15(20B,6,0,36);                 COMP  512
     LDSP := IC + 1;                                                        COMP  512
     IF PC.CP <> 1 THEN LDSP := LDSP + 1;                                   COMP  512
     GEN30(71B,7,0,LDSP,2); GEN15(12B,7,6,7);                               COMP  513
     GEN15(20B,I,0,42); GEN15(73B,6,I,0);                                   COMP  513
     GEN30(02B,7,0,0,0); NOOP                                               COMP  513
    END;                                                                    COMP  513
   CLEARREGS;                                                               COMP  513
   LC := PVDISP;                                                            COMP  513
   B6DPL := LB6DPL;                                                         COMP  513
    IF FCP↑.KLASS = FUNC THEN                                               COMP  513
     BEGIN                                                                  COMP  513
      IF B6DPL <> 3 THEN GEN30(61B,6,6,-B6DPL,0);                           COMP  513
      XRGS := LXRGS;                                                        COMP  514
      IF XRGS[6].XCONT <> AVAIL THEN                                        COMP  514
       BEGIN NEEDX(0,7,I); GEN15(10B,I,6,0) END                             COMP  514
      ELSE I := 6;                                                          COMP  514
      CLEARREGS;                                                            COMP  514
      WITH XRGS[I] DO                                                       COMP  514
       BEGIN XCONT := OTHER; REFNR := 1 END;                                COMP  514
      RELOADREFXRGS(LXRGS);                                                 COMP  514
      WITH GATTR DO                                                         COMP  514
       BEGIN TYPTR := FCP↑.IDTYPE; KIND := EXPR;                            COMP  514
        EXPREG := I                                                         COMP  515
       END                                                                  COMP  515
     END                                                                    COMP  515
  END (*CALLNONSTANDARD*) ;                                                 COMP  515

BEGIN (*CALL*)                                                              COMP  515
 IF FCP↑.PFDECKIND = STANDARD THEN                                          COMP  515
  BEGIN                                                                     COMP  515
   LKEY := FCP↑.KEY;                                                        COMP  515
   IF FCP↑.KLASS = PROC THEN                                                COMP  515
    BEGIN                                                                   COMP  516
     CASE LKEY OF                                                           COMP  516
       1,2,    (*GET,PUT*)                                                  COMP  516
       3,4,    (*RESET,REWRITE*)                                            COMP  516
       5,6:    (*GETSEG,PUTSEG*)                                            COMP  516
               STDFLPROCS;                                                  COMP  516
         7:    LINELIMIT;                                                   COMP  516
       8,9:    (*READ,READLN*)                                              COMP  516
               READ;                                                        COMP  516
     10,11:    (*WRITE,WRITELN*)                                            COMP  516
               WRITE;                                                       COMP  517
        12:    MESSAGE;                                                     COMP  517
        13:    PAGE;                                                        COMP  517
     14,15:    TIMEDATE;                                                    COMP  517
        16:    HALT;                                                        COMP  517
        17:    PACK;                                                        COMP  517
        18:    UNPACK;                                                      COMP  517
     19,20:    NEWDISPOSE;                                                  COMP  517
        21:    RELEASE;                                                     COMP  517
     22,23:    (*DUMMY*)                                                    COMP  517
     END                                                                    COMP  518
```

```
    END                                                                  COMP      518
  ELSE                                                                   COMP      518
    BEGIN                                                                COMP      518
     CASE LKEY OF                                                        COMP      518
      1,2,    (*EOF,EOS*)                          (*PREDICATES*)        COMP      518
      3:      (*EOLN*)                                                   COMP      518
              STDFLFUNCS;                                                COMP      518
       4,      (* ODD *)                                                 COMP      518
       5,      (* UNDEFINED *)                                           COMP      518
       6,      (* ROUND *)                                               COMP      519
       7,      (* TRUNC *)                                               COMP      519
       8,      (* EXPO *)                                                COMP      519
       9,      (* ABS *)                                                 COMP      519
       10,     (* SQR *)                                                 COMP      519
       11,     (* ORD *)                                                 COMP      519
       12,     (* CHR *)                                                 COMP      519
       13,     (* PRED *)                                                COMP      519
       14,     (* SUCC *)                                                COMP      519
       15:     (* CARD *)                                                COMP      519
               STDINLINEFUNCS;                                            COMP      520
      16:     CLOCKF;                                                     COMP      520
      17,18: (*DUMMY*);                                                   COMP      520
      19,20,                                                              COMP      520
      21,22;                                                              COMP      520
      23,24,25: STDARITHFUNCS                                             MSURAND
     END                                                                  COMP      520
    END;                                                                  COMP      520
  END (*STANDARD PROCEDURES AND FUNCTIONS*)                               COMP      520
 ELSE CALLNONSTANDARD                                                     COMP      520
END (*CALL*) ;                                                            COMP      521
                                                                          COMP      521
PROCEDURE EXPRESSION;                                                     COMP      521
 VAR LATTR: ATTR; LOP: OPERATOR; WRDS,LADDR: ADDRRANGE;                   COMP      521
     BTS : BITRANGE; I,J,K,L,M,N,R,II,JJ : REGNR;                         COMP      521
   LOPCD: OPRANGE; FLSJMP: 0..3;                                          COMP      521
   LVENTOUT:BOOLEAN; LOW,HIGH:INTEGER;                                    COMP      521
                                                                          COMP      521
 PROCEDURE SIMPLEEXPRESSION(FSYS: SETOFSYS);                              COMP      521
  VAR LATTR: ATTR; LOP: OPERATOR; SIGNED: BOOLEAN;                        COMP      521
    LOPD: 0..1;                                                           COMP      522
    I,J,K: REGNR;                                                         COMP      522
                                                                          COMP      522
 PROCEDURE BOOLOP(VAR FATTR: ATTR; FOP: OPERATOR);                        COMP      522
  (*GENERATE CODE FOR <FATTR> <FOP> <GATTR> (FOP IN [ANDOP,OROP])*)       COMP      522
  (*RESULTING ATTRIBUTES IN GATTR*)                                       COMP      522
  VAR I,J,K,L: REGNR; IND1,IND2,SHFT,OPCD: INTEGER;                       COMP      522
                                                                          COMP      522
  PROCEDURE LDOPD(VAR FATTR: ATTR; VAR FI: REGNR;VAR FIND: INTEGER);      COMP      522
   (*LOAD OPERAND DESCRIBED BY FATTR INTO X-FI AND SET INDICATOR          COMP      522
   FIND TO DISTINGUISH BETWEEN 5 CASES:                                   COMP      523
      VALUE OF FIND:     0      1      2      3      4                    COMP      523
      X-FI CONTAINS:  LOGICAL ZR     NZ     PL     NG*)                  COMP      523
   VAR I: REGNR;                                                          COMP      523
   BEGIN                                                                  COMP      523
    WITH FATTR DO                                                         COMP      523
     IF KIND = COND THEN                                                  COMP      523
      BEGIN                                                               COMP      523
       IF CONDCD IN [ZR,NZ] THEN                                          COMP      523
        BEGIN NEEDX(0,7,I); GEN15(13B,I,I,I); GEN15(37B,I,I,CDR);         COMP      523
         IF CONDCD = ZR THEN GEN15(13B,I,CDR,I)                           COMP      524
         ELSE GEN15(17B,I,CDR,I);                                         COMP      524
         DECREFX(CDR); FI := I                                            COMP      524
        END                                                               COMP      524
       ELSE FI := CDR;                                                    COMP      524
       FIND := ORD(CONDCD) + 1                                            COMP      524
      END                                                                 COMP      524
     ELSE                                                                 COMP      524
      BEGIN LOAD(FATTR,FI); FIND := 0 END                                 COMP      524
  END (*LDOPD*) ;                                                         COMP      524
                                                                          COMP      525
 BEGIN (*BOOLOP*) LDOPD(FATTR,I,IND1); LDOPD(GATTR,J,IND2);               COMP      525
  IF IND2 < IND1 THEN (*TRANSPOSE OPS*)                                   COMP      525
   BEGIN K := I; I := J; J := K;                                          COMP      525
    K := IND1; IND1 := IND2; IND2 := K                                    COMP      525
   END;                                                                   COMP      525
  IF (IND1=0)AND (IND2 IN [3,4]) THEN                                     COMP      525
   BEGIN                                                                  COMP      525
    IF FOP =ANDOP THEN                                                    COMP      525
     BEGIN K := J; SHFT := 1 END                                          COMP      525
    ELSE                                                                  COMP      526
     BEGIN K := I; SHFT := 59 END;                                        COMP      526
```

```
      WITH XRGS[K] DO
       IF REFNR = 1 THEN
        BEGIN GEN15(20B,K,0,SHFT); XCONT := OTHER END
       ELSE
        BEGIN NEEDX(0,7,L); GEN15(10B,L,K,0);
         GEN15(20B,L,0,SHFT); DECREFX(K);
         IF FOP = ANDOP THEN J := L ELSE I := L
        END
     END;
    NEEDX(0,7,K);
    (*SET RESULT ATTRIBUTES:*)
    WITH GATTR DO
     BEGIN TYPTR := BOOLPTR;
      IF IND1 = 0 THEN
       BEGIN KIND := EXPR; EXPREG := K END
      ELSE
       BEGIN KIND := COND; CONDCD := PL; CDR := K END
     END;
    IF FOP = ANDOP THEN OPCD := 11B ELSE OPCD := 12B;
    IF IND2 = 4 THEN
     IF IND1 <> 4 THEN OPCD := OPCD + 4
     ELSE
      BEGIN OPCD := 23B - OPCD; GATTR.CONDCD := NG END;
    GEN15(OPCD,K,I,J);
    IF (FOP = OROP)AND (IND1 = 0)AND (IND2 <> 0) THEN
     BEGIN
      WITH GATTR DO
       BEGIN KIND := COND; CONDCD := PL; CDR := K END;
      IF IND2 IN [1,2] THEN GEN15(20B,K,0,59)
     END;
    DECREFX(I); DECREFX(J)
   END (*BOOLOP*) ;

    PROCEDURE TERM(FSYS: SETOFSYS);
     VAR LATTR: ATTR; LOP: OPERATOR;
       I,J,K,L: REGNR; LREC: CSTREC;

     PROCEDURE LOADANDCHECKX(VAR FI: REGNR);
      (*LOAD GATTR INTO X-FI, TEST IT AGAINST ZERO*)
     BEGIN
      IF GATTR.KIND = CST THEN
       BEGIN IF GATTR.CVAL.IVAL = 0 THEN ERROR(300);
        LOAD(GATTR,FI)
       END
      ELSE
       BEGIN LOAD(GATTR,FI);
        IF DEBUG THEN
         BEGIN GEN30(51B,0,0,IC,0);
          GEN30(03B,0,FI,DIVERR,0)
         END
       END
     END (*LOADANDCHECKX*) ;

     PROCEDURE FACTOR(FSYS: SETOFSYS);
      VAR LCP: CTP; LSP: STP; I,J,K,L,M: REGNR; LCSTATTR,LATTR: ATTR;
        VARPART,EXITLOOP: BOOLEAN; N: INTEGER;
     BEGIN
      IF NOT (SY IN FACBEGSYS) THEN
       BEGIN ERROR(58); SKIP(FSYS+FACBEGSYS);
        GATTR.TYPTR := NIL
       END;
      REPEAT
       IF SY IN FACBEGSYS THEN
       BEGIN
        CASE SY OF
(*ID*)   IDENT:
          BEGIN SEARCHID([KONST,VARS,FIELD,FUNC],LCP);
           INSYMBOL;
           CASE LCP^.KLASS OF
            KONST:
             WITH LCP^, GATTR DO
              BEGIN TYPTR := IDTYPE; KIND := CST;
               CVAL := VALUES
              END;
            VARS,
            FIELD:
             SELECTOR(FSYS,LCP);
            FUNC:
             CALL(FSYS,LCP)
           END
          END;
```

```
(*CST*)    INTCONST:
            BEGIN
             WITH GATTR DO
              BEGIN TYPTR := INTPTR; KIND := CST;
               CVAL.IVAL := IVAL
              END;
             INSYMBOL
            END;
           REALCONST:
            BEGIN
             WITH GATTR DO
              BEGIN TYPTR := REALPTR; KIND := CST;
               CVAL.RVAL := RVAL
              END;
             INSYMBOL
            END;
           CHARCONST:
            BEGIN
             WITH GATTR DO
              BEGIN TYPTR := CHARPTR; KIND := CST;
               CVAL.IVAL := IVAL
              END;
             INSYMBOL
            END;
           STRINGCONST:
            BEGIN
             WITH GATTR DO
              BEGIN STRINGTYPE(TYPTR); KIND := CST;
               CVAL.VALP := CONSTP
              END;
             INSYMBOL
            END;
 (*(*)     LPARENT:
            BEGIN INSYMBOL; EXPRESSION(FSYS+[RPARENT]);
             TEST1(RPARENT,4)
            END;
(*NOT*)   NOTSY:
            BEGIN INSYMBOL; FACTOR(FSYS);
             IF COMPTYPES(GATTR.TYPTR,BOOLPTR,FALSE) AND
               (GATTR.TYPTR <> NIL) THEN
              WITH GATTR DO
               IF KIND = COND THEN
                CASE CONDCD OF
                 ZR: CONDCD := NZ;
                 NZ: CONDCD := ZR;
                 PL: CONDCD := NG;
                 NG: CONDCD := PL
                END
               ELSE
                BEGIN LOAD(GATTR,I); NEEDX(0,7,K);
                 GEN15(43B,K,0,59); GEN15(17B,K,I,K); DECREFX(I);
                 GATTR.EXPREG := K
                END
             ELSE
              BEGIN ERROR(135); GATTR.TYPTR := NIL END
            END;
(*[*)      LBRACK:
            BEGIN INSYMBOL;
             NEW(LSP,POWER);
             WITH LSP↑ DO
              BEGIN ELSET := NIL; FORM := POWER; PCKDSET := FALSE;
               FTYPE := FALSE;
               WITH SIZE DO
                BEGIN WORDS := 1; BITS := 0 END
              END;
             VARPART := FALSE;
             WITH LCSTATTR DO
              BEGIN TYPTR := LSP; KIND := CST; CVAL.PVAL := [ ]
              END;
             IF SY = RBRACK THEN INSYMBOL
             ELSE
              BEGIN
               (*LOOP UNTIL SY <> COMMA:*)
               REPEAT EXPRESSION(FSYS+[COMMA,COLON,RBRACK]);
                IF GATTR.TYPTR <> NIL THEN
                 IF GATTR.TYPTR↑.FORM > SUBRANGE THEN
                  BEGIN ERROR(136); GATTR.TYPTR := NIL END
                 ELSE
                  IF NOT COMPTYPES(LSP↑.ELSET,GATTR.TYPTR,FALSE)
                   THEN ERROR(137);
                IF SY = COLON THEN
```

```
      BEGIN LATTR := GATTR; INSYMBOL;                                    COMP    542
       EXPRESSION(FSYS+[COMMA,RBRACK]);                                  COMP    542
       IF GATTR.TYPTR <> NIL THEN                                        COMP    542
        IF GATTR.TYPTR↑.FORM > SUBRANGE THEN          000068              COMP    542
         BEGIN ERROR(136); GATTR.TYPTR := NIL                            COMP    542
         END                                                             COMP    542
        ELSE                                                             COMP    543
         IF NOT COMPTYPES(LATTR.TYPTR,GATTR.TYPTR,                       COMP    543
          FALSE) THEN ERROR(137);                                        COMP    543
       IF (LATTR.TYPTR <> NIL)AND (GATTR.TYPTR <> NIL)                   COMP    543
        THEN                                                             COMP    543
        BEGIN                                                            COMP    543
         IF (LATTR.KIND = CST)AND (GATTR.KIND = CST)                     COMP    543
          THEN                                                           COMP    543
          BEGIN                                                          COMP    543
           IF (LATTR.CVAL.IVAL<0)OR (GATTR.CVAL                          COMP    543
           .IVAL>58) THEN ERROR(304);                                    COMP    544
            FOR N := LATTR.CVAL.IVAL TO GATTR                            COMP    544
             .CVAL.IVAL DO                                               COMP    544
             LCSTATTR.CVAL.PVAL := LCSTATTR                              COMP    544
              .CVAL.PVAL+[N]                                             COMP    544
          END                                                            COMP    544
         ELSE                                                            COMP    544
          BEGIN LOAD(LATTR,I);                                           COMP    544
           IF DEBUG THEN CHECKBNDS(I,0,58,                               COMP    544
            ASSERR);                                                     COMP    544
           LOAD(GATTR,J);                                                COMP    545
           IF DEBUG THEN CHECKBNDS(J,0,58,                               COMP    545
            ASSERR);                                                     COMP    545
           NEEDX(0,7,K); GEN15(37B,K,J,I);                               COMP    545
           DECREFX(I); NEEDX(0,7,I);                                     COMP    545
           GEN15(13B,I,I,I);                                             COMP    545
           IF PC.CP = 1 THEN                                             COMP    545
            GEN30(03B,3,K,IC+1,2)                                        COMP    545
           ELSE GEN30(03B,3,K,IC+2,2);                                   COMP    545
           NEEDB(L); GEN15(63B,L,K,0);                                   COMP    545
           GEN15(43B,K,0,1); GEN15(23B,K,L,K);                           COMP    546
           GEN15(63B,L,J,1); DECREFX(J);                                 COMP    546
           GEN15(22B,I,L,K); DECREFX(K);                                 COMP    546
           BRGS[L].BCONT := FREE; NOOP;                                  COMP    546
           IF VARPART THEN                                               COMP    546
            BEGIN GEN15(12B,M,I,M); DECREFX(I)                           COMP    546
            END                                                          COMP    546
           ELSE                                                          COMP    546
            BEGIN M := I; VARPART := TRUE END                            COMP    546
          END                                                            COMP    546
        END                                                              COMP    547
      END (*COLON*)                                                      COMP    547
     ELSE                                                                COMP    547
      IF GATTR.TYPTR <> NIL THEN                                         COMP    547
       IF GATTR.KIND = CST THEN                                          COMP    547
        BEGIN                                                            COMP    547
         IF (GATTR.CVAL.IVAL<0)OR (GATTR.CVAL.IVAL                       COMP    547
          >58) THEN ERROR(304);                                          COMP    547
         LCSTATTR.CVAL.PVAL := LCSTATTR.CVAL.PVAL                        COMP    547
         +[GATTR.CVAL.IVAL]                                              COMP    547
        END                                                              COMP    548
       ELSE                                                              COMP    548
        BEGIN LOAD(GATTR,I);                                             COMP    548
         IF DEBUG THEN CHECKBNDS(I,0,58,ASSERR);                         COMP    548
         NEEDB(J);                                                       COMP    548
         GEN15(63B,J,I,0); DECREFX(I);                                   COMP    548
         NEEDX(0,7,I); GEN15(76B,I,1,0);                                 COMP    548
         GEN15(22B,I,J,I); BRGS[J].BCONT := FREE;                        COMP    548
         IF VARPART THEN                                                 COMP    548
          BEGIN GEN15(12B,M,I,M); DECREFX(I) END                         COMP    548
         ELSE                                                            COMP    549
          BEGIN M := I; VARPART := TRUE END                              COMP    549
        END;                                                             COMP    549
      IF GATTR.TYPTR <> NIL THEN                                         COMP    549
       BEGIN IF COMPTYPES(GATTR.TYPTR,REALPTR,FALSE)                     COMP    549
        THEN ERROR(109);                                                 COMP    549
       LSP↑.ELSET := GATTR.TYPTR                                         COMP    549
      END;                                                               COMP    549
     EXITLOOP := SY <> COMMA;                                            COMP    549
     IF NOT EXITLOOP THEN INSYMBOL                                       COMP    549
    UNTIL EXITLOOP;                                                      COMP    550
     TEST1(RBRACK,12)                                                    COMP    550
    END;                                                                 COMP    550
   IF VARPART THEN                                                       COMP    550
    BEGIN                                                                COMP    550
```

```
                IF LCSTATTR.CVAL.PVAL <> [ ] THEN                                  COMP    550
                 BEGIN LOAD(LCSTATTR,I); GEN15(12B,M,I,M);                         COMP    550
                  DECREFX(I)                                                       COMP    550
                 END;                                                              COMP    550
                WITH GATTR DO                                                      COMP    550
                 BEGIN TYPTR := LSP; KIND := EXPR; EXPREG := M                     COMP    550
                 END                                                               COMP    551
               END                                                                 COMP    551
              ELSE GATTR := LCSTATTR                                               COMP    551
             END                                                                   COMP    551
           END (*CASE*) ;                                                          COMP    551
           TEST2(FSYS,6,FACBEGSYS)                                                 COMP    551
         END (*IF*)                                                                COMP    551
        UNTIL SY IN FSYS                                                           COMP    551
       END (*FACTOR*) ;                                                            COMP    551
                                                                                   COMP    552
      BEGIN (*TERM*)                                                               COMP    552
       FACTOR(FSYS+[MULOP]);                                                       COMP    552
       WHILE SY = MULOP DO                                                         COMP    552
        BEGIN LATTR := GATTR; LOP := OP;                                           COMP    552
         INSYMBOL; FACTOR(FSYS+[MULOP]); K := 0;                                   COMP    552
         IF (LATTR.TYPTR <> NIL)AND (GATTR.TYPTR <> NIL) THEN                      COMP    552
          CASE LOP OF                                                              COMP    552
(***)        MUL:  IF COMPTYPES(LATTR.TYPTR,INTPTR,FALSE)AND                       COMP    552
                   COMPTYPES(GATTR.TYPTR,INTPTR,FALSE) THEN                        COMP    552
                  BEGIN                                                            COMP    553
                   IF LATTR.KIND = CST THEN                                        COMP    553
                    EXPREP(LATTR.CVAL.IVAL,LREC)                                   COMP    553
                   ELSE LREC.CKIND := NOP;                                         COMP    553
                   IF LREC.CKIND = NOP THEN                                        COMP    553
                    BEGIN LOAD(LATTR,I);                                           COMP    553
                     IF GATTR.KIND = CST THEN                                      COMP    553
                      EXPREP(GATTR.CVAL.IVAL,LREC)                                 COMP    553
                    END                                                            COMP    553
                   ELSE LOAD(GATTR,I);                                             COMP    553
                   IF LREC.CKIND = NOP THEN                                        COMP    553
                    BEGIN LOAD(GATTR,J); OPERATION(42B,K,I,J);                     COMP    554
                     NEEDX(0,7,I); GEN15(13B,I,I,I);                               COMP    554
                     GEN15(36B,K,K,I); DECREFX(I)                                  COMP    554
                    END                                                            COMP    554
                   ELSE                                                            COMP    554
                    OPTMULT(I,LREC,NOT (XRGS[I].XCONT IN [SIMPVAR,                 COMP    554
                    INDVAR]),K)                                                    COMP    554
                  END                                                              COMP    554
                 ELSE                                                              COMP    554
                  BEGIN LOAD(LATTR,I); LOAD(GATTR,J);                              COMP    555
                   IF COMPTYPES(LATTR.TYPTR,INTPTR,FALSE) THEN                     COMP    555
                    BEGIN PACKANDNORM(I);                                          COMP    555
                     LATTR.TYPTR := REALPTR                                        COMP    555
                    END                                                            COMP    555
                   ELSE                                                            COMP    555
                    IF COMPTYPES(GATTR.TYPTR,INTPTR,FALSE) THEN                    COMP    555
                     BEGIN PACKANDNORM(J);                                         COMP    555
                      GATTR.TYPTR := REALPTR                                       COMP    555
                     END;                                                          COMP    555
                   IF COMPTYPES(LATTR.TYPTR,REALPTR,FALSE)                         COMP    556
                    AND COMPTYPES(GATTR.TYPTR,REALPTR,FALSE) THEN                  COMP    556
                    OPERATION(41B,K,I,J)                                           COMP    556
                   ELSE                                                            COMP    556
                    IF (LATTR.TYPTR↑.FORM = POWER)                                 COMP    556
                     AND COMPTYPES(LATTR.TYPTR,GATTR.TYPTR,FALSE)                  COMP    556
                     THEN                                                          COMP    556
                     OPERATION(11B,K,I,J)                                          COMP    556
                    ELSE BEGIN ERROR(134); GATTR.TYPTR := NIL END                  COMP    556
                  END;                                                             COMP    556
(*/*)        RDIV: BEGIN LOAD(LATTR,I); LOAD(GATTR,J);                           COMP    557
                   IF COMPTYPES(LATTR.TYPTR,INTPTR,FALSE) THEN                     COMP    557
                    BEGIN PACKANDNORM(I);                                          COMP    557
                     LATTR.TYPTR := REALPTR                                        COMP    557
                    END;                                                           COMP    557
                   IF COMPTYPES(GATTR.TYPTR,INTPTR,FALSE) THEN                     COMP    557
                    BEGIN PACKANDNORM(J);                                          COMP    557
                     GATTR.TYPTR := REALPTR                                        COMP    557
                    END;                                                           COMP    557
                   IF COMPTYPES(LATTR.TYPTR,REALPTR,FALSE)                         COMP    557
                    AND COMPTYPES(GATTR.TYPTR,REALPTR,FALSE) THEN                  COMP    558
                    OPERATION(45B,K,I,J)                                           COMP    558
                   ELSE BEGIN ERROR(134); GATTR.TYPTR := NIL END                   COMP    558
                  END;                                                             COMP    558
(*DIV*)      IDIV: IF COMPTYPES(LATTR.TYPTR,INTPTR,FALSE)                          COMP    558
                   AND COMPTYPES(GATTR.TYPTR,INTPTR,FALSE) THEN                    COMP    558
```

000069

```
            BEGIN LOAD(LATTR,I);
             IF GATTR.KIND = CST THEN
              EXPREP(GATTR.CVAL.IVAL,LREC)
             ELSE LREC.CKIND := NOP;
             IF LREC.CKIND = PUREP THEN
              IF LREC.EXP <> 0 THEN
               BEGIN NEEDX(0,7,K); GEN15(10B,K,I,0);
                DECREFX(I); GEN15(21B,K,0,LREC.EXP);
                NEEDX(0,7,I); GEN15(13B,I,I,I);
                GEN15(36B,K,K,I); DECREFX(I)
               END
              ELSE K := I
             ELSE
              BEGIN LOADANDCHECKX(J);
               IF DEBUG THEN PACKOFL(I); DECREFX(I);
               NEEDX(0,7,K); GEN15(27B,K,0,I);
               PACKANDNORM(J); GEN15(44B,K,K,J); NEEDB(I);
               GEN15(26B,K,I,K); GEN15(22B,K,I,K);
               BRGS[I].BCONT := FREE; GEN15(13B,J,J,J);
               GEN15(36B,K,K,J); DECREFX(J)
              END;
             GATTR.TYPTR := INTPTR
            END
           ELSE BEGIN ERROR(134); GATTR.TYPTR := NIL END;
(*MOD*)   IMOD: IF COMPTYPES(LATTR.TYPTR,INTPTR,FALSE)
            AND COMPTYPES(GATTR.TYPTR,INTPTR,FALSE) THEN
             BEGIN LOAD(LATTR,I);
              IF GATTR.KIND = CST THEN
               EXPREP(GATTR.CVAL.IVAL,LREC)
              ELSE LREC.CKIND := NOP;
              IF LREC.CKIND = PUREP THEN
               BEGIN NEEDX(0,7,K); GEN15(10B,K,I,0);
                GEN15(21B,K,0,LREC.EXP);
                GEN15(20B,K,0,LREC.EXP);
                GEN15(37B,K,I,K); DECREFX(I)
               END
              ELSE
               BEGIN LOADANDCHECKX(J);
                IF DEBUG THEN PACKOFL(I);
                NEEDX(0,7,K); NEEDX(0,7,L);
                GEN15(27B,L,0,I); GEN15(27B,K,0,J);
                GEN15(24B,K,0,K); GEN15(44B,K,L,K);
                DECREFX(L); NEEDB(L); GEN15(26B,K,L,K);
                GEN15(22B,K,L,K); BRGS[L].BCONT := FREE;
                IF LREC.CKIND <> NOP THEN
                 OPTMULT(K,LREC,TRUE,K)
                ELSE GEN15(42B,K,J,K);
                DECREFX(J); GEN15(37B,K,I,K);
                DECREFX(I)
               END;
             GATTR.TYPTR := INTPTR
            END
           ELSE BEGIN ERROR(134); GATTR.TYPTR := NIL END;
(*AND*)   ANDOP:IF COMPTYPES(LATTR.TYPTR,BOOLPTR,FALSE)AND
            COMPTYPES(GATTR.TYPTR,BOOLPTR,FALSE) THEN
            BOOLOP(LATTR,ANDOP)
           ELSE BEGIN ERROR(134); GATTR.TYPTR := NIL END
       END (*CASE*)
      ELSE GATTR.TYPTR := NIL;
      IF LOP <> ANDOP THEN
       WITH GATTR DO
        BEGIN KIND := EXPR; EXPREG := K END;
     END (*WHILE*)
   END (*TERM*) ;

  BEGIN (*SIMPLEEXPRESSION*)
   SIGNED := FALSE;
   IF OP IN [PLUS,MINUS] THEN
    BEGIN SIGNED := OP = MINUS; INSYMBOL END;
   TERM(FSYS+[ADDOP]);
   IF SIGNED THEN
    IF COMPTYPES(GATTR.TYPTR,INTPTR,FALSE)OR COMPTYPES(GATTR.TYPTR,
     REALPTR,FALSE) THEN
     BEGIN LOAD(GATTR,I); NEEDX(0,7,K); GEN15(13B,K,K,K);
      GEN15(37B,K,K,I); DECREFX(I); GATTR.EXPREG := K
     END
    ELSE
     BEGIN ERROR(134); GATTR.TYPTR := NIL END;
   WHILE SY = ADDOP DO
    BEGIN LATTR := GATTR; LOP := OP;
     INSYMBOL; TERM(FSYS+[ADDOP]); K := 0;
```

```
    IF (LATTR.TYPTR <> NIL)AND (GATTR.TYPTR <> NIL) THEN                    COMP      566
     CASE LOP OF                                                            COMP      566
  (*+*)PLUS,                                                                COMP      566
  (*-*)MINUS:                                                               COMP      567
       BEGIN LOAD(LATTR,I); LOAD(GATTR,J);                                  COMP      567
       LOPD := ORD(LOP = MINUS);                                            COMP      567
       IF COMPTYPES(LATTR.TYPTR,INTPTR,FALSE)                    000071     COMP      567
        AND COMPTYPES(GATTR.TYPTR,INTPTR,FALSE) THEN                        COMP      567
        OPERATION(36B+LOPD,K,I,J)                                           COMP      567
       ELSE                                                                 COMP      567
        BEGIN                                                               COMP      567
         IF COMPTYPES(LATTR.TYPTR,INTPTR,FALSE) THEN                        COMP      567
          BEGIN PACKANDNORM(I);                                             COMP      567
           LATTR.TYPTR := REALPTR                                           COMP      568
          END                                                               COMP      568
         ELSE                                                               COMP      568
          IF COMPTYPES(GATTR.TYPTR,INTPTR,FALSE) THEN                       COMP      568
          BEGIN PACKANDNORM(J);                                             COMP      568
            GATTR.TYPTR := REALPTR                                          COMP      568
          END;                                                              COMP      568
         IF COMPTYPES(LATTR.TYPTR,REALPTR,FALSE)                            COMP      568
          AND COMPTYPES(GATTR.TYPTR,REALPTR,FALSE) THEN                     COMP      568
          BEGIN OPERATION(34B+LOPD,K,I,J); GEN15(24B,K,0,K) END             COMP      568
         ELSE                                                               COMP      569
          IF (LATTR.TYPTR↑.FORM = POWER)                                    COMP      569
           AND COMPTYPES(LATTR.TYPTR,GATTR.TYPTR,FALSE) THEN                COMP      569
           OPERATION(12B+3*LOPD,K,I,J)                                      COMP      569
          ELSE BEGIN ERROR(134); GATTR.TYPTR := NIL END                     COMP      569
        END;                                                                COMP      569
       END;                                                                 COMP      569
  (*OR*)OROP:                                                               COMP      569
       IF COMPTYPES(LATTR.TYPTR,BOOLPTR,FALSE)AND                           COMP      569
        COMPTYPES(GATTR.TYPTR,BOOLPTR,FALSE) THEN                           COMP      569
        BOOLOP(LATTR,OROP)                                                  COMP      570
       ELSE BEGIN ERROR(134); GATTR.TYPTR := NIL END                        COMP      570
     END (*CASE*)                                                           COMP      570
    ELSE GATTR.TYPTR := NIL;                                                COMP      570
    IF LOP <> OROP THEN                                                     COMP      570
     WITH GATTR DO                                                          COMP      570
      BEGIN KIND := EXPR; EXPREG := K END                                   COMP      570
   END (*WHILE*)                                                            COMP      570
 END (*SIMPLEEXPRESSION*) ;                                                 COMP      570

BEGIN (*EXPRESSION*)                                                        COMP      570
 SIMPLEEXPRESSION(FSYS+[RELOP]);                                            COMP      571
 IF SY = RELOP THEN                                                         COMP      571
  BEGIN                                                                     COMP      571
   LATTR := GATTR; LOP := OP;                                               COMP      571
   INSYMBOL; SIMPLEEXPRESSION(FSYS);                                        COMP      571
   IF LATTR.TYPTR <> NIL THEN                                               COMP      571
    WITH LATTR.TYPTR↑ DO                                                    COMP      571
     IF (FORM = ARRAYS) AND (SIZE.WORDS > 1) AND COMPTYPES                  COMP      571
      (LATTR.TYPTR,GATTR.TYPTR,FALSE) THEN                                  COMP      571
      BEGIN                                                                 COMP      572
        LOADADDRESS(LATTR,II); LOADADDRESS(GATTR,JJ)                        COMP      572
      END                                                                   COMP      572
     ELSE                                                                   COMP      572
      BEGIN                                                                 COMP      572
       IF LOP = INOP THEN (* DECIDE WETHER ELEMENT EVENTUALLY               COMP      572
                             OUT OF [0..63] *)                              COMP      572
        BEGIN LVENTOUT:=TRUE;                                               COMP      572
         WITH LATTR DO                                                      COMP      572
          IF KIND = CST THEN                                                COMP      572
           BEGIN IF (CVAL.IVAL >= 0) AND (CVAL.IVAL <= 63) THEN             COMP      573
                                                 LVENTOUT:=FALSE            COMP      573
           END                                                              COMP      573
          ELSE IF KIND = VARBL THEN                                         COMP      573
           IF (TYPTR <> NIL) AND (TYPTR <> INTPTR) THEN                     COMP      573
            IF TYPTR↑.FORM IN [SCALAR,SUBRANGE] THEN                        COMP      573
             BEGIN GETBOUNDS(TYPTR,LOW,HIGH);                               COMP      573
              IF (LOW >= 0) AND (HIGH <= 63) THEN LVENTOUT:=FALSE           COMP      573
             END                                                            COMP      573
        END;                                                                COMP      573
       LOAD(LATTR,I); LOAD(GATTR,J);                                        COMP      574
      END;                                                                  COMP      574
      K := 0;                                                               COMP      574
   IF (LATTR.TYPTR <> NIL)AND (GATTR.TYPTR <> NIL) THEN                     COMP      574
    IF LOP = INOP THEN                                                      COMP      574
     IF GATTR.TYPTR↑.FORM = POWER THEN                                      COMP      574
      IF COMPTYPES(LATTR.TYPTR,GATTR.TYPTR↑.ELSET,FALSE) THEN               COMP      574
       BEGIN NEEDX(0,7,K);                                                  COMP      574
```

```
    IF  LVENTOUT THEN                                                    COMP  574
     BEGIN GEN15(43B,K,0,54); GEN15(11B,K,I,K);                          COMP  574
      GEN30(03B,1,K,IC+1,2)                                              COMP  575
     END;                                                                COMP  575
    NEEDB(L); GEN15(63B,L,I,0);                                          COMP  575
    DECREFX(I); DECREFX(J); GEN15(23B,K,L,J);          000072            COMP  575
    IF LVENTOUT THEN NOOP;                                               COMP  575
    GEN15(20B,K,0,59); BRGS[L].BCONT := FREE;                            COMP  575
    WITH GATTR DO                                                        COMP  575
     BEGIN TYPTR := BOOLPTR; CONDCD := PL END;                           COMP  575
   END                                                                   COMP  575
  ELSE BEGIN ERROR(129); GATTR.TYPTR := NIL END                          COMP  575
 ELSE BEGIN ERROR(130); GATTR.TYPTR := NIL END                           COMP  576
ELSE                                                                     COMP  576
 BEGIN                                                                   COMP  576
  IF NOT COMPTYPES(LATTR.TYPTR,GATTR.TYPTR,FALSE) THEN                   COMP  576
   IF COMPTYPES(LATTR.TYPTR,INTPTR,FALSE) THEN                           COMP  576
    BEGIN PACKANDNORM(I);                                                COMP  576
     LATTR.TYPTR := REALPTR                                              COMP  576
    END                                                                  COMP  576
   ELSE                                                                  COMP  576
    IF COMPTYPES(GATTR.TYPTR,INTPTR,FALSE) THEN                          COMP  576
     BEGIN PACKANDNORM(J);                                               COMP  577
      GATTR.TYPTR := REALPTR                                             COMP  577
     END;                                                                COMP  577
  IF COMPTYPES(LATTR.TYPTR,GATTR.TYPTR,FALSE) THEN                       COMP  577
   BEGIN                                                                 COMP  577
    CASE LATTR.TYPTR↑.FORM OF                                            COMP  577
     SCALAR,                                                             COMP  577
     SUBRANGE:                                                           COMP  577
      ;                                                                  COMP  577
     POINTER:                                                            COMP  577
      BEGIN                                                              COMP  578
       IF LOP IN [LTOP,LEOP,GTOP,GEOP] THEN ERROR(131);                  COMP  578
      END;                                                               COMP  578
     POWER:                                                              COMP  578
      BEGIN IF LOP IN [LTOP,GTOP] THEN ERROR(132);                       COMP  578
     END;                                                                COMP  578
     ARRAYS:                                                             COMP  578
      IF NOT STRING(LATTR.TYPTR) THEN ERROR(134);                        COMP  578
     RECORDS: ERROR(134);                                                COMP  578
     FILES:                                                              COMP  578
      ERROR(133)                                                         COMP  579
    END;                                                                 COMP  579
  CASE LATTR.TYPTR↑.FORM OF                                              COMP  579
   SCALAR,                                                               COMP  579
   SUBRANGE,                                                             COMP  579
   POINTER:                                                              COMP  579
    BEGIN                                                                COMP  579
     IF COMPTYPES(LATTR.TYPTR,REALPTR,FALSE) THEN                        COMP  579
      LOPCD := 31B                                                       COMP  579
     ELSE                                                                COMP  579
      LOPCD := 37B;                                                      COMP  580
     CASE LOP OF                                                         COMP  580
      LTOP,GEOP: OPERATION(LOPCD,K,I,J);                                 COMP  580
      LEOP,GTOP: OPERATION(LOPCD,K,J,I);                                 COMP  580
      NEOP,EQOP: OPERATION(37B,K,I,J)                                    COMP  580
     END;                                                                COMP  580
     WITH GATTR DO                                                       COMP  580
      BEGIN TYPTR := BOOLPTR;                                            COMP  580
       CASE LOP OF                                                       COMP  580
        LTOP,GTOP: CONDCD := PL;                                         COMP  580
        LEOP,GEOP: CONDCD := NG;                                         COMP  581
           NEOP: CONDCD := ZR;                                           COMP  581
           EQOP: CONDCD := NZ                                            COMP  581
       END                                                               COMP  581
      END                                                                COMP  581
    END ;                                                                COMP  581
   POWER:                                                                COMP  581
    BEGIN                                                                COMP  581
     CASE LOP OF                                                         COMP  581
      LTOP,GTOP: ;                                                       COMP  581
      LEOP: OPERATION(15B,K,I,J);                                        COMP  582
      GEOP: OPERATION(15B,K,J,I);                                        COMP  582
      NEOP,EQOP: OPERATION(13B,K,I,J)                                    COMP  582
     END;                                                                COMP  582
     WITH GATTR DO                                                       COMP  582
      BEGIN TYPTR := BOOLPTR;                                            COMP  582
       IF LOP = NEOP THEN CONDCD := ZR ELSE CONDCD := NZ                 COMP  582
      END                                                                COMP  582
    END;                                                                 COMP  582
```

```
      ARRAYS:                                                                  COMP      582
       BEGIN                                                                   COMP      582
        WITH LATTR.TYPTR↑.SIZE DO                                              COMP      582
         BEGIN WRDS := WORDS; BTS := BITS END;                                 COMP      582
        IF WRDS = 0 THEN (*PART WORD COMPARISON*)                              COMP      582
         BEGIN                                                    000073       COMP      582
          NEEDX(0,7,L); GEN15(43B,L,0,BTS);                                    COMP      582
          DECREFX(I); NEEDX(0,7,II); GEN15(11B,II,L,I);                        COMP      583
          DECREFX(J); NEEDX(0,7,JJ); GEN15(11B,JJ,L,J);                        COMP      583
          DECREFX(L);                                                          COMP      583
          GEN15(20B,II,0,BTS); GEN15(20B,JJ,0,BTS);                            COMP      583
          IF LOP IN [LEOP,GTOP] THEN OPERATION(37B,K,JJ,II)                    COMP      584
          ELSE OPERATION(37B,K,II,JJ);                                         COMP      584
         END                                                                   COMP      584
        ELSE                                                                   COMP      584
         IF BTS <> 0 THEN ERROR(398)                                           COMP      584
         ELSE                                                                  COMP      584
          BEGIN                                                                COMP      584
           IF WRDS > 1 THEN                                                    COMP      584
            BEGIN                                                              COMP      584
             NEEDB(L); GEN15(66B,L,0,0);                                       COMP      584
             NEEDB(R); GEN30(61B,R,0,WRDS-1,0);                                COMP      585
             NOOP; LADDR := IC;                                                COMP      585
             NEEDX(1,5,I); NEEDX(1,5,J);                                       COMP      585
             ARGS[I].ACONT := UNSPECADDR;                                      COMP      585
             ARGS[J].ACONT := UNSPECADDR;                                      COMP      585
             GEN15(53B,I,II,L); GEN15(53B,J,JJ,L);                             COMP      585
            END;                                                               COMP      585
            IF LOP IN [GTOP,GEOP] THEN                                         COMP      585
             BEGIN K := I; I := J; J := K END;                                 COMP      585
            NEEDX(0,7,K);                                                      COMP      586
            IF LOP IN [LTOP,LEOP,GEOP,GTOP] THEN                               COMP      586
             BEGIN GEN15(17B,K,I,J); DECREFX(I); NEEDX(0,7,M);                 COMP      586
              IF (LOP IN [LEOP,GEOP]) AND (WRDS = 1) THEN                      COMP      586
               BEGIN GEN15(37B,M,J,I); NEEDX(0,7,I);                           COMP      586
                GEN15(15B,I,K,M)                                               COMP      586
               END                                                             COMP      586
              ELSE                                                             COMP      586
               BEGIN GEN15(37B,M,I,J); NEEDX(0,7,I);                           COMP      586
                GEN15(11B,I,K,M)                                               COMP      586
               END;                                                            COMP      586
              NEEDX(0,7,N); GEN15(15B,N,J,K); GEN15(12B,K,I,N);                COMP      587
              DECREFX(N); DECREFX(M);                                          COMP      587
             END                                                               COMP      587
            ELSE GEN15(37B,K,I,J);                                             COMP      587
            DECREFX(I); DECREFX(J);                                            COMP      587
            IF WRDS > 1 THEN                                                   COMP      587
             BEGIN GEN15(66B,L,L,1);                                           COMP      587
              IF LOP IN [LTOP,LEOP,GEOP,GTOP] THEN                             COMP      587
               BEGIN GEN30(03B,3,K,IC+2,2); GEN30(03B,1,M,IC+1,2)              COMP      587
               END                                                             COMP      587
              ELSE  GEN30(03B,1,K,IC+1,2);                                     COMP      588
              GEN30(06B,R,L,LADDR,2);                                          COMP      588
              IF LOP IN [LEOP,GEOP] THEN GEN15(14B,K,0,K);                     COMP      588
              NOOP;                                                            COMP      588
             DECREFX(II); DECREFX(JJ);                                         COMP      588
              BRGS[L].BCONT:=FREE; BRGS[R].BCONT:=FREE;                        COMP      588
            END                                                                COMP      588
          END;                                                                 COMP      588
       WITH GATTR DO                                                           COMP      588
        BEGIN TYPTR := BOOLPTR;                                                COMP      588
         CASE LOP OF                                                           COMP      589
          LTOP,GTOP: CONDCD := PL;                                             COMP      589
          LEOP,GEOP: IF WRDS = 0 THEN CONDCD := NG                             COMP      589
                     ELSE CONDCD := PL;                                        COMP      589
             NEOP: CONDCD := ZR;                                               COMP      589
             EQOP: CONDCD := NZ                                                COMP      589
         END                                                                   COMP      589
        END                                                                    COMP      589
      END;                                                                     COMP      589
      RECORDS,                                                                 COMP      589
      FILES:                                                                   COMP      590
     END (*CASE*) ;                                                            COMP      590
       END                                                                     COMP      590
      ELSE ERROR(129);                                                         COMP      590
    END (*SY <> INCP*)                                                         COMP      590
   ELSE GATTR.TYPTR := NIL;                                                    COMP      590
  WITH GATTR DO                                                                COMP      590
   BEGIN KIND := COND; CDR := K END;                                           COMP      590
 END (*SY = RELOP*) ;                                                          COMP      590
END (*EXPRESSION*) ;                                                           COMP      590
```

```
PROCEDURE GENFJMP(FADDR: ADDRRANGE);
 (*GENERATE A FALSE JUMP TO FADDR (ON GATTR)*)
 VAR I: REGNR;
BEGIN
 WITH GATTR DO
  IF TYPTR <> NIL THEN
   IF KIND = COND THEN
    BEGIN GEN30(03B,ORD(CONDCD),CDR,FADDR,2); DECREFX(CDR) END
   ELSE
    BEGIN LOAD(GATTR,I); GEN30(03B,0,I,FADDR,2); DECREFX(I) END
END (*GENFJMP*) ;

PROCEDURE ASSIGNMENT(FCP: CTP);
 VAR LATTR: ATTR;
BEGIN SELECTOR(FSYS+[BECOMES],FCP);
 IF SY = BECOMES THEN
  BEGIN
   LATTR := GATTR;
   INSYMBOL; EXPRESSION(FSYS);
   ASSIGNTO(LATTR);
  END (*SY = BECOMES*)
 ELSE ERROR(51)
END (*ASSIGNMENT*) ;

PROCEDURE GOTOSTATEMENT;
 LABEL 1;
 VAR LLP: LBP; I: LEVRANGE; LFSTOCC: LOCOFREF;
BEGIN CLEARREGS;
 IF SY = INTCONST THEN
  BEGIN LLP := FSTLABP;
   WHILE LLP <> FLABP DO (*DECIDE WHETHER LOCALLY DECLARED*)
    WITH LLP↑ DO
     IF LABVAL = IVAL THEN
      BEGIN
       IF DEFINED THEN GEN30(04B,0,0,LABADDR,2)
       ELSE
        BEGIN GEN30(04B,0,0,0,2); LFSTOCC := FSTOCC;
         LINKOCC(LFSTOCC); FSTOCC := LFSTOCC
        END;
       GOTO 1
      END
     ELSE LLP := NEXTLAB;
   WHILE LLP <> NIL DO (*DECIDE WHETHER GLOBALLY DECLARED*)
    WITH LLP↑ DO
     IF LABVAL = IVAL THEN
      BEGIN
       IF LCNT = 0 THEN
        IF LABCNT = 36 THEN ERROR(260)
        ELSE
         BEGIN LABCNT := LABCNT + 1;
          LCNT := LABCNT
         END;
       IF LABLEV = 1 THEN GEN15(66B,3,2,0)
       ELSE
        BEGIN GEN15(56B,1,5,0);
         FOR I := LABLEV TO LEVEL - 2 DO GEN15(53B,1,1,0);
         GEN15(63B,3,1,0)
        END;
       NOOP;
       GEN15(66B,6,5,0); GEN15(56B,1,5,1);
       GEN15(20B,1,0,42); GEN15(63B,5,1,0);
       GEN30(05B,3,5,IC-1,2);
       PASCL[7] := CHR(LCNT);
       SEARCHEXTID(PASCL);  GEN30(04B,0,0,0,0);
       GOTO 1
      END
     ELSE LLP := NEXTLAB;
   ERROR(167);
1:   INSYMBOL
  END
 ELSE ERROR(15)
END (*GOTOSTATEMENT*) ;

PROCEDURE COMPOUNDSTATEMENT;
BEGIN
 (*LOOP UNTIL SY <> SEMICOLON:*)
 REPEAT STATEMENT(FSYS+[SEMICOLON,ENDSY]);
  IF SY IN STATBEGSYS THEN ERROR(14);
 UNTIL NOT (SY IN STATBEGSYS);
 WHILE SY = SEMICOLON DO
```

```
  BEGIN INSYMBOL;                                                    COMP     599
   REPEAT STATEMENT(FSYS+[SEMICOLON,ENDSY]);                         COMP     599
    IF SY IN STATBEGSYS THEN ERROR(14)                               COMP     599
   UNTIL NOT (SY IN STATBEGSYS)                                      COMP     599
  END;                                                               COMP     599
 TEST1(ENDSY,13)                                                     COMP     599
END (*COMPOUNDSTATEMENET*) ;                          000075        COMP     599
                                                                     COMP     599
PROCEDURE IFSTATEMENT;                                               COMP     599
 VAR LPL1,LPL2: PLACE; I: REGNR;                                     COMP     600
   LARGS: ARGSTATUS; LXRGS: XRGSTATUS; LBRGS: BRGSTATUS;             COMP     600
   LBRG: BASREGS; LLEVELS: SET OF LEVRANGE;                          COMP     600
BEGIN EXPRESSION(FSYS+[THENSY]);                                     COMP     600
 IF NOT COMPTYPES(GATTR.TYPTR,BOOLPTR,FALSE) THEN ERROR(144);        COMP     600
 GENFJMP(0);                                                         COMP     600
 LPL1 := PC;                                                         COMP     600
 LARGS := ARGS; LXRGS := XRGS; LBRGS := BRGS;                        COMP     600
 LBRG := BRG; LLEVELS := LEVELS;                                     COMP     600
 TEST1(THENSY,52);                                                   COMP     600
 STATEMENT(FSYS+[ELSESY]);                                           COMP     601
 IF SY = ELSESY THEN                                                 COMP     601
  BEGIN GEN30(04B,0,0,0,2); LPL2 := PC;                              COMP     601
   NOOP; INS(IC,LPL1);                                               COMP     601
   ARGS := LARGS; XRGS := LXRGS; BRGS := LBRGS;                      COMP     601
   BRG := LBRG; LEVELS := LLEVELS;                                   COMP     601
   INSYMBOL; STATEMENT(FSYS);                                        COMP     601
   NOOP; INS(IC,LPL2);                                               COMP     601
  END                                                                COMP     601
 ELSE                                                                COMP     601
   BEGIN NOOP; INS(IC,LPL1) END;                                     COMP     602
 CLEARREGS                                                           COMP     602
END (*IFSTATEMENT*) ;                                                COMP     602
                                                                     COMP     602
PROCEDURE CASESTATEMENT;                                             COMP     602
 LABEL 1;                                                            COMP     602
 CONST CASLABMAX = 1000;                                             COMP     602
 TYPE CIP = ↑CASEREC;                                                COMP     602
    CASEREC = PACKED                                                 COMP     602
         RECORD NEXT: CIP;                                           COMP     602
          CSLAB: INTEGER;                                            COMP     603
          CSADDR: ADDRRANGE                                          COMP     603
         END;                                                        COMP     603
 VAR LSP,LSP1: STP; FSTPTR,LPT1,LPT2,LPT3: CIP; LVAL: VALU;          COMP     603
   FSTLOCP: LOCOFREF; LPL,LPL1,LPL2: PLACE;                          COMP     603
   LDEBUG,EXITLOOP: BOOLEAN;                                         COMP     603
   I,J,K: REGNR; LARGS: ARGSTATUS; LBRGS: BRGSTATUS;                 COMP     603
   LXRGS: XRGSTATUS;                                                 COMP     603
   LBRG: BASREGS; LLEVELS: SET OF LEVRANGE;                          COMP     603
   LADDR: ADDRRANGE;                                                 COMP     603
   LMIN,LMAX: INTEGER;                                               COMP     604
BEGIN                                                                COMP     604
 EXPRESSION(FSYS+[OFSY,COMMA,COLON]);                                COMP     604
 LSP := GATTR.TYPTR; FSTLOCP := NIL;                                 COMP     604
 IF LSP <> NIL THEN                                                  COMP     604
  IF (LSP↑.FORM > SUBRANGE)OR COMPTYPES(LSP,REALPTR,FALSE) THEN      COMP     604
   BEGIN ERROR(144); LSP := NIL END;                                 COMP     604
 LOAD(GATTR,I);                                                      COMP     604
 IF DEBUG THEN                                                       COMP     604
  BEGIN GEN30(51B,0,0,IC,0); NEEDX(0,7,J);                           COMP     604
   GEN30(71B,J,0,0,0); LPL1 := PC;                                   COMP     605
   GEN15(37B,J,I,J); NEEDX(0,7,K);                                   COMP     605
   GEN30(71B,K,0,0,0); LPL2 := PC;                                   COMP     605
   GEN15(37B,K,K,I); GEN15(12B,K,K,J);                               COMP     605
   GEN30(03B,3,K,INXERR,0); DECREFX(J); DECREFX(K)                   COMP     605
  END;                                                               COMP     605
 LDEBUG := DEBUG;                                                    COMP     605
 NEEDB(J); GEN15(63B,J,I,0);                                         COMP     605
 DECREFX(I); GEN30(02B,J,0,0,2); LPL := PC;                          COMP     605
 BRGS[J].BCONT := FREE; LARGS := ARGS; LXRGS := XRGS; LBRGS := BRGS; COMP     605
 LBRG := BRG; LLEVELS := LEVELS;                                     COMP     606
 TEST1(OFSY,8);                                                      COMP     606
 FSTPTR := NIL; LPT3 := NIL;                                         COMP     606
 (*LOOP UNTIL SY <> SEMICOLON*)                                      COMP     606
 REPEAT                                                              COMP     606
  IF NOT (SY IN [SEMICOLON,ENDSY]) THEN                              COMP     606
  BEGIN NOOP;                                                        COMP     606
  (*LOOP UNTIL SY <> COMMA:*)                                        COMP     606
  REPEAT CONSTANT(FSYS+[COMMA,COLON],LSP1,LVAL);                     COMP     606
   IF LSP1 <> NIL THEN                                               COMP     606
    IF COMPTYPES(LSP,LSP1,FALSE) THEN                                COMP     607
     BEGIN LPT1 := FSTPTR; LPT2 := NIL;                              COMP     607
```

```
      WHILE LPT1 <> NIL DO                                                    COMP     607
       WITH LPT1↑ DO                                                          COMP     607
        BEGIN                                                                 COMP     607
         IF CSLAB <= LVAL.IVAL THEN                          000076           COMP     607
          BEGIN IF CSLAB = LVAL.IVAL THEN ERROR(156);                         COMP     607
           GOTO 1                                                             COMP     607
          END;                                                                COMP     607
         LPT2 := LPT1; LPT1 := NEXT                                           COMP     607
        END;                                                                  COMP     608
1:      NEW(LPT3);                                                            COMP     608
      WITH LPT3↑ DO                                                           COMP     608
       BEGIN NEXT := LPT1; CSLAB := LVAL.IVAL;                                COMP     608
        CSADDR := IC                                                          COMP     608
       END;                                                                   COMP     608
      IF LPT2 = NIL THEN FSTPTR := LPT3                                       COMP     608
      ELSE LPT2↑.NEXT := LPT3                                                 COMP     608
     END                                                                      COMP     608
    ELSE ERROR(147);                                                          COMP     608
  EXITLOOP := SY <> COMMA;                                                    COMP     609
   IF NOT EXITLOOP THEN INSYMBOL                                              COMP     609
  UNTIL EXITLOOP;                                                             COMP     609
  TEST1(COLON,5);                                                             COMP     609
  REPEAT STATEMENT(FSYS+[SEMICOLON]);                                         COMP     609
   IF SY IN STATBEGSYS THEN ERROR(14);                                        COMP     609
  UNTIL NOT (SY IN STATBEGSYS);                                               COMP     609
  GEN30(04B,0,0,0,2); LINKCCC(FSTLOCP);                                       COMP     609
  END (*SY <> ENDSY*) ;                                                       COMP     609
  EXITLOOP := SY <> SEMICOLON;                                                COMP     609
  IF NOT EXITLOOP THEN                                                        COMP     610
  BEGIN ARGS := LARGS; XRGS := LXRGS; BRGS := LBRGS;                          COMP     610
   BRG := LBRG; LEVELS := LLEVELS;                                            COMP     610
   INSYMBOL                                                                   COMP     610
  END;                                                                        COMP     610
 UNTIL EXITLOOP;                                                              COMP     610
 IF FSTPTR <> NIL THEN                                                        COMP     610
  BEGIN LMAX := FSTPTR↑.CSLAB;                                                COMP     610
   IF LDEBUG THEN INS(LMAX,LPL2);                                             COMP     610
   (*REVERSE POINTERS*)                                                       COMP     610
   LPT1 := FSTPTR; FSTPTR := NIL;                                             COMP     611
   REPEAT LPT2 := LPT1↑.NEXT; LPT1↑.NEXT := FSTPTR;                           COMP     611
    FSTPTR := LPT1; LPT1 := LPT2                                              COMP     611
   UNTIL LPT1 = NIL;                                                          COMP     611
   LMIN := FSTPTR↑.CSLAB;                                                     COMP     611
   IF LDEBUG THEN INS(LMIN,LPL1);                                             COMP     611
   NOOP; INS(IC-LMIN,LPL);                                                    COMP     611
   IF LMAX - LMIN < CASLABMAX THEN                                            COMP     611
    BEGIN LADDR := IC + LMAX - LMIN + 1;                                      COMP     611
     REPEAT                                                                   COMP     611
      WITH FSTPTR↑ DO                                                         COMP     612
       BEGIN                                                                  COMP     612
        WHILE CSLAB > LMIN DO                                                 COMP     612
         BEGIN                                                                COMP     612
          IF LDEBUG THEN                                                      COMP     612
           BEGIN GEN30(51B,0,0,IC,0); GEN30(04B,0,0,INXERR,0)                 COMP     612
           END                                                                COMP     612
          ELSE                                                                COMP     612
           BEGIN GEN30(04B,0,0,LADDR,2); NOOP END;                            COMP     612
          LMIN := LMIN + 1                                                    COMP     612
         END;                                                                 COMP     613
        GEN30(04B,0,0,CSADDR,2); NOOP;                                        COMP     613
        FSTPTR := NEXT; LMIN := LMIN + 1                                      COMP     613
       END                                                                    COMP     613
     UNTIL FSTPTR = NIL;                                                      COMP     613
     WHILE FSTLOCP <> NIL DO                                                  COMP     613
      WITH FSTLOCP↑ DO                                                        COMP     613
      BEGIN INS(LADDR,LOC); FSTLOCP := NXTREF END                             COMP     613
    END                                                                       COMP     613
   ELSE ERROR(157)                                                            COMP     613
  END (*FSTPTR <> NIL*)                                                       COMP     614
 ELSE ERROR(6);                                                               COMP     614
 CLEARREGS;                                                                   COMP     614
  TEST1(ENDSY,13)                                                             COMP     614
END (*CASESTATEMENT*) ;                                                       COMP     614
                                                                              COMP     614
PROCEDURE REPEATSTATEMENT;                                                    COMP     614
 VAR LADDR: ADDRRANGE; I: REGNR;                                              COMP     614
BEGIN CLEARREGS;                                                              COMP     614
 NOOP; LADDR := IC;                                                           COMP     614
 (*LOOP UNTIL SY <> SEMICOLON:*)                                              COMP     615
 REPEAT STATEMENT(FSYS+[SEMICOLON,UNTILSY]);                                  COMP     615
  IF SY IN STATBEGSYS THEN ERROR(14)                                          COMP     615
```

```
 UNTIL NOT (SY IN STATBEGSYS);                                                  COMP      615
 WHILE SY = SEMICOLON DO                                                        COMP      615
  BEGIN INSYMBOL;                                                               COMP      615
   REPEAT STATEMENT(FSYS+[SEMICOLON,UNTILSY]);                                  COMP      615
    IF SY IN STATBEGSYS THEN ERROR(14);                                         COMP      615
   UNTIL NOT (SY IN STATBEGSYS);                                    000077     COMP      615
  END;                                                                          COMP      615
 IF SY = UNTILSY THEN                                                           COMP      616
  BEGIN INSYMBOL; EXPRESSION(FSYS);                                             COMP      616
   IF NOT COMPTYPES(GATTR.TYPTR,BOOLPTR,FALSE) THEN ERROR(144);                 COMP      616
   GENFJMP(LADDR)                                                               COMP      616
  END                                                                           COMP      616
 ELSE ERROR(53)                                                                 COMP      616
END (*REPEATSTATEMENT*) ;                                                       COMP      616
                                                                                COMP      616
PROCEDURE WHILESTATEMENT;                                                       COMP      616
 VAR LADDR: ADDRRANGE; LPL: PLACE; I: REGNR;                                    COMP      616
   LARGS: ARGSTATUS; LXRGS: XRGSTATUS; LBRGS: BRGSTATUS;                        COMP      617
   LBRG: BASREGS; LLEVELS: SET OF LEVRANGE;                                     COMP      617
BEGIN CLEARREGS;                                                                COMP      617
 NOOP; LADDR := IC;                                                             COMP      617
 EXPRESSION(FSYS+[DOSY]);                                                       COMP      617
 IF NOT COMPTYPES(GATTR.TYPTR,BOOLPTR,FALSE) THEN ERROR(144);                   COMP      617
 GENFJMP(0);                                                                    COMP      617
 LPL := PC;                                                                     COMP      617
 LARGS := ARGS; LXRGS := XRGS; LBRGS := BRGS;                                   COMP      617
 LBRG := BRG; LLEVELS := LEVELS;                                                COMP      617
 TEST1(DOSY,54);                                                                COMP      618
 STATEMENT(FSYS);                                                               COMP      618
 GEN30(04B,0,0,LADDR,2); NOOP; INS(IC,LPL);                                     COMP      618
 ARGS := LARGS; XRGS := LXRGS; BRGS := LBRGS;                                   COMP      618
 BRG := LBRG; LEVELS := LLEVELS;                                                COMP      618
END (*WHILESTATEMENT*) ;                                                        COMP      618
                                                                                COMP      618
PROCEDURE FORSTATEMENT;                                                         COMP      618
 VAR LATTR,LIMIT: ATTR; LSP: STP; LSY: SYMBOL; LADDR: ADDRRANGE;                COMP      618
   LPL: PLACE; I,J,K: REGNR; LCP: CTP;                                          COMP      618
   LMIN,LMAX: INTEGER;                                                          COMP      619
BEGIN (* DEFINE LATTR TO PREVENT BLOW UP IN CASE OF ERRORS*)                    COMP      619
 WITH LATTR DO                                                                  COMP      619
  BEGIN TYPTR := NIL; KIND := VARBL; WORDACC := DRCT;                           COMP      619
  VLEVEL := LEVEL; CWDISPL := 0; PCKD := FALSE                                  COMP      619
 END;                                                                           COMP      619
 IF SY = IDENT THEN                                                             COMP      619
  BEGIN SEARCHID([VARS],LCP);                                                   COMP      619
   WITH LCP↑, LATTR DO                                                          COMP      619
    BEGIN TYPTR := IDTYPE;                                                      COMP      619
     IF VKIND = DRCT THEN                                                       COMP      620
      IF VLEV IN [1,LEVEL] THEN                                                 COMP      620
       BEGIN VLEVEL := VLEV; CWDISPL := VADDR END                               COMP      620
      ELSE ERROR(155)                                                           COMP      620
     ELSE ERROR(180)                                                            COMP      620
    END;                                                                        COMP      620
   IF LATTR.TYPTR <> NIL THEN                                                   COMP      620
    IF (LATTR.TYPTR↑.FORM > SUBRANGE)                                           COMP      620
     OR COMPTYPES(REALPTR,LATTR.TYPTR,FALSE) THEN                               COMP      620
     BEGIN ERROR(143); LATTR.TYPTR := NIL END;                                  COMP      620
   INSYMBOL                                                                     COMP      621
  END                                                                           COMP      621
 ELSE                                                                           COMP      621
  BEGIN ERROR(2); SKIP(FSYS+[BECOMES,TOSY,DOWNTOSY,DOSY]) END;                  COMP      621
 I := 0;                                                                        COMP      621
 IF SY = BECOMES THEN                                                           COMP      621
  BEGIN INSYMBOL; EXPRESSION(FSYS+[TOSY,DOWNTOSY,DOSY]);                        COMP      621
   IF GATTR.TYPTR <> NIL THEN                                                   COMP      621
    IF GATTR.TYPTR↑.FORM > SUBRANGE THEN ERROR(144)                             COMP      621
    ELSE                                                                        COMP      621
     IF COMPTYPES(LATTR.TYPTR,GATTR.TYPTR,FALSE) THEN                           COMP      622
      BEGIN LOAD(GATTR,I);                                                      COMP      622
       IF I <> 6 THEN                                                           COMP      622
        BEGIN NEEDX(6,6,J); BXIXJ(6,I) END;                                     COMP      622
       WITH XRGS[6] DO                                                          COMP      622
        BEGIN XCONT := OTHER; REFNR := 1 END                                    COMP      622
      END                                                                       COMP      622
     ELSE ERROR(145)                                                            COMP      622
  END                                                                           COMP      622
 ELSE                                                                           COMP      622
  BEGIN ERROR(51); SKIP(FSYS+[TOSY,DOWNTOSY,DOSY]) END;                         COMP      623
 LSY := SY; LIMIT.TYPTR := NIL;                                                 COMP      623
 IF SY IN [TOSY,DOWNTOSY] THEN                                                  COMP      623
  BEGIN INSYMBOL; EXPRESSION(FSYS+[DOSY]);                                      COMP      623
```

```
    IF GATTR.TYPTR <> NIL THEN                                                  COMP      623
    IF GATTR.TYPTR^.FORM > SUBRANGE THEN ERROR(144)          000078             COMP      623
     ELSE                                                                       COMP      623
      IF COMPTYPES(LATTR.TYPTR,GATTR.TYPTR,FALSE) THEN                          COMP      623
       BEGIN LIMIT := GATTR; LOAD(GATTR,I);                                     COMP      623
        IF (LIMIT.KIND = CST)AND (ABS(LIMIT.CVAL.IVAL) < TWOTO17)               COMP      623
         THEN CLEARREGS                                                         COMP      624
        ELSE                                                                    COMP      624
         BEGIN                                                                  COMP      624
          IF I IN [0,6,7] THEN                                                  COMP      624
           BEGIN NEEDX(1,5,K); BXIXJ(K,I); I := K END;                          COMP      624
          CLEARREGS;                                                            COMP      624
          WITH XRGS[I] DO                                                       COMP      624
           BEGIN XCONT := SIMPVAR; REFNR := 0; LASTREF := IC;                   COMP      624
            SHFTCNT := 0;                                                       COMP      624
            XLEV := LEVEL; XADDR := LC                                          COMP      624
           END;                                                                 COMP      625
          WITH LIMIT DO                                                         COMP      625
           BEGIN KIND := VARBL; WORDACC := DRCT;                                COMP      625
            VLEVEL := LEVEL; CWDISPL := LC; PCKD := FALSE                       COMP      625
           END;                                                                 COMP      625
          GEN15(10B,7,I,0); GEN30(51B,7,BRG[LEVEL],LC,0);                       COMP      625
          LC := LC + 1;                                                         COMP      625
          IF LC > LCMAX THEN LCMAX := LC;                                       COMP      625
         END;                                                                   COMP      625
        WITH XRGS[6] DO                                                         COMP      625
         BEGIN XCONT := SIMPVAR; REFNR := 1; LASTREF := IC;                     COMP      626
          SHFTCNT := 0;                                                         COMP      626
          XLEV := LATTR.VLEVEL; XADDR := LATTR.CWDISPL                          COMP      626
         END                                                                    COMP      626
       END                                                                      COMP      626
      ELSE ERROR(145)                                                           COMP      626
  END                                                                           COMP      626
 ELSE BEGIN ERROR(55); SKIP(FSYS+[DOSY]) END;                                   COMP      626
 TEST1(DOSY,54);                                                                COMP      626
 NOOP; LADDR := IC; NEEDX(0,7,K);                                               COMP      626
 IF LSY = TOSY THEN GEN15(37B,K,1,6) ELSE GEN15(37B,K,6,I);                     COMP      627
  GEN30(03B,3,K,0,2); LPL := PC; DECREFX(K);                                    COMP      627
 IF LATTR.TYPTR <> NIL THEN                                                     COMP      627
  BEGIN                                                                         COMP      627
   IF DEBUG THEN                                                                COMP      627
    IF LATTR.TYPTR <> INTPTR THEN                                               COMP      627
     BEGIN GETBOUNDS(LATTR.TYPTR,LMIN,LMAX);                                    COMP      627
      CHECKBNDS(6,LMIN,LMAX,ASSERR)                                             COMP      627
     END;                                                                       COMP      627
   GEN30(51B,6,BRG[LATTR.VLEVEL],LATTR.CWDISPL,0);                              COMP      627
   WITH ARGS[6] DO                                                              COMP      628
    BEGIN ACONT := SIMPADDR; ALEV := LATTR.VLEVEL;                              COMP      628
     AADDR := LATTR.CWDISPL                                                     COMP      628
    END                                                                         COMP      628
  END;                                                                          COMP      628
 DECREFX(6);                                                                    COMP      628
 STATEMENT(FSYS);                                                               COMP      628
 LOAD(LATTR,J); NEEDX(0,7,K); GEN15(76B,K,1,0);                                 COMP      628
 IF LSY = TOSY THEN GEN15(36B,6,J,K) ELSE GEN15(37B,6,J,K);                     COMP      628
 DECREFX(J); DECREFX(K);                                                        COMP      628
 IF LIMIT.TYPTR <> NIL THEN                                                     COMP      629
  IF LIMIT.KIND = CST THEN                                                      COMP      629
   BEGIN NEEDX(I,I,I);                                                          COMP      629
    IF LIMIT.CVAL.IVAL = 1 THEN                                                 COMP      629
     GEN15(76B,I,1,0)                                                           COMP      629
    ELSE GEN30(71B,I,0,LIMIT.CVAL.IVAL,0)                                       COMP      629
   END                                                                          COMP      629
  ELSE                                                                          COMP      629
   BEGIN LOAD(LIMIT,J);                                                         COMP      629
    IF J <> I THEN BXIXJ(I,J);                                                  COMP      629
    LC := LC - 1                                                                COMP      630
   END;                                                                         COMP      630
 GEN30(04B,0,0,LADDR,2); NOOP; INS(IC,LPL);                                     COMP      630
 CLEARREGS                                                                      COMP      630
END (*FORSTATEMENT*) ;                                                          COMP      630
                                                                                COMP      630
PROCEDURE WITHSTATEMENT;                                                        COMP      630
 VAR LCP: CTP; OLDTOP: DISPRANGE; LLC: ADDRRANGE; LATTR: ATTR;                  COMP      630
   I: REGNR; EXITLOOP: BOOLEAN;                                                 COMP      630
BEGIN OLDTOP := TOP; LLC := LC;                                                 COMP      630
 (*LOOP UNTIL SY <> COMMA:*)                                                    COMP      631
 REPEAT                                                                         COMP      631
  IF SY = IDENT THEN                                                            COMP      631
   BEGIN SEARCHID([VARS,FIELD],LCP); INSYMBOL END                               COMP      631
  ELSE BEGIN ERROR(2); LCP := UVARPTR END;                                      COMP      631
```

```
    SELECTOR(FSYS+[COMMA,DOSY],LCP);                                                    COMP    631
    IF GATTR.TYPTR <> NIL THEN                                                          COMP    631
     IF GATTR.TYPTR↑.FORM = RECORDS THEN                                                COMP    631
      IF TOP < DISPLIMIT THEN                                                           COMP    631
       BEGIN TOP := TOP + 1;                                                            COMP    631
        WITH DISPLAY[TOP], GATTR DO                                    000079           COMP    632
         BEGIN FNAME := TYPTR↑.FIELDS; OCCUR := REC;                                    COMP    632
          IF WORDACC = DRCT THEN                                                        COMP    632
           BEGIN WACC := DRCT;                                                          COMP    632
            LEV := VLEVEL; CWDSPL := CWDISPL                                            COMP    632
           END                                                                          COMP    632
          ELSE                                                                          COMP    632
           BEGIN                                                                        COMP    632
            LOADADDRESS(GATTR,I);                                                       COMP    632
            WITH LATTR DO                                                               COMP    632
             BEGIN TYPTR := GATTR.TYPTR; KIND := VARBL;                                 COMP    633
              WORDACC := DRCT; VLEVEL := LEVEL;                                         COMP    633
              CWDISPL := LC; PCKD := FALSE                                              COMP    633
             END;                                                                       COMP    633
            STORE(LATTR,I);                                                             COMP    633
            LEV := LEVEL; CWDSPL := LC;                                                 COMP    633
            LC := LC + 1;                                                               COMP    633
            WACC := INDRCT                                                              COMP    633
           END;                                                                         COMP    633
          IF PCKD THEN                                                                  COMP    633
           BEGIN PKD := TRUE;                                                           COMP    634
            IF BITREG = NONE THEN                                                       COMP    634
             BEGIN BACC := DRCT; BDSPL := CBDISPL END                                   COMP    634
            ELSE                                                                        COMP    634
             BEGIN                                                                      COMP    634
              IF CBDISPL <> 0 THEN                                                      COMP    634
               BEGIN NEEDX(0,7,I);                                                      COMP    634
                IF CBDISPL = 1 THEN GEN15(76B,I,0,1)                                    COMP    634
                ELSE GEN30(71B,I,0,CBDISPL,0);                                          COMP    634
                GEN15(36B,I,VBDISPL,I);                                                 COMP    634
                DECREFX(VBDISPL)                                                        COMP    635
               END                                                                      COMP    635
              ELSE I := VBDISPL;                                                        COMP    635
              WITH LATTR DO                                                             COMP    635
               BEGIN TYPTR := GATTR.TYPTR; KIND := VARBL;                               COMP    635
                WORDACC := DRCT; VLEVEL := LEVEL;                                       COMP    635
                CWDISPL := LC; PCKD := FALSE                                            COMP    635
               END;                                                                     COMP    635
              STORE(LATTR,I);                                                           COMP    635
              BACC := INDRCT; BDSPL := LC;                                              COMP    635
              LC := LC + 1;                                                             COMP    636
             END                                                                        COMP    636
           END (*PCKD*)                                                                 COMP    636
          ELSE PKD := FALSE                                                             COMP    636
         END                                                                            COMP    636
       END                                                                              COMP    636
      ELSE ERROR(250)                                                                   COMP    636
     ELSE ERROR(140);                                                                   COMP    636
    EXITLOOP := SY <> COMMA;                                                            COMP    636
    IF NOT EXITLOOP THEN INSYMBOL                                                       COMP    636
    UNTIL EXITLOOP;                                                                     COMP    637
    IF LC > LCMAX THEN LCMAX := LC;                                                     COMP    637
    TEST1(DOSY,54);                                                                     COMP    637
    STATEMENT(FSYS);                                                                    COMP    637
    (*DISPOSE LOCALLY USED X-REGISTERS*)                                                COMP    637
    FOR I := 0 TO 7 DO                                                                  COMP    637
     WITH XRGS[I] DO                                                                    COMP    637
      IF XCONT = SIMPVAR THEN                                                           COMP    637
       IF (XLEV = LEVEL)AND (XADDR >= LLC) THEN XCONT := AVAIL;                         COMP    637
    FOR I := 0 TO 7 DO                                                                  COMP    637
     WITH XRGS[I] DO                                                                    COMP    638
      IF XCONT = INDVAR THEN                                                            COMP    638
       IF XRGS[XREG].XCONT = AVAIL THEN XCONT := AVAIL;                                 COMP    638
    TOP := OLDTOP; LC := LLC                                                            COMP    638
  END (*WITHSTATEMENT*) ;                                                               COMP    638
                                                                                        COMP    638
BEGIN (*STATEMENT*)                                                                     COMP    638
  IF SY = INTCONST THEN (*LABEL*)                                                       COMP    638
   BEGIN CLEARREGS; NOOP;                                                               COMP    638
    IF TRAPSET THEN                                                                     COMP    638
    IF IVAL = TRAPLAB THEN INS(IC,LPL2);                                                COMP    639
    LLP := FSTLABP;                                                                     COMP    639
    WHILE LLP <> FLABP DO                                                               COMP    639
     WITH LLP↑ DO                                                                       COMP    639
      IF LABVAL = IVAL THEN                                                             COMP    639
       BEGIN
```

```
        IF DEFINED THEN ERROR(165)                                                    COMP      639
        ELSE                                                                          COMP      639
         BEGIN LOCP := FSTOCC;                                                        COMP      639
          WHILE LOCP <> NIL DO                                  000080                COMP      639
           WITH LOCP^ DO                                                              COMP      640
            BEGIN INS(IC,LOC); LOCP := NXTREF END;                                    COMP      640
          DEFINED := TRUE; LABADDR := IC                                              COMP      640
         END;                                                                         COMP      640
        GOTO 1                                                                        COMP      640
       END                                                                            COMP      640
      ELSE LLP := NEXTLAB;                                                            COMP      640
    ERROR(167);                                                                       COMP      640
1:  INSYMBOL;                                                                         COMP      640
    TEST1(COLON,5)                                                                    COMP      640
   END;                                                                               COMP      641
  IF NOT (SY IN FSYS+[IDENT]) THEN                                                    COMP      641
   BEGIN ERROR(6); SKIP(FSYS) END;                                                    COMP      641
  IF SY IN STATBEGSYS+[IDENT] THEN                                                    COMP      641
   BEGIN                                                                              COMP      641
    LASTSY := SY;                                                                     COMP      641
    IF SY = IDENT                                                                     COMP      641
     THEN BEGIN SEARCHID([VARS,FIELD,FUNC,PROC],LCP);                                 COMP      641
      INSYMBOL;                                                                       COMP      641
      IF LCP^.KLASS = PROC                                                            COMP      641
       THEN CALL(FSYS,LCP)                                                            COMP      642
       ELSE ASSIGNMENT(LCP)                                                           COMP      642
      END                                                                             COMP      642
     ELSE BEGIN                                                                       COMP      642
      INSYMBOL;                                                                       COMP      642
      CASE LASTSY OF                                                                  COMP      642
       BEGINSY   : COMPOUNDSTATEMENT;                                                 COMP      642
       GOTOSY    : GOTOSTATEMENT;                                                     COMP      642
       IFSY      : IFSTATEMENT;                                                       COMP      642
       CASESY    : CASESTATEMENT;                                                     COMP      642
       WHILESY   : WHILESTATEMENT;                                                    COMP      643
       REPEATSY  : REPEATSTATEMENT;                                                   COMP      643
       FORSY     : FORSTATEMENT;                                                      COMP      643
       WITHSY    : WITHSTATEMENT                                                      COMP      643
       END;                                                                           COMP      643
      END;                                                                            COMP      643
    TEST2(FSYS,6,[])                                                                  COMP      643
   END                                                                                COMP      643
 END (*STATEMENT*) ;                                                                  COMP      643
                                                                                      COMP      643
BEGIN (*BODY*)                                                                        COMP      644
  DP := FALSE; RCIX := 0; RCP := 15;                                                  COMP      644
 WITH PC DO BEGIN SIX := 1; CIX := 0; CP := 4 END;                                    COMP      644
 NEW(CSEGP); CSEGP^.NXTSEG := NIL;                                                    COMP      644
  BONUS[SHRTCST] := 20; BONUS[LONGCST] := 10;                                         COMP      644
  BONUS[SIMPVAR] := 4; BONUS[INDVAR] := 3;                                            COMP      644
  BRG[LEVEL] := 5; BRG[1] := 2; BRG[0] := 6;                                          COMP      644
 IC := 0; CADDR := 0;                                                                 COMP      644
 EXT := NIL; EXTROOT := NIL; EXTIDX := 0; EXTRX := 0;                                 COMP      644
  CLEARREGS;                                                                          COMP      644
  FCLSP := NIL;                                                                       COMP      645
 IF LEVEL = 1 THEN                                                                    COMP      645
 BEGIN                                                                                COMP      645
  IF PMD THEN SEARCHEXTID(=PASCPMD    =);                                             COMP      645
  GEN30(40B,0,0,0,0);                                                                 COMP      645
  (*DISPL OF OUTPUT EFET[0]*)                                                         COMP      645
  GEN30(0,0,0,OUTPUTPTR^.VADDR+CHEFET,0);                                             COMP      645
  ALFINT.A := PROGNAME; GEN60(ALFINT.I); GEN30(0,0,0,IC+1,0);                         COMP      645
  GEN30(0,0,0,0,0); LPL1 := PC;                                                       COMP      645
  GEN30(61B,7,0,EXTFILS,0);                                                           COMP      645
  RJTOEXT(=P.INIT     =);                                                             COMP      646
 GEN30(61B,6,5,0,0); LPL := PC;                                                       COMP      646
  GEN30(61B,7,6,100,0); GEN30(51B,0,0,IC,0);                                          COMP      646
  GEN30(06B,7,4,STOVLERR,0);                                                          COMP      646
  (*OPEN GLOBAL FILES*)                                                               COMP      646
  OPENFILES(DISPLAY[1].FNAME);                                                        COMP      646
  (*SEARCH FOR UNDECLARED EXTERNAL FILES*)                                            COMP      646
  EXFILP := FEXFILP;                                                                  COMP      646
  WHILE EXFILP <> NIL DO                                                              COMP      646
   WITH EXFILP^ DO                                                                    COMP      646
    BEGIN                                                                             COMP      647
     IF NOT DECLARED AND (FILENAME<>=INPUT      =)                                    COMP      647
      AND (FILENAME <> =OUTPUT     =) THEN                                           COMP      647
      BEGIN ERROR(172); WRITELN;                                                      COMP      647
       WRITELN(= ***     UNDECLARED FILE: =,FILENAME);                                COMP      647
       SPACES;                                                                        COMP      647
      END;                                                                            COMP      647
```

```
    EXFILP := NXTP
   END;
  (*CALL P.GETS TO OPEN CURRENT SEGMENT*)
  IF INPUTPTR <> NIL THEN
   BEGIN GEN30(51B,1,BRG[1],INPUTPTR↑.VADDR+CHEFET,0);
    GEN30(71B,2,0,1,0);
    RJTOEXT(=P.GETS    =);
   END;
  (*SET DEFAULT VALUE FOR OUTPUT:*)
  GEN30(71B,6,0,1000,0); GEN30(51B,6,BRG[1],OUTPUTPTR↑.VADDR,0)
 END ELSE
 BEGIN
  ALFINT.A := FPROCP↑.NAME; GEN60(ALFINT.I); GEN30(0,0,0,IC+1,0);
 GEN30(0,0,0,0,0); LPL1 := PC;
  IF LEVEL > 2 THEN GEN15(56B,6,6,0);
  GEN15(76B,6,5,0); GEN15(20B,6,2,2); GEN15(12B,7,7,6);
  GEN15(66B,5,6,0); GEN15(56B,7,5,1);
 GEN30(61B,6,6,0,0); LPL := PC;
  GEN30(61B,7,6,100,0); GEN30(51B,0,0,IC,0);
  GEN30(06B,7,4,STOVLERR,0);
  LCP := FPROCP↑.NEXT;
  I := 0; LDISP := 3;
  WHILE (LCP <> NIL) AND (I < XPARMAX) DO   (*STORE X-I AT PARAM LOC*)
   BEGIN
    WITH GATTR DO
     BEGIN TYPTR := INTPTR; KIND := VARBL; WORDACC := DRCT;
      VLEVEL := LEVEL; CWDISPL := LDISP; PCKD := FALSE
     END;
    WITH XRGS[I] DO
     BEGIN XCONT := OTHER; REFNR := 1 END;
    STORE(GATTR,I);
    I := I + 1; LDISP := LDISP + 1; LCP := LCP↑.NEXT
   END;
  LOCFILES := FALSE;
  (*OPEN LOCAL FILES:*)
  OPENFILES(DISPLAY[TOP].FNAME);
  IF LOCFILES THEN CLEARREGS;   (*ASSUME REGISTER CONTENTS DESTROYED*)
  (*COPY MULTIPLE VALUES INTO LOCAL CELLS:*)
  LCP := FPROCP↑.NEXT; LDISP := 3;
  WHILE LCP <> NIL DO
   WITH LCP↑ DO
    BEGIN
     IF IDTYPE <> NIL THEN
      BEGIN LSZ := FULLWORDS(IDTYPE↑.SIZE);
       IF (KLASS = VARS)AND (VKIND = DRCT)AND (LSZ <> 1) THEN
        BEGIN GEN30(51B,4,5,LDISP,0); GEN30(71B,7,5,VADDR,0);
         NEEDB(I); GEN30(61B,I,0,LSZ-1,0); NOOP;
         GEN15(53B,5,4,I); GEN15(10B,6,5,0);
         GEN15(53B,6,7,I); GEN15(67B,I,I,1);
         GEN30(06B,I,0,IC-1,2); NOOP;
         BRGS[I].BCONT := FREE;
         FOR I := 4 TO 7 DO
          BEGIN XRGS[I].XCONT := AVAIL;
           ARGS[I].ACONT := UNSPECADDR
          END
        END
      END;
     LDISP := LDISP + 1; LCP := NEXT
    END;
 END;
 IF TRAPSET THEN
  BEGIN CLEARREGS; (*ASSUME REGISTER CONTENTS DESTROYED*)
   GEN30(51B,1,0,57B,0);   (* TRAP LOCATION *)
   GEN15(10B,6,1,0); GEN30(51B,6,BRG[LEVEL],TRADD,0);
   GEN30(71B,6,0,0,2); LPL2 := PC;
   GEN15(76B,1,BRG[LEVEL],0); GEN15(20B,1,2,2);
   GEN15(12B,6,6,1); GEN15(54B,6,1,0)
  END;
 LCMAX := LC;
 (*LOOP UNTIL SY <> SEMICOLON:*)
 REPEAT STATEMENT(FSYS+[SEMICOLON,ENDSY]);
  IF SY IN STATBEGSYS THEN ERROR(14)
 UNTIL NOT (SY IN STATBEGSYS);
 WHILE SY = SEMICOLON DO
  BEGIN INSYMBOL;
   REPEAT STATEMENT(FSYS+[SEMICOLON,ENDSY]);
    IF SY IN STATBEGSYS THEN ERROR(14)
   UNTIL NOT (SY IN STATBEGSYS);
  END;
 INS(LCMAX,LPL);
 TEST1(ENDSY,13);
```

```
  (*CLOSE LOCAL FILES:*)
  CLSP := FCLSP;
  IF CLSP <> NIL THEN CLEARREGS;
  WHILE CLSP <> NIL DO
   WITH CLSP↑ DO
    BEGIN GEN30(51B,1,BRG[LEVEL],EFETADDR,0);
     RJTOEXT(≡P.CLOSE   ≡);
     CLSP := NXTP
    END;
  IF LEVEL = 1 THEN
   BEGIN LGOHEAD(PROGNAME);
     (*CLOSE,OUTPUT*)
    GEN30(51B,1,BRG[1],OUTPUTPTR↑.VADDR+CHEFET,0);
    RJTOEXT(≡P.PUTS     ≡);
     (*END REQUEST:*)
    RJTOEXT(≡P.END      ≡);
   END
  ELSE
   BEGIN
    WITH FPROCP↑ DO
     IF PFCNT = 0 THEN LGOHEAD(NAME)
     ELSE
      BEGIN FFNAME(PFCNT); LGOHEAD(PNAME) END;
    IF FPROCP↑.KLASS = FUNC THEN (*LOAD FUNCTION RESULT:*)
     BEGIN
      WITH GATTR DO
       BEGIN TYPTR := FPROCP↑.IDTYPE; KIND := VARBL; WORDACC := DRCT;
        VLEVEL := LEVEL; CWDISPL := 2; PCKD := FALSE
       END;
      LOAD(GATTR,I);
      IF I <> 6 THEN BXIXJ(6,I)
     END;
    CLEARREGS;
    IF TRAPSET THEN
     BEGIN GEN30(51B,1,BRG[LEVEL],TRADD,0);
      GEN15(10B,7,1,0); GEN30(51B,7,0,57B,0)   (* TRAP LOCATION *)
     END;
    GEN15(56B,1,5,1); GEN15(66B,6,5,0); GEN15(63B,7,1,0);
    GEN15(20B,1,5,2); GEN15(63B,5,1,0); GEN30(02B,7,0,0,0)
   END;
  (*APPEND CONSTANTS TO CODE, FIXUP THEIR REFERENCES*)
  LCMAX := IC;
  LCSP := FSTCSP;
  WHILE LCSP <> NIL DO
   WITH LCSP↑ DO
    BEGIN LOCP := CREF;
     WHILE LOCP <> NIL DO
      WITH LOCP↑ DO
      BEGIN INS(IC,LOC); LOCP := NXTREF END;
     CREF := NIL;
     LP := CSTP;
     WHILE LP <> NIL DO
      WITH LP↑ DO
       BEGIN GEN60(CSVAL); LP := NXTCSP END;
     LCSP := NXTCSP;
    END;
  FSTCSP := LFSTCSP;
  INS(LCMAX*100000B+IC,LPL1);
  IF PMD THEN PMDINFO(DISPLAY[LEVEL].FNAME);
  GEN60(0);
  LGOTEXT; LGOEND;
 END (*BODY*) ;

BEGIN (*BLOCK*)
 FLABP := FSTLABP;  LFSTCSP := FSTCSP;
  LFORWCNT := 0;
 TRAPSET := FALSE;
 REPEAT
  IF SY = LABELSY THEN
   BEGIN INSYMBOL; LABELDECLARATION END;
  IF SY = CONSTSY THEN
   BEGIN INSYMBOL; CONSTDECLARATION END;
  IF SY = TYPESY THEN
   BEGIN INSYMBOL; TYPEDECLARATION END;
  IF SY = VARSY THEN
   BEGIN INSYMBOL; VARDECLARATION END;
  WHILE SY IN [PROCEDURESY,FUNCTIONSY] DO
   BEGIN LSY := SY; INSYMBOL; PROCEDUREDECLARATION(LSY) END;
  IF SY <> BEGINSY THEN
   BEGIN ERROR(18); SKIP(FSYS) END
 UNTIL SY IN STATBEGSYS;
```

```
  IF LFORWCNT > 0 THEN CHECKFORW(DISPLAY[LEVEL].FNAME);                    COMP     663
 TEST1(BEGINSY,17);                                                        COMP     663
 REPEAT BODY(FSYS+[CASESY]);                                               COMP     664
  IF SY <> FSY THEN                                                        COMP     664
   BEGIN ERROR(6); SKIP(FSYS) END                                          COMP     664
 UNTIL (SY = FSY)OR (SY IN BLOCKBEGSYS);                          000083    COMP     664
END (*BLOCK*) ;                                                           COMP     664
                                                                          COMP     664
                                                                          COMP     664
PROCEDURE PROGRAMME(FSYS: SETOFSYS);                                      COMP     664
 (*SHOULD PREFERABLY SURROUND PROC BLOCK WHICH WAS NOT POSSIBLE*)          COMP     664
 (*BEFORE BOOTSTRAP DUE TO COMPILER RESTRICTION ON PROC NESTING*)          COMP     664
 LABEL 1;                                                                 COMP     665
 VAR EXFILP: EXTFILEP; LCP: CTP;                                          COMP     665
BEGIN PROGNAME := ΞP.MAIN    Ξ; FEXFILP := NIL; EXTFILS := 0;             COMP     665
 (*DECLARE OUTPUT WITH FIX ADDRESS B2 + 6*)                               COMP     665
 NEW(LCP,VARS);                                                           COMP     665
 WITH LCP↑ DO                                                             COMP     665
  BEGIN NAME := ΞOUTPUT    Ξ; IDTYPE := TEXTPTR;                          COMP     665
   KLASS := VARS; VKIND := DRCT; NEXT := NIL;                             COMP     665
   VLEV := 1; VADDR := LC                                                 COMP     665
  END;                                                                    COMP     665
 ENTERID(LCP);                                                            COMP     666
 LC := LC + TEXTPTR↑.SIZE.WORDS;                                          COMP     666
 IF SY = PROGRAMSY THEN                                                   COMP     666
  BEGIN INSYMBOL;                                                         COMP     666
   IF SY = IDENT THEN                                                     COMP     666
    BEGIN PROGNAME := ID; INSYMBOL;                                       COMP     666
     TEST2([SEMICOLON,LPARENT],7,FSYS);                                   COMP     666
     IF SY = LPARENT THEN                                                 COMP     666
      BEGIN                                                               COMP     666
       REPEAT INSYMBOL;                                                   COMP     666
        IF SY = IDENT THEN                                                COMP     667
         BEGIN                                                            COMP     667
          IF ID = ΞINPUT     Ξ THEN                                       COMP     667
           BEGIN NEW(INPUTPTR,VARS);                                      COMP     667
            WITH INPUTPTR↑ DO                                             COMP     667
             BEGIN NAME := ΞINPUT     Ξ; IDTYPE := TEXTPTR;               COMP     667
              KLASS := VARS; VKIND := DRCT; NEXT := NIL;                  COMP     667
              VLEV := 1; VADDR := LC                                      COMP     667
             END;                                                         COMP     667
            ENTERID(INPUTPTR);                                            COMP     667
            LC := LC + TEXTPTR↑.SIZE.WORDS;                               COMP     668
           END                                                            COMP     668
          ELSE                                                            COMP     668
           IF ID = ΞOUTPUT    Ξ THEN OUTPUTPTR := LCP;                    COMP     668
          EXTFILS := EXTFILS + 1;                                         COMP     668
          EXFILP := FEXFILP;                                              COMP     668
          WHILE EXFILP <> NIL DO                                          COMP     668
           WITH EXFILP↑ DO                                                COMP     668
            BEGIN                                                         COMP     668
             IF FILENAME = ID THEN                                        COMP     668
              BEGIN ERROR(101); GOTO 1 END;                               COMP     669
             EXFILP := NXTP                                               COMP     669
            END;                                                          COMP     669
1:        NEW(EXFILP);                                                    COMP     669
          WITH EXFILP↑ DO                                                 COMP     669
           BEGIN FILENAME := ID; NXTP := FEXFILP;                         COMP     669
            DECLARED := FALSE;                                            COMP     669
            INSYMBOL;                                                     COMP     669
            IF OP = MUL THEN                                              COMP     669
             BEGIN READONLY := TRUE; INSYMBOL END                         COMP     669
            ELSE READONLY := FALSE;                                       COMP     670
            IF FILENAME = ΞINPUT     Ξ THEN READONLY := TRUE;             COMP     670
            IF FILENAME = ΞOUTPUT    Ξ THEN READONLY := FALSE;            COMP     670
            IF OP = PLUS THEN                                             COMP     670
             BEGIN IF FILENAME = ΞINPUT     Ξ THEN                        COMP     670
                    INPUTPTR↑.IDTYPE := STEXTPTR                          COMP     670
                   ELSE IF FILENAME = ΞOUTPUT    Ξ THEN                   COMP     670
                         OUTPUTPTR↑.IDTYPE := STEXTPTR                    COMP     670
                        ELSE ERROR(6);                                    COMP     670
                   INSYMBOL                                               COMP     670
             END;                                                         COMP     671
            SYSLOC := EXTFILS + 1                                         COMP     671
           END;                                                           COMP     671
          FEXFILP := EXFILP                                               COMP     671
        END                                                               COMP     671
       ELSE ERROR(2);                                                     COMP     671
       TEST2([COMMA,RPARENT],6,FSYS)                                      COMP     671
      UNTIL SY <> COMMA;                                                  COMP     671
      TEST1(RPARENT,4)                                                    COMP     671
```

```
      END;
    TEST1(SEMICOLON,14)
   END
  ELSE BEGIN ERROR(2); SKIP(FSYS) END                                    000084
 END
ELSE BEGIN ERROR(3); SKIP(FSYS) END;
IF OUTPUTPTR = NIL THEN
 BEGIN ERROR(176); OUTPUTPTR := LCP END;
REPEAT BLOCK(FSYS,PERIOD,NIL)
UNTIL SY = PERIOD
END (*PROGRAMME*) ;

PROCEDURE STDTYPENTRIES;
 VAR SP: STP;
BEGIN                                                          (*TYPE*)
                                                               (******)

 NEW(INTPTR,SCALAR,STANDARD);                                   (*INTEGER*)
 WITH INTPTR↑ DO
  BEGIN FORM := SCALAR; SCALKIND := STANDARD; FTYPE := FALSE;
   WITH SIZE DO
    BEGIN WORDS := 1; BITS := 0 END
  END;
 NEW(REALPTR,SCALAR,STANDARD);                                  (*REAL*)
 WITH REALPTR↑ DO
  BEGIN FORM := SCALAR; SCALKIND := STANDARD; FTYPE := FALSE;
   WITH SIZE DO
    BEGIN WORDS := 1; BITS := 0 END
  END;
 NEW(CHARPTR,SCALAR,STANDARD);                                  (*CHAR*)
 WITH CHARPTR↑ DO
  BEGIN FORM := SCALAR; SCALKIND := STANDARD; FTYPE := FALSE;
   WITH SIZE DO
    BEGIN WORDS := 0; BITS := 6 END
  END;
 NEW(BOOLPTR,SCALAR,DECLARED);                                  (*BOOLEAN*)
 WITH BOOLPTR↑ DO
  BEGIN FORM := SCALAR; SCALKIND := DECLARED; FTYPE := FALSE;
   WITH SIZE DO
    BEGIN WORDS := 0; BITS := 1 END;
  END;
 NEW(NILPTR,POINTER);                                           (*NIL*)
 WITH NILPTR↑ DO
  BEGIN ELTYPE := NIL; FORM := POINTER; FTYPE := FALSE;
  WITH SIZE DO
   BEGIN WORDS := 0; BITS := 18 END
  END;
 NEW(TEXTPTR,FILES);                                            (*TEXT*)
 WITH TEXTPTR↑ DO
  BEGIN FILTYPE := CHARPTR; PCKDFIL := TRUE; FORM := FILES;
   TEXTFILE := TRUE; SEGFILE := FALSE; FTYPE := TRUE;
   WITH SIZE DO
    BEGIN WORDS := 128*BUFFAC + 1 + CHEFETSZ; BITS := 0 END
  END;
 NEW(STEXTPTR,FILES);
 WITH STEXTPTR↑ DO
  BEGIN FILTYPE := CHARPTR; PCKDFIL := TRUE; FORM := FILES;
   TEXTFILE := TRUE; SEGFILE := TRUE; FTYPE := TRUE;
   WITH SIZE DO
    BEGIN WORDS := 128*BUFFAC + 1 + CHEFETSZ; BITS := 0 END
  END;
 NEW(SP,SUBRANGE);                                              (*ALFA*)
 WITH SP↑ DO
  BEGIN FORM := SUBRANGE; RANGETYPE := INTPTR; FTYPE := FALSE;
   MIN.IVAL := 1; MAX.IVAL := ALFALENG;
   WITH SIZE DO
    BEGIN WORDS := 0; BITS := 4 END;
  END;
 NEW(ALFAPTR,ARRAYS);
 WITH ALFAPTR↑ DO
  BEGIN FORM := ARRAYS; FTYPE := FALSE;
   AELTYPE := CHARPTR; INXTYPE := SP;
   PCKDARR := TRUE; PARTWORDELS := TRUE; ELSPERWORD := ALFALENG;
   WITH SIZE DO
    BEGIN WORDS := 1; BITS := 0 END
  END
END (*STDTYPENTRIES*);

PROCEDURE STDNAMENTRIES;
 VAR CP,CP1: CTP; I,LCNT: INTEGER;
     NA : ARRAY[1..52] OF ALFA;
```

COMP 671 … COMP 679 … COMP 679
MSURAND

```
BEGIN                                                         (*NAME:*)
                                                              (********)

 NEW(CP,TYPES);                                               (*INTEGER*)
 WITH CP↑ DO
  BEGIN NAME := ≡INTEGER   ≡; IDTYPE := INTPTR; KLASS := TYPES END;
 ENTERID(CP);
 NEW(CP,TYPES);                                               (*REAL*)
 WITH CP↑ DO
  BEGIN NAME := ≡REAL      ≡; IDTYPE := REALPTR; KLASS := TYPES END;
 ENTERID(CP);
 NEW(CP,TYPES);                                               (*CHAR*)
 WITH CP↑ DO
  BEGIN NAME := ≡CHAR      ≡; IDTYPE := CHARPTR; KLASS := TYPES END;
 ENTERID(CP);
 NEW(CP,TYPES);                                               (*BOOLEAN*)
 WITH CP↑ DO
  BEGIN NAME := ≡BOOLEAN   ≡; IDTYPE := BOOLPTR; KLASS := TYPES END;
 ENTERID(CP);
 NEW(CP,TYPES);                                               (*TEXT*)
 WITH CP↑ DO
  BEGIN NAME := ≡TEXT      ≡; IDTYPE := TEXTPTR; KLASS := TYPES END;
 ENTERID(CP);
 NEW(CP,TYPES);                                               (*ALFA*)
 WITH CP↑ DO
  BEGIN NAME := ≡ALFA      ≡; IDTYPE := ALFAPTR; KLASS := TYPES END;
 ENTERID(CP);
 NEW(CP,KONST);                                               (*NIL*)
 WITH CP↑ DO
  BEGIN NAME := ≡NIL       ≡; IDTYPE := NILPTR; KLASS := KONST;
   NEXT := NIL; VALUES.IVAL := 377777B;
  END;
 ENTERID(CP);
 NEW(CP,KONST);                                               (*COLON*)
 WITH CP↑ DO
  BEGIN NAME := ≡COL       ≡; IDTYPE := CHARPTR; KLASS := KONST;
   NEXT := NIL; VALUES.IVAL := 0
  END;
 ENTERID(CP);
 NEW(CP,KONST);                                               (*MAXINT*)
 WITH CP↑ DO
  BEGIN NAME := ≡MAXINT    ≡; IDTYPE := INTPTR; KLASS := KONST;
   NEXT := NIL; VALUES.IVAL := 7777777777777777B;
  END;
 ENTERID(CP);
 NA[ 1] := ≡FALSE     ≡;NA[ 2] := ≡TRUE      ≡;NA[ 3] := ≡OUTPUT    ≡;
 NA[ 4] := ≡INPUT     ≡;NA[ 5] := ≡GET       ≡;NA[ 6] := ≡PUT       ≡;
 NA[ 7] := ≡RESET     ≡;NA[ 8] := ≡REWRITE   ≡;NA[ 9] := ≡GETSEG    ≡;
 NA[10] := ≡PUTSEG    ≡;NA[11] := ≡LINELIMIT ≡;NA[12] := ≡READ      ≡;
 NA[13] := ≡READLN    ≡;NA[14] := ≡WRITE     ≡;NA[15] := ≡WRITELN   ≡;
 NA[16] := ≡MESSAGE   ≡;NA[17] := ≡PAGE      ≡;NA[18] := ≡TIME      ≡;
 NA[19] := ≡DATE      ≡;NA[20] := ≡HALT      ≡;NA[21] := ≡PACK      ≡;
 NA[22] := ≡UNPACK    ≡;NA[23] := ≡NEW       ≡;NA[24] := ≡DISPOSE   ≡;
 NA[25] := ≡RELEASE   ≡;NA[26] := ≡*DUMMY2   ≡;NA[27] := ≡*DUMMY3   ≡;
 NA[28] := ≡EOF       ≡;NA[29] := ≡EOS       ≡;NA[30] := ≡EOLN      ≡;
 NA[31] := ≡ODD       ≡;NA[32] := ≡UNDEFINED ≡;NA[33] := ≡ROUND     ≡;
 NA[34] := ≡TRUNC     ≡;NA[35] := ≡EXPO      ≡;NA[36] := ≡ABS       ≡;
 NA[37] := ≡SQR       ≡;NA[38] := ≡ORD       ≡;NA[39] := ≡CHR       ≡;
 NA[40] := ≡PRED      ≡;NA[41] := ≡SUCC      ≡;NA[42] := ≡CARD      ≡;
 NA[43] := ≡CLOCK     ≡;NA[44] := ≡*DUMMY4   ≡;NA[45] := ≡*DUMMY5   ≡;
 NA[46] := ≡SIN       ≡;NA[47] := ≡COS       ≡;NA[48] := ≡EXP       ≡;
 NA[49] := ≡SQRT      ≡;NA[50] := ≡LN        ≡;NA[51] := ≡ARCTAN    ≡;
 NA[52] := ≡RANDOM    ≡;
 CP1 := NIL; LCNT := 0;
 FOR I := 1 TO 2 DO
  BEGIN NEW(CP,KONST);                                        (*FALSE,TRUE*)
   WITH CP↑ DO
    BEGIN NAME := NA[LCNT+I]; IDTYPE := BOOLPTR; KLASS := KONST;
     NEXT := CP1; VALUES.IVAL := I - 1
    END;
   ENTERID(CP); CP1 := CP
  END;
 LCNT := LCNT + 4;
 BOOLPTR↑.FCONST := CP;
 FOR I := 1 TO NRSTDPROC DO
  BEGIN NEW(CP,PROC,STANDARD);                     (*GET,PUT,RESET*)
   WITH CP↑ DO                                     (*REWRITE,GETSEG*)
    BEGIN NAME := NA[LCNT+I]; IDTYPE := NIL;       (*PUTSEG,LINELIMIT*)
     NEXT := NIL; KEY := I;                        (*READ,READLN*)
     KLASS := PROC; PFDECKIND := STANDARD          (*WRITE,WRITELN*)
    END;                                           (*MESSAGE,PAGE*)
```

```
   ENTERID(CP)                                     (*TIME,DATE,HALT*)
  END;                                             (*PACK,UNPACK,NEW*)
 LCNT := LCNT + NRSTDPROC;                         (*DISPOSE,RELEASE*)
 FOR I := 1 TO NRSTDFUNC DO                        (*EOF,EOS,EOLN,ODD*)
  BEGIN NEW(CP,FUNC,STANDARD);                     (*UNDEFINED,ROUND*)
   WITH CP↑ DO                                     (*TRUNC,EXPO,ABS*)
    BEGIN NAME:=NA[LCNT+I]; IDTYPE:=NILPTR;     (*SQR,ORD,CHR,PRED *)
     KLASS := FUNC; PFDECKIND := STANDARD;         (*SUCC,CARD,CLOCK*)
      NEXT := NIL; KEY := I;
   END;
   ENTERID(CP)
  END;
 LCNT := LCNT + NRSTDFUNC;
 FOR I := 1 TO NREXFUNC DO                         (*SIN,COS,EXP*)
  BEGIN NEW(CP,FUNC,STANDARD);                     (*SQRT,LN,ARCTAN*)
   WITH CP↑ DO
    BEGIN NAME := NA[LCNT+I]; IDTYPE := REALPTR;
     KLASS := FUNC; PFDECKIND := STANDARD;
     NEXT := NIL; KEY := NRSTDFUNC + I
    END;
   ENTERID(CP)
  END;
END (*STDNAMENTRIES*);

PROCEDURE ENTERUNDECL;
BEGIN
 NEW(UTYPPTR,TYPES);
 WITH UTYPPTR↑ DO
  BEGIN NAME := ≡           ≡; IDTYPE := NIL; KLASS := TYPES END;
 NEW(UCSTPTR,KONST);
 WITH UCSTPTR↑ DO
  BEGIN NAME := ≡           ≡; IDTYPE := NIL; NEXT := NIL;
   VALUES.IVAL := 0; KLASS := KONST;
  END;
 NEW(UVARPTR,VARS);
 WITH UVARPTR↑ DO
  BEGIN NAME := ≡           ≡; IDTYPE := NIL; VKIND := DRCT;
   NEXT := NIL; VLEV := 0; VADDR := 0; KLASS := VARS
  END;
 NEW(UFLDPTR,FIELD);
 WITH UFLDPTR↑ DO
  BEGIN NAME := ≡           ≡; IDTYPE := NIL; NEXT := NIL; FLDADDR := 0;
   KLASS := FIELD
  END;
 NEW(UPRCPTR,PROC,DECLARED,ACTUAL);
 WITH UPRCPTR↑ DO
  BEGIN NAME := ≡           ≡; IDTYPE := NIL;
   KLASS := PROC; PFDECKIND := DECLARED; PFKIND := ACTUAL;
   PFXCPT:=4;
   NEXT := NIL; PFDECL := DECL; PFLEV := 0; PFADDR := 0
  END;
 NEW(UFCTPTR,FUNC,DECLARED,ACTUAL);
 WITH UFCTPTR↑ DO
  BEGIN NAME := ≡           ≡; IDTYPE := NIL; NEXT := NIL;
   KLASS := FUNC; PFDECKIND := DECLARED; PFKIND := ACTUAL;
   PFXCPT:=4;
   PFDECL := DECL; PFLEV := 0; PFADDR := 0
  END
END (*ENTERUNDECL*) ;

PROCEDURE INITSCALARS;
BEGIN FWPTR := NIL; FSTLABP := NIL; FSTCSP := NIL; FILECNT := 0;
  FSTPCRP := NIL;
 INPUTPTR := NIL; OUTPUTPTR := NIL;
 LABCNT := 0; ERRORS := FALSE; B6DPL := 3;
 DEBUG := TRUE; EXTON := FALSE; LISTON := TRUE; PMD := TRUE;
 DP := TRUE; PRTERR := TRUE; ERRINX := 0;
XPARMAX := 4;
 BUFFAC := 1;    (*WILL NOT ALLOW INPUT AND OUTPUT TO BE TAPE FILES*)
                  (*ANY MORE, BUT WILL REDUCE PROGRAM SIZE BY 1400B*)
 IC := 0; CODEADDR := 0; PCNT := 0;
 LC := 6; (*NEVER TOUCH THIS INITIALISATION; 3-6 ARE RESERVED FOR
        FUTURE SYSTEM USE*)
 MAXCHCNT := 72;                                                        MSUDF72
END (*INITSCALARS*) ;

PROCEDURE INITSETS;
BEGIN
 DIGITS := [≡0≡..≡9≡]; LETTERS := [≡A≡..≡Z≡];
 CONSTBEGSYS := [ADDOP,INTCONST,REALCONST,CHARCONST,STRINGCONST,IDENT];
 SIMPTYPEBEGSYS := [LPARENT]+CONSTBEGSYS;
```

```
  TYPEBEGSYS := [ARROW,PACKEDSY,SEGMENTEDSY,ARRAYSY,RECORDSY,SETSY,
        FILESY]+SIMPTYPEBEGSYS;
  TYPEDELS := [ARRAYSY,RECORDSY,SETSY,FILESY];
  BLOCKBEGSYS := [LABELSY,CONSTSY,TYPESY,VARSY,PROCEDURESY,FUNCTIONSY,
        BEGINSY];
  SELECTSYS := [ARROW,PERIOD,LBRACK];
  FACBEGSYS := [INTCONST,REALCONST,CHARCONST,STRINGCONST,IDENT,LPARENT,
        LBRACK,NOTSY];
  STATBEGSYS := [BEGINSY,GOTOSY,IFSY,WHILESY,REPEATSY,FORSY,WITHSY,
        CASESY];
END (*INITSETS*) ;

PROCEDURE INITTABLES;                                          000087
  VAR I: 1..ALFALENG; CH1: CHAR;
  PROCEDURE RESERVEDWORDS;
  BEGIN
    RW[ 1] := 'IF       ';RW[ 2] := 'DO       ';RW[ 3] := 'OF       ';
    RW[ 4] := 'TO       ';RW[ 5] := 'IN       ';RW[ 6] := 'OR       ';
    RW[ 7] := 'END      ';RW[ 8] := 'FOR      ';RW[ 9] := 'VAR      ';
    RW[10] := 'DIV      ';RW[11] := 'MOD      ';RW[12] := 'SET      ';
    RW[13] := 'AND      ';RW[14] := 'NOT      ';RW[15] := 'THEN     ';
    RW[16] := 'ELSE     ';RW[17] := 'WITH     ';RW[18] := 'GOTO     ';
    RW[19] := 'CASE     ';RW[20] := 'TYPE     ';RW[21] := 'FILE     ';
    RW[22] := 'BEGIN    ';RW[23] := 'UNTIL    ';RW[24] := 'WHILE    ';
    RW[25] := 'ARRAY    ';RW[26] := 'CONST    ';RW[27] := 'LABEL    ';
    RW[28] := 'REPEAT   ';RW[29] := 'RECORD   ';RW[30] := 'DOWNTO   ';
    RW[31] := 'PACKED   ';RW[32] := 'PROGRAM  ';RW[33] := 'FUNCTION ';
    RW[34] := 'PROCEDURE';RW[35] := 'SEGMENTED';
    LRW[0] :=  0; LRW[1] :=  0; LRW[2] :=  6; LRW[3] := 14; LRW[4] := 21;
    LRW[5] := 27; LRW[6] := 31; LRW[7] := 32; LRW[8] := 33; LRW[9] := 35;
    LRW[10] := 35;
  END (*RESWORDS*) ;

  PROCEDURE SYMBOLS;
  BEGIN
    RSY[1] := IFSY; RSY[2] := DOSY; RSY[3] := OFSY; RSY[4] := TOSY;
    RSY[5] := RELOP; RSY[6] := ADDOP; RSY[7] := ENDSY; RSY[8] := FORSY;
    RSY[9] := VARSY; RSY[10] := MULOP;RSY[11] := MULOP; RSY[12] := SETSY;
    RSY[13] := MULOP; RSY[14] := NOTSY; RSY[15] := THENSY;
    RSY[16] := ELSESY; RSY[17] := WITHSY; RSY[18] := GOTOSY;
    RSY[19] := CASESY; RSY[20] := TYPESY; RSY[21] := FILESY;
    RSY[22] := BEGINSY; RSY[23] := UNTILSY; RSY[24] := WHILESY;
    RSY[25] := ARRAYSY; RSY[26] := CONSTSY; RSY[27] := LABELSY;
    RSY[28] := REPEATSY; RSY[29] := RECORDSY; RSY[30] := DOWNTOSY;
    RSY[31] := PACKEDSY; RSY[32] := PROGRAMSY; RSY[33] := FUNCTIONSY;
    RSY[34] := PROCEDURESY; RSY[35] := SEGMENTEDSY;
    SSY['+'] := ADDOP; SSY['-'] := ADDOP; SSY['*'] := MULOP;
    SSY['/'] := MULOP; SSY['('] := LPARENT; SSY[')'] := RPARENT;
    SSY['$'] := OTHERSY; SSY['='] := RELOP; SSY[' '] := OTHERSY;
    SSY[','] := COMMA; SSY['.'] := PERIOD; SSY['''] := OTHERSY;
    SSY['['] := LBRACK; SSY[']'] := RBRACK; SSY[':'] := COLON;
    SSY['≠'] := RELOP; SSY['↑'] := OTHERSY;
    SSY['∨'] := ADDOP; SSY['^'] := MULOP;
    SSY['↑'] := ARROW; SSY['↓'] := OTHERSY;
    SSY['<'] := RELOP; SSY['>'] := RELOP;
    SSY['≤'] := RELOP; SSY['≥'] := RELOP;
    SSY['¬'] := NOTSY; SSY[';'] := SEMICOLON;
  END (*SYMBOLS*) ;

  PROCEDURE OPERATORS;
    VAR I: INTEGER; CH: CHAR;
  BEGIN
    FOR I := 1 TO RESWORDS DO ROP[I] := NOOP;
    ROP[5] := INOP; ROP[10] := IDIV; ROP[11] := IMOD;
    ROP[6] := OROP; ROP[13] := ANDOP;
    FOR CH := '+' TO ';' DO SOP[CH] := NOOP;
    SOP['+'] := PLUS;SOP['-'] := MINUS;SOP['*'] := MUL;SOP['/'] := RDIV;
    SOP['='] := EQOP;SOP['≠'] := NEOP;SOP['∨'] := OROP;SOP['^'] := ANDOP;
    SOP['<'] := LTOP;SOP['>'] := GTOP;SOP['≤'] := LEOP;SOP['≥'] := GEOP;
  END (*OPERATORS*) ;

BEGIN (*INITTABLES*)
  RESERVEDWORDS; SYMBOLS; OPERATORS;
  PASCF := 'PASCFL    ';
  FOR I := 7 TO ALFALENG DO PASCF[I] := CHR(0);
  PASCL := PASCF; PASCL[5] := 'L'; PASCL[6] := '.';
  PNAME := 'PRC       ';
  FOR I := 1 TO ERRMAX DO ERLIST[I] := FALSE;
  FOR I:= 8 TO ALFALENG DO PNAME[I] := CHR(0)
END (*INITTABLES*) ;
```

```
  PROCEDURE PASCERR(VAR F: TEXT; EL: ERLISTT); EXTERN;                                  COMP      704
                                                                                        COMP      704
BEGIN                                                                                   COMP      704
                                                                                        COMP      704
 (*INITIALIZE:*)                                                                 000088 COMP      704
 (*************)                                                                        COMP      704
 INITSCALARS; INITSETS; INITTABLES;                                                     COMP      704
 LINELIMIT(OUTPUT,-1); (* UNLIMITED OUTPUT *)                                           COMP      704
 DATE(CURDATE); TIME(CURTIME);                                                          MSUTITLE
 WRITE(=T=);                                                                            MSUTITLE
 WRITE(TITLEMSG,SUBVERSION,= =:20,CURDATE,CURTIME);                                     MSUTITLE
 ENDOFLINE;                                                                             COMP      705
 CH := INPUT^; CHCNT := 1; IF LISTON THEN WRITE(CH);                                    COMP      705
                                                                                        COMP      705
 (*ENTER STANDARD NAMES AND STANDARD TYPES:*)                                           COMP      705
 (*****************************************)                                            COMP      705
                                                                                        COMP      705
 LEVEL := 0; TOP := 0;                                                                  COMP      706
 WITH DISPLAY[0] DO                                                                     COMP      706
  BEGIN FNAME := NIL; OCCUR := BLCK END;                                                COMP      706
 STDTYPENTRIES; STDNAMENTRIES; ENTERUNDECL;                                             COMP      706
 TOP := 1; LEVEL := 1;                                                                  COMP      706
 WITH DISPLAY[1] DO                                                                     COMP      706
  BEGIN FNAME := NIL; OCCUR := BLCK END;                                                COMP      706
 MESSAGE(=- PASCAL VERSION 3.00.01 10/22/76=);                                          MSUTITLE
                                                                                        COMP      706
 (*COMPILE:*)                                                                          COMP      706
 (**********)                                                                          COMP      706
                                                                                        COMP      706
 INSYMBOL;                                                                              COMP      706
 PROGRAMME(BLOCKBEGSYS+STATBEGSYS-[CASESY]);                                            COMP      706
 WHILE NOT (EOS(INPUT) OR EOLN(INPUT)) DO                                               COMP      707
  BEGIN GET(INPUT); IF LISTON THEN WRITE(INPUT^) END;                                   COMP      707
 IF LISTON THEN WRITELN;                                                                COMP      707
 IF ERRINX > 0 THEN WRITEERRORS;                                                        COMP      707
 IF NOT EOS(INPUT) THEN READLN(INPUT);                                                  COMP      707
 WHILE NOT EOS(INPUT) DO                                                                COMP      707
  BEGIN                                                                                 COMP      707
  IF LISTON THEN WRITE(=         =);                                                    COMP      707
  WHILE NOT EOLN(INPUT) DO                                                              COMP      707
   BEGIN IF LISTON THEN WRITE(INPUT^); GET(INPUT) END;                                  COMP      707
  IF LISTON THEN WRITELN; IF NOT EOS(INPUT) THEN READLN;                                COMP      708
  END;                                                                                  COMP      708
13:                                                                                     COMP      708
 IF ERRORS THEN PASCERR(OUTPUT, ERLIST);                                                COMP      708
 WRITELN; WRITELN;                                                                      COMP      708
END .                                                                                   COMP      708
```