MMARY OF UPDATE IDENTIFIERS WITHIN DECK - YANK$$$

000001

IDENTIFIER	TOTAL	ACTIVE

LIST OF CONTROL, ACTIVE, AND/OR INACTIVE CARDS IN PASCSYM

10/30/7

```
*DECK PASCSYM                                                          PAS
      ↱**************************************************             PAS
       *                                                *             PAS
       *          P A S C A L   C O M P I L E R         *             PAS
       *          ****************************          *             PAS
       *                                                *             PAS
       *                    20/06/72                    *             PAS
       *                                                *             PAS
       *  AUTHORS:  U. AMMANN,  R. SCHILD               *             PAS
       *                                                *             PAS
       *             FACHGRUPPE COMPUTERWISSENSCHAFTEN  *             PAS
       *             EIDG. TECHNISCHE HOCHSCHULE        *             PAS
       *    CH-8006 ZUERICH                             *             PAS
       *                                                *             PAS
       **************************************************↓            PAS
                                                                       PAS
      ↱$T-,O+,R+   COMPILE WITH STACK OVERFLOW TEST AND                 PAS
                   ROUNDING INSTRUCTIONS↓                              PAS
                                                                       PAS
   CONST  NOP = 46000B, NILVAL = 200000B,                              PAS
      JMPCST = 6165000000040000000B, EQJMP = 0400000000460004600B,     PAS
      MAX10 = 3146314631463146313B, MASK = -777777B,                   PAS
      MAXLABS = 20, MAXEXLABS = 10, JMPMAX = 49,                       PAS
      CSTMAX = 40, CODMAX =  1000, UNDMAX = 70,                        PAS
      DISPLIMIT = 20, PTLIMIT = 10, FILLIMIT = 10,                     PAS
      MAXLEVEL = 7,                                                    PAS
      ↱ DONΞT INCREASE WITHOUT INCREASING UPPER BOUND OF RG3 TOO↓       PAS
      WORDLENGTH = 60,                                                 PAS
                                                                       PAS
      PROCDR = -1,  OBJ = +1,  ↱FOR PMDFILE↓                           PAS
                                                                       PAS
      GETC = 101B, PUTC = 103B,                                        PAS
   ↱ADDRESSES OF RESIDENT STANDARD ROUTINES↓                           PAS
                                                                       PAS
      ASSERR = 170B, INXERR = 171B, DIVERR = 172B, STOFLERR = 174B,    PAS
   ↱RUNTIME ERROR EXITS↓                                               PAS
                                                                       PAS
      INPT = 212B, OUTPT = 233B;                                       PAS
   ↱ ADDRESSES OF STANDARD FILES INPUT AND OUTPUT ↓                    PAS
                                                                       PAS
   TYPE CTP = ↑CONTEXTTABLE;                                           PAS
      CWORD = ARRAY [1..10] OF CHAR;                                   PAS
      ADDRESS = 0..177777B ;                                           PAS
      SHRTINT = -377777B .. +377777B;                                  PAS
```

WIRTH (FROM JIM WHITE RALPH PARSONS+Co)

LIST OF CONTROL, ACTIVE, AND/OR INACTIVE CARDS IN PASCSYM 000002

```
      BITRANGE = 0 .. 60;  RG3 = 0 .. 7;                                    PA
      ATTRKIND = (VARBL,SVAL,LVAL,LCOND);                                   PA
      ATTR = RECORD TYPTR : CTP ;                                           PA
               CASE KIND : ATTRKIND OF                                      PA
                  VARBL: (ACCESS : (DRCT,INDRCT,INXD) ;                     PA
                          BREG : RG3 ; DPLMT : SHRTINT ;                    PA
                          CASE PCKD : BOOLEAN OF                            PA
                             FALSE: ;                                       PA
                             TRUE: (BITADR, BITSZ : BITRANGE)) ;            PA
                    SVAL:  (VAL : SHRTINT) ;                                PA
                    LVAL:  (CTERM : SHRTINT) ;                              PA
                    LCOND: (JMP : 0 .. 3 ; ARITH : BOOLEAN)                 PA
            END ;                                                           PA
      ↱DESCRIBES EXPRESSIONS TO BE COMPILED.    EXAMPLES:                   PA
          VARBL:          DRCT:                 I              **)          PA
                          INDRCT:               INPUT↑         *)           PA
                          INXD:                 A[K]           *)           PA
          SVAL:                                 4              **)          PA
          LVAL:                                 2*I - 1        *)           PA
          LCOND:          ARITH:                X ≥ 4.1        *)           PA
                         ¬ARITH:                B < TRUE       *)           PA
                                                                            PA
          *) RESULT OF CODEGENERATION IN REGISTER X-RP                      PA
         **) NO CODE GENERATION UNTIL NOW↓                                  PA
                                                                            PA
                                                                            PA
      OPTPWR = (NOOPT,PUREP,POSP,NEGP);                                     PA
      IDCLASS = (TYPES,KONST,PROC,VARS,FIELD,TAGFIELD,DUMMYCLASS);          PA
      TYPFORM = (NUMERIC,SYMBOLIC,POINTER,POWER,                            PA
                 ARRAYS,RECORDS,CLASSS,FILES);                              PA
      IDKINDS = (ACTUAL,FORMAL);                                            PA
      WHERE = (BLOCK,CWITH,VWITH);                                          PA
      PKINDS = PROCDR .. OBJ; ↱FOR PMDFILE↓                                 PA
                                                                            PA
    VAR PARMLIST : RECORD   ICO : ADDRESS ;                                 PA
                            ERRFLAG : BOOLEAN ;                             PA
                            ERRNRS : ARRAY [0..3] OF POWERSET 0..31 ;       PA
                            LOADPT : ADDRESS ;                              PA
                            LIMCODE : ADDRESS ;                             PA
                            LPJMPTAB : ADDRESS ;                            PA
                            EXTFLAGS : POWERSET 0..4 ;                      PA
                            ENTRYPT : ADDRESS                               PA
                    END ;                                                   PA
      JMPTAB : ARRAY [0..JMPMAX] OF INTEGER;                                PA
      JMPIX : SHRTINT;                                                      PA
      ↱TRANSFER VECTOR FOR CALLS OF FORWARD DECLARED PROCEDURES AND         PA
       GOTO STATEMENTS LEADING OUT OF PROCEDURES↓                           PA
                                                                            PA
      CSTTB : ARRAY [1..CSTMAX] OF                                          PA
         RECORD VALU : INTEGER; INX : SHRTINT  END;                         PA
      LCX : SHRTINT;                                                        PA
      ↱CONTAINS ALL CONSTANTS C, ABS(INT(C)) ≥ 2**17, OCCURRING IN THE      PA
```

LIST OF CONTROL, ACTIVE, AND/OR INACTIVE CARDS IN PASCSYM 000003

```
      PROCEDURE ACTUALLY BEING COMPILED TOGETHER WITH AN INDEX INTO          PA
      UNDLAB WHERE THEIR OCCURRENCES IN THE CODE OF THIS PROCEDURE           PA
      ARE CHAINED↓                                                          PA
                                                                            PA
    PILEV,FILEV : ARRAY [0..MAXLEVEL] OF SHRTINT;                           PA
    PFL : ARRAY[0..FILLIMIT] OF ADDRESS;                                    PA
    FILPTS : ARRAY [0..FILLIMIT] OF CTP;                                    PA
    PFTOP,FILTOP : SHRTINT;                                                 PA
    ↱PFL CONTAINS THE ADDRESSES OF LOCAL CLASSES,                           PA
       PILEV CONTAINS POINTERS INTO PFL, PFTOP = TOP OF PFL.                PA
      FILPTS CONTAINS POINTERS TO ENTRIES FOR LOCAL FILES,                  PA
        FILEV CONTAINS POINTERS INTO FILPTS, FILTOP = TOP OF FILPTS↓        PA
                                                                            PA
    EXTAB : ARRAY [1..MAXEXLABS] OF                                         PA
       PACKED RECORD EXVAL,JMPTABIX : SHRTINT  END;                         PA
    CEXTABIX : SHRTINT;                                                     PA
    ↱CONTAINS THE EXPLICITLY DECLARED LABELS OF ALL PROCEDURES              PA
      NOT YET CLOSED, TOGETHER WITH THEIR CORRESPONDING INDEX               PA
      INTO JMPTAB↓                                                          PA
                                                                            PA
    LABTAB : ARRAY [1..MAXLABS] OF                                          PA
       PACKED RECORD LABVAL,FLD2,FLD3 : SHRTINT  END;                       PA
    CLABIX : SHRTINT;                                                       PA
    ↱CONTAINS ALL LABELS MET SO FAR IN THE BODY OF THE PROCEDURE            PA
      ACTUALLY BEING COMPILED, TOGETHER WITH INFORMATION WHETHER            PA
      LABEL DEFINITION (<LABEL>:) ALREADY FOUND OR NOT (FLD2).              PA
      IN THE FORMER CASE FLD3 CONTAINS THE CORRESPONDING ADDRESS,           PA
      WHERE IN THE LATTER CASE FLD3 CONTAINS AN INDEX INTO UNDLAB           PA
      WHERE THE OCCURRENCES ARE CHAINED↓                                    PA
                                                                            PA
    UNDLAB : ARRAY [1..UNDMAX] OF                                           PA
       PACKED RECORD SUCC, PLACE : SHRTINT ;                                PA
                     LFTSH : BITRANGE ;                                     PA
              END ;                                                         PA
    CHNIX : SHRTINT;                                                        PA
    ↱ACTS AS LISTSTRUCTURE, CHAINING OCCURRENCES OF CONSTANTS AND           PA
      JUMPS TO NOT YET REACHED LABELS IN THE CODE OF THE PROCEDURE          PA
      ACTUALLY BEING COMPILED.                                              PA
      CHNIX = HEAD OF FREE LIST↓                                            PA
                                                                            PA
    PTLIST : ARRAY [0..PTLIMIT] OF                                          PA
       RECORD HNAME : ALFA;  PPTR : CTP  END;                               PA
    PTX : SHRTINT;                                                          PA
    ↱PTLIST CONTAINS NAMES OF YET UNDECLARED CLASSES AND                    PA
       POINTERS TO THE CORRESPONDING POINTER-TYPE ENTRIES                   PA
      PTX = TOP OF PTLIST ↓                                                 PA
                                                                            PA
    ERRLIST : ARRAY [1..10] OF                                              PA
       PACKED RECORD POS,NMR : SHRTINT  END;                                PA
    ERRINX,CHCNT : SHRTINT;  EOLFLAG,ERR : BOOLEAN;                         PA
    ↱ERRLIST CONTAINS THE POSITIONS AND NUMBERS OF THE ERRORS               PA
       ON ONE LINE                                                          PA
```

LIST OF CONTROL, ACTIVE, AND/OR INACTIVE CARDS IN PASCSYM 000004

```
      ERRINX = TOP OF ERRLIST                                                  PAS
      CHCNT = POSITION OF LAST READ CHARACTER                                  PAS
     ERRLIST,ERRINX,CHCNT,EOLFLAG ARE USED BY NEXTCH                           PAS
      AND ERROR.                                                               PAS
     ERR IS SET WHENEVER ERROR HAS BEEN CALLED AND IS TESTED                   PAS
      (AND RESET) BY SEVERAL PROCEDURES ↓                                      PAS
                                                                               PAS
    DISPLAY : ARRAY [0..DISPLIMIT] OF                                          PAS
    PACKED RECORD FNAME : CTP;                                                 PAS
        CASE OCCUR : WHERE OF                                                  PAS
          BLOCK : ;                                                            PAS
          CWITH : (CDSPL : SHRTINT; CLEV : RG3);                               PAS
          VWITH : (VDSPL : SHRTINT);                                           PAS
    END;                                                                       PAS
    TOP,DISX : SHRTINT;                                                        PAS
    ↱TOP = TOP OF DISPLAY, DISX IS RETURNED BY SEARCH↓                         PAS
                                                                               PAS
    CODE : ARRAY [0..CODMAX] OF INTEGER;                                       PAS
    CA,CP : SHRTINT;  BUF : INTEGER;                                           PAS
    PASCLGO[OUT] : FILE OF INTEGER;                                            PAS
    ↱CODE IS WRITTEN ONTO PASCLGO.                                             PAS
     BUF IS THE CURRENT ELEMENT OF CODE.                                       PAS
     CA = INDEX TO CODE,  CP = NUMBER OF QUARTER-WORDS USED ↓                  PAS
    LASTOP, LASTI : SHRTINT ;   ↱TO AVOID B XI XJ - INSTRUCTIONS↓              PAS
                                                                               PAS
    PMDFILE[OUT]: FILE 1 OF                                                    PAS
      PACKED RECORD PNAME : ALFA;                                              PAS
        CASE PKIND : PKINDS OF                                                 PAS
        OBJ: (ACCESS,TYP,RLA : SHRTINT);                                       PAS
        PROCDR: (LINK,SA,COUNT : SHRTINT);                                     PAS
      END;                                                                     PAS
    PMCTR,LASTLINK : SHRTINT;                                                  PAS
                                                                               PAS
    LC,IC : ADDRESS;                                                           PAS
    ↱LOCATION COUNTER AND INSTRUCTION COUNTER↓                                 PAS
                                                                               PAS
    LEVEL,RP,RP1 : RG3;                                                        PAS
    ↱RP,RP1 = REGISTER-POINTERS (STACK OF X-REGS)↓                             PAS
    DP : BOOLEAN ;                                                             PAS
    ↱DP = TRUE : DECLARATION PART (USED BY NEXTCH)↓                            PAS
                                                                               PAS
    PRCODE,ASSCHECK,INXCHECK,DIVCHECK,STOFLCHECK,                              PAS
    ROUNDING : BOOLEAN ;   ↱COMPILER OPTIONS↓                                  PAS
                                                                               PAS
    GATTR : ATTR;   ↱GLOBAL ATTRIBUTE RECORD ↓                                 PAS
                                                                               PAS
    CONTEXTTABLE : CLASS 450 OF                                                PAS
    PACKED RECORD                                                              PAS
    NAME : ALFA;  NXTEL : CTP;                                                 PAS
      CASE KLASS : IDCLASS OF                                                  PAS
      TYPES : (SIZE : SHRTINT;                                                 PAS
        CASE FORM : TYPFORM OF                                                 PAS
```

LIST OF CONTROL, ACTIVE, AND/OR INACTIVE CARDS IN PASCSYM 000005

```
            NUMERIC : (BITS : BITRANGE; MIN,MAX : INTEGER);                    PAS
            SYMBOLIC : (FCONST,PWSET : CTP; BITSIZE : BITRANGE);               PAS
            POINTER : (DOMAIN,ELTYPE : CTP);                                   PAS
            POWER : (ELSET : CTP; PWBITS : BITRANGE);                          PAS
            ARRAYS : (AELTYPE,INXTYPE : CTP;  LO,HI : SHRTINT;                 PAS
               OPTTYP : OPTPWR; EXP1,EXP2 : BITRANGE);                         PAS
            RECORDS : (FSTFLD,RECVAR : CTP);                                   PAS
            CLASSS : (PELTYPE : CTP);                                          PAS
            FILES : (FELTYPE : CTP)    );                                      PAS
         KONST : (CONTYPE : CTP;                                               PAS
            CASE CONKIND : IDKINDS OF                                          PAS
            ACTUAL : (SUCC : CTP; VALUES : INTEGER);                           PAS
            FORMAL : (CADDR : ADDRESS; CLEVEL : RG3)    );                     PAS
         PROC : (PROCTYPE,FORMALS : CTP;                                       PAS
            PROCKIND : IDKINDS;                                                PAS
            PROCADDR : ADDRESS; PROCLEVEL : RG3;                               PAS
            SEGSIZE : SHRTINT);                                                PAS
         VARS : (VTYPE : CTP; VKIND : IDKINDS;                                 PAS
            VADDR : ADDRESS; VLEVEL : RG3);                                    PAS
         FIELD : (FLDTYPE : CTP; FLDADDR : ADDRESS;                            PAS
            BITDISPL,BITWIDTH : BITRANGE);                                     PAS
         TAGFIELD : (CASESIZE : SHRTINT; VARIANTS : CTP;                       PAS
            CASE TAGVAL : BOOLEAN OF                                           PAS
            FALSE : (CASETYPE : CTP);                                          PAS
            TRUE : (CASEVAL : INTEGER)    );                                   PAS
      END;                                                                     PAS
                                                                               PAS
      INTPTR,REALPTR,ALFAPTR,CHARPTR,BOOLPTR,PREDEFP,NILPTR,                   PAS
         UNDECPTR,PNUMPTR,EXTPTR,LAMPTR : CTP; *CONSTANT POINTERS*             PAS
                                                                               PAS
      NEXT,CTPTR,MAXCTP : CTP;                                                 PAS
      *VARIABLES POINTING INTO CONTEXTTABLE*                                   PAS
                                                                               PAS
      CH : CHAR; AVAL : ALFA; IVAL : INTEGER; A : CWORD;                       PAS
      NO,CL : SHRTINT;                                                         PAS
      *CH IS OUTPUT BY NEXTCH AND USED BY INSYMBOL.                            PAS
         AVAL/IVAL, NO, CL  ARE OUTPUT BY INSYMBOL*                            PAS
                                                                               PAS
      TCT,TMAX,B6DPL,KK : SHRTINT ;                                            PAS
      PT : CTP; IT,IT1 : INTEGER;                                              PAS
   *AUXILIARY VARIABLES, USED IN MAIN PGM AND SEVERAL PROCEDURES*              PAS
                                                                               PAS
      DIGITS : POWERSET CHAR ;                                                 PAS
      *0: SIN/COS, 1: EXP, 2: LN, 3: SQRT, 4: ARCTAN*                          PAS
                                                                               PAS
      *CONSTANTS AND CONSTANT TABLES: *                                        PAS
      TWOTO17 : INTEGER;                                                       PAS
      SPLITSTAT : ARRAY [0..48] OF SHRTINT;                                    PAS
      ERRCL,TERRCL : ARRAY [0..48] OF (IRRELSY,BEGSY,ENDSY);                   PAS
      BLANK : ALFA;                                                            PAS
      WD : ARRAY [0..32] OF ALFA;                                              PAS
      WNO,WCL : ARRAY [0..32] OF SHRTINT;                                      PAS
```

LIST OF CONTROL, ACTIVE, AND/OR INACTIVE CARDS IN PASCSYM 003006

```
      WL : ARRAY [0..11] OF SHRTINT;                                          PAS
      ↱ WD = WORD DELIMITERS                                                  PAS
       WNO,WCL = CORRESPONDING VALUES OF  NO  AND  CL                         PAS
       WORD DELIMITERS OF LENGTH  I  START IN WD[WL[I]]  ↓                    PAS
      SYMNO,SYMCL : ARRAY [Ξ+Ξ..Ξ;Ξ] OF SHRTINT;                             PAS
      ↱NO AND CL OF EACH CHARACTER↓                                           PAS
      JB1,JB2,JX1,JX2,REL,FS,R1 : ARRAY [0..7] OF CHAR;                       PAS
      SH : ARRAY [12..23] OF CHAR;                                            PAS
      ↱ARRAYS JB1 ... SH ARE USED BY PRTCOMP↓                                 PAS
      INITNAM : ARRAY [-1..37] OF ALFA;                                       PAS
      ↱CONTAINS THE PREDEFINED IDENTIFIERS,                                   PAS
         USED FOR INITIALIZING THE CONTEXT TABLE↓                             PAS
                                                                              PAS
  ↱ ---------------------------------------------------------------- ↓        PAS
                                                                              PAS
  ↱ VARIABLE INITIALIZATIONS  ↓                                               PAS
    VALUE                                                                     PAS
      TWOTO17 = 400000B ;                                                     PAS
      SPLITSTAT = (1,2,19*1,3,1,4,1,1,5,1,6,1,7,1,8,2*1,9,12*1,10) ;          PAS
      ERRCL = (16*IRRELSY,ENDSY,4*IRRELSY,BEGSY,ENDSY,BEGSY,IRRELSY,          PAS
               ENDSY,BEGSY,IRRELSY,BEGSY,ENDSY,BEGSY,IRRELSY,BEGSY,           PAS
               2*IRRELSY,BEGSY,IRRELSY,ENDSY,2*IRRELSY,2*ENDSY,               PAS
               IRRELSY,3*ENDSY,2*IRRELSY,BEGSY);                              PAS
      TERRCL = (9*IRRELSY,BEGSY,ENDSY,IRRELSY,ENDSY,3*IRRELSY,                PAS
         ENDSY,IRRELSY,BEGSY,2*IRRELSY,3*ENDSY,2*IRRELSY,ENDSY,               PAS
         IRRELSY,ENDSY,IRRELSY,ENDSY,IRRELSY,ENDSY,2*IRRELSY,                 PAS
         ENDSY,IRRELSY,ENDSY,BEGSY,IRRELSY,2*ENDSY,IRRELSY,                   PAS
         3*ENDSY,2*IRRELSY,ENDSY);                                            PAS
      BLANK = Ξ  Ξ ;                                                          PAS
      WD = (ΞIFΞ,ΞDOΞ,ΞTOΞ,ΞOFΞ,ΞINΞ,                                         PAS
         ΞENDΞ,ΞNILΞ,ΞFORΞ,ΞDIVΞ,ΞMODΞ,ΞVARΞ,                                PAS
         ΞTHENΞ,ΞELSEΞ,ΞGOTOΞ,ΞCASEΞ,ΞWITHΞ,ΞTYPEΞ,ΞFILEΞ,                   PAS
         ΞBEGINΞ,ΞUNTILΞ,ΞWHILEΞ,ΞARRAYΞ,ΞVALUEΞ,ΞCLASSΞ,                     PAS
            ΞCONSTΞ,ΞLABELΞ,                                                  PAS
         ΞREPEATΞ,ΞDOWNTOΞ,ΞRECORDΞ,ΞPACKEDΞ,                                  PAS
         ΞFUNCTIONΞ,ΞPOWERSETΞ,ΞPROCEDUREΞ);                                   PAS
      WNO = (23,31,33,27,8,                                                   PAS
         22,36,32,6,6,43,                                                     PAS
         24,25,35,26,48,37,38,                                                PAS
         21,29,30,38,47,38,41,40,                                             PAS
         28,33,38,42,                                                         PAS
         44,38,45);                                                           PAS
      WCL = (0,0,1,0,7,                                                       PAS
         0,0,0,4,5,0,                                                         PAS
         6*0,3,                                                               PAS
         0,0,0,1,0,4,0,0,                                                     PAS
         0,2,2,0,                                                             PAS
         0,5,0);                                                              PAS
      WL = (0,0,0,5,11,18,26,30,30,32,33,33);                                 PAS
      SYMNO = (7,7,6,6,9,10,4,8,0,15,17,0,11,12,                              PAS
         19,8,13,7,6,18,14,8,8,8,8,5,16);                                     PAS
      SYMCL = (1,2,1,2,0,0,1,6,3,0,0,2,0,0,                                   PAS
```

```
        0,5,0,3,3,0,0,1,4,2,3,1,0);                                          PAS
      JB1 = (=P=,=R=,=J=,= =,=E=,=N=,=G=,=L=);                               PAS
      JB2 = (=S=,=J=,=P=,= =,=Q=,=E=,=E=,=T=);                               PAS
      JX1 = (=Z=,=N=,=P=,=N=,=I=,=O=,=D=,=I=);                               PAS
      JX2 = (=R=,=Z=,=L=,=G=,=R=,=R=,=F=,=D=);                               PAS
      REL = (= =,=^=,=v=,=-=,==,=≠=,=≥=,=<=);                                PAS
      FS  = (=F=,=R=,=D=,= =,= =,=A=,=B=,=X=);                               PAS
      R1  = (=A=,=B=,=X=,=X=,=A=,=A=,=B=,=B=);                               PAS
      SH  = (=F=,=D=,=R=,=I=,=L=,=A=,=L=,=A=,=N=,=Z=,=U=,=P=);               PAS
      INITNAM = (=TEXT=,=WEOR=,=GET=,=PUT=,=RESET=,=ALLOC=,                  PAS
        =PACK=,=UNPACK=,=INSERT=,=APPEND=,                                   PAS
        =READ=,=WRITE=,                                                      PAS
        =ODD=,=INT=,=CHR=,=EOF=,=ABS=,=SQR=,                                PAS
        =TRUNC=,=PRED=,=SUCC=,=EOL=,=ALFALENG=,                             PAS
        =INPUT=,=OUTPUT=,=ALFA=,=REAL=,=CHAR=,=BOOLEAN=,                    PAS
        =FALSE=,=TRUE=,=INTEGER=,=   =,                                     PAS
        =SIN=,=COS=,=EXP=,=LN=,=SQRT=,=ARCTAN=) ;                           PAS
                                                                             PAS
  ↱TWO STANDARD PROCEDURES  INSERT  AND  APPEND  ARE PROVIDED FOR            PAS
    CODE GENERATION.                                                         PAS
   APPEND(A,B,C) SHIFTS THE CONTENTS OF A LEFT B BITS AND                    PAS
    =OR=S C INTO IT. RESULT IN A. B,C UNCHANGED.                             PAS
   INSERT(A,B,C) SHIFTS THE CONTENTS OF A LEFT B BITS AND                    PAS
    =OR=S THEM INTO C. RESULT IN C. A,B UNCHANGED                            PAS
   A,B,C ARE OF TYPE INTEGER  ↳                                              PAS
                                                                             PAS
                                                                             PAS
  PROCEDURE PRTERR ;                                                         PAS
    VAR LASTPOS,FREEPOS,CURRPOS,CURRNMR,F,K : SHRTINT ;                      PAS
  BEGIN WRITE(= **** = : 7) ;                                                PAS
    LASTPOS := 0 ; FREEPOS := 1 ;                                            PAS
    FOR K := 1 TO ERRINX DO                                                  PAS
    BEGIN WITH ERRLIST[K] DO                                                 PAS
      BEGIN CURRPOS := POS ; CURRNMR := NMR END ;                            PAS
      IF CURRPOS = LASTPOS THEN WRITE(=,=) ELSE                              PAS
      BEGIN WHILE FREEPOS < CURRPOS DO                                       PAS
        BEGIN WRITE(= =) ;                                                   PAS
          FREEPOS := FREEPOS + 1                                             PAS
        END ;                                                                PAS
        WRITE(=↑=) ;                                                        PAS
        LASTPOS := CURRPOS                                                   PAS
      END ;                                                                  PAS
      IF CURRNMR < 10 THEN F := 1 ELSE                                       PAS
      IF CURRNMR < 100 THEN F := 2 ELSE F := 3 ;                             PAS
      WRITE(CURRNMR : F) ;                                                   PAS
      FREEPOS := FREEPOS + F + 1 ;                                           PAS
    END ;                                                                    PAS
    WRITE(EOL) ; ERRINX := 0                                                 PAS
  END ↱PRTERR↳ ;                                                             PAS
                                                                             PAS
  PROCEDURE NEXTCH;                                                          PAS
    ↱READS NEXT CHAR OF FILE INPUT.                                          PAS
```

LIST OF CONTROL, ACTIVE, AND/OR INACTIVE CARDS IN PASCSYM 000008

```
        CONVERTS EOL TO BLANK,                                                  PAS
        PRINTS ERROR SUMMARY AFTER EACH LINE (IF ERRORS PRESENT)                PAS
        AND PRINTS ADDRESS AT BEGINNING OF LINE ↓                               PAS
  BEGIN                                                                         PAS
    IF EOLFLAG THEN                                                             PAS
    BEGIN IF ERRINX > 0 THEN ↱PRINT ERRORS↓ PRTERR;                             PAS
      EOLFLAG := FALSE;  CHCNT := 0;                                            PAS
      OUTPUT↑ := Ξ Ξ; PUT(OUTPUT);                                              PAS
      IF DP THEN WRITE(LC : 6 OCT) ELSE WRITE(IC : 6 OCT) ;                     PAS
      OUTPUT↑ := Ξ Ξ;  PUT(OUTPUT);                                             PAS
    END ↱EOLFLAG↓;                                                              PAS
    GET(INPUT); CH := INPUT↑;                                                   PAS
    OUTPUT↑ := CH; PUT(OUTPUT); CHCNT := CHCNT + 1;                             PAS
    IF CH = EOL THEN                                                            PAS
    BEGIN CH := Ξ Ξ;  EOLFLAG := TRUE  END;                                     PAS
  END ↱NEXTCH↓;                                                                 PAS
                                                                                PAS
  PROCEDURE ERROR(I:SHRTINT);                                                   PAS
  BEGIN ERR := TRUE ;                                                           PAS
    WITH PARMLIST DO                                                            PAS
    BEGIN ERRFLAG := TRUE ;                                                     PAS
      ERRNRS[I DIV 32] := ERRNRS[I DIV 32]∨[I MOD 32]                           PAS
    END ;                                                                       PAS
    IF ERRINX ≥ 9 THEN                                                          PAS
    BEGIN ERRLIST[10].NMR := 104 ; ERRINX := 10 ;                               PAS
      WITH PARMLIST DO                                                          PAS
      ERRNRS[3] := ERRNRS[3]∨[8] ;                                              PAS
    END ELSE                                                                    PAS
    BEGIN ERRINX := ERRINX + 1 ;                                                PAS
      ERRLIST[ERRINX].NMR := I                                                  PAS
    END ;                                                                       PAS
    ERRLIST[ERRINX].POS := CHCNT                                                PAS
  END ↱ERROR↓;                                                                  PAS
                                                                                PAS
                                                                                PAS
↱ ---------------------------------------------------------------------- ↓     PAS
                                                                                PAS
↱ I N S Y M B O L          READS ONE PASCAL-SYMBOL ↓                            PAS
  ↱ LIST OF SYMBOL NUMBERS                                                      PAS
  NO   CL   SYMBOL            NO   CL   SYMBOL                                  PAS
                                                                                PAS
   1        ID                20        :=                                     PAS
   2    1   INTEGER CONST     21        BEGIN                                  PAS
   2    2   REAL    CONST     22        END                                    PAS
   2    3   ALFA    CONST     23        IF                                     PAS
   2    4   CHAR    CONST     24        THEN                                   PAS
                              25        ELSE                                   PAS
                              26        CASE                                   PAS
   5    1   ¬                 27        OF                                     PAS
   6    1   *                 28        REPEAT                                 PAS
   6    2   /                 29        UNTIL                                  PAS
   6    3   ^                 30        WHILE                                  PAS
```

LIST OF CONTROL, ACTIVE, AND/OR INACTIVE CARDS IN PASCSYM 000009

```
     6    4   DIV                 31        DO                                PAS
     6    5   MOD                 32        FOR                               PAS
     7    1   +                   33    1   TO                                PAS
     7    2   -                   33    2   DOWNTO                            PAS
     7    3   v                   35        GOTO                              PAS
     8    1   <                   36        NIL                               PAS
     8    2   ≤                   37        TYPE                              PAS
     8    3   ≥                   38    1   ARRAY                             PAS
     8    4   >                   38    2   RECORD                            PAS
     8    5   ≠                   38    3   FILE                              PAS
     8    6   =                   38    4   CLASS                             PAS
     8    7   IN                  38    5   POWERSET                          PAS
     9        (                   40        LABEL                             PAS
    10        )                   41        CONST                             PAS
    11        [                   42        PACKED                            PAS
    12        ]                   43        VAR                               PAS
    15        ,                   44        FUNCTION                          PAS
    16        ;                   45        PROCEDURE                         PAS
    17        .                   47        VALUE                             PAS
    18        ↑                   48        WITH                              PAS
    19        :                                                               PAS
                                                                              PAS
      NOT PASCAL SYMBOLS : BLANK  Ξ  ↱  $  ↳                                  PAS
                                                                              PAS
  PROCEDURE INSYMBOL;                                                         PAS
    VAR  SCALE,EXP : SHRTINT;  R,FAC,RVAL : REAL;                             PAS
      DIGIT : ARRAY[0..19] OF SHRTINT;                                        PAS
              ↱FOR WORDLENGTH DIV 3 = 20 DIGITS↳                              PAS
      I,K : SHRTINT;  BT1,SIGN : BOOLEAN;                                     PAS
      AA : ALFA; CH1 : CHAR;                                                  PAS
                                                                              PAS
  BEGIN                                                                       PAS
1:  WHILE CH=Ξ Ξ DO NEXTCH;                                                   PAS
    IF CH ≤ ΞZΞ THEN                                                          PAS
    BEGIN  ↱ READ IDENTIFIER OR WORD DELIMITER ↳                              PAS
      K := 0;                                                                 PAS
      REPEAT IF K < ALFALENG THEN                                             PAS
        BEGIN K := K + 1;  A[K] := CH END;                                    PAS
        NEXTCH;                                                               PAS
      UNTIL CH >Ξ9Ξ;                                                          PAS
      IF K ≥ KK THEN KK := K ELSE                                             PAS
      REPEAT A[KK] := Ξ Ξ ; KK := KK - 1 UNTIL KK = K ;                       PAS
      PACK(A,1,AA);                                                           PAS
      ↱ TEST FOR WORD DELIMITER ↳                                             PAS
      FOR I := WL[K] TO WL[K+1]-1 DO                                          PAS
        IF AA = WD[I] THEN                                                    PAS
        BEGIN NO := WNO[I];  CL := WCL[I];                                    PAS
          IVAL := 0;  GOTO 2;                                                 PAS
        END;                                                                  PAS
      ↱ IDENTIFIER ↳                                                          PAS
      NO := 1;  CL := K;  AVAL := AA;                                         PAS
2:  END  ↱LETTER↳  ELSE                                                       PAS
```

LIST OF CONTROL, ACTIVE, AND/OR INACTIVE CARDS IN PASCSYM 000010

```
      IF CH ≤ Ξ9Ξ THEN                                                    PAS
      BEGIN NO := 2; CL := 1; I := 0;                                     PAS
         WHILE CH IN DIGITS DO                                            PAS
         BEGIN IF I < 20 ↱20 = WORDLENGTH DIV 3↳ THEN                     PAS
            DIGIT[I] := INT(CH) - INT(Ξ0Ξ);                               PAS
            I := I + 1; NEXTCH                                            PAS
         END;                                                             PAS
         IVAL := 0;                                                       PAS
         IF CH = ΞBΞ THEN ↱OCTAL↳                                          PAS
         BEGIN IF I > 20 THEN ERROR(2) ELSE                               PAS
            FOR K := 0 TO I-1 DO IVAL := 8*IVAL + DIGIT[K];               PAS
            NEXTCH                                                        PAS
         END ↱OCTAL↳ ELSE                                                 PAS
         BEGIN IF I > 17 THEN ERROR(2) ELSE                               PAS
            FOR K := 0 TO I - 1 DO                                        PAS
               BEGIN IF IVAL ≤ MAX10 THEN                                 PAS
                  IVAL := 10*IVAL + DIGIT[K] ELSE                         PAS
                  BEGIN ERROR(2); IVAL := 0; GOTO 40 END                  PAS
               END;                                                       PAS
40:         SCALE := 0;                                                   PAS
            IF CH = Ξ.Ξ THEN                                              PAS
            BEGIN NEXTCH; IF CH = Ξ.Ξ THEN CH := Ξ:Ξ ELSE                 PAS
               BEGIN RVAL := IVAL; CL := 2;                               PAS
                  IF ¬(CH IN DIGITS) THEN ERROR(3) ELSE                   PAS
                  REPEAT R := INT(CH) - INT(Ξ0Ξ);                         PAS
                     RVAL := 10.0*RVAL + R;                               PAS
                     SCALE := SCALE - 1; NEXTCH                           PAS
                  UNTIL ¬(CH IN DIGITS);                                  PAS
               END;                                                       PAS
            END;                                                          PAS
            IF CH = ΞEΞ THEN                                              PAS
            BEGIN IF SCALE = 0 THEN                                       PAS
               BEGIN RVAL := IVAL; CL := 2                                PAS
               END;                                                       PAS
               SIGN := FALSE; NEXTCH;                                     PAS
               IF CH = Ξ+Ξ THEN NEXTCH ELSE                               PAS
               IF CH = Ξ-Ξ THEN                                           PAS
                  BEGIN NEXTCH; SIGN := TRUE                              PAS
                  END;                                                    PAS
               EXP := 0;                                                  PAS
               WHILE CH IN DIGITS DO                                      PAS
               BEGIN EXP := 10*EXP + (INT(CH) - INT(Ξ0Ξ));                PAS
                  NEXTCH                                                  PAS
               END;                                                       PAS
               IF SIGN THEN EXP := -EXP;                                  PAS
               SCALE := SCALE + EXP                                       PAS
            END;                                                          PAS
            IF SCALE ≠ 0 THEN                                             PAS
            BEGIN R := 1.0;                                               PAS
               IF SCALE < 0 THEN BEGIN FAC := 0.1; SCALE := -SCALE END    PAS
                            ELSE FAC := 10.0;                             PAS
               REPEAT IF ODD(SCALE) THEN R := R*FAC;                      PAS
```

```
                FAC := SQR(FAC); SCALE := SCALE DIV 2                                PAS
              UNTIL SCALE = 0; ↱R = 10 ** SCALE↳                                     PAS
              RVAL := RVAL*R                                                         PAS
           END;                                                                      PAS
           IF CL = 2 THEN IVAL := INT(RVAL)                                          PAS
        END ↱NOT OCTAL↳                                                              PAS
     END ↱NUMBER↳ ELSE                                                               PAS
     BEGIN  ↱ SPECIAL CHARACTER ↳                                                    PAS
        IF CH = ΞΞΞΞ THEN  ↱ ALFA OR CHAR CONSTANT ↳                                 PAS
        BEGIN NO := 2; K := 0; BT1 := FALSE;                                         PAS
           REPEAT NEXTCH;                                                            PAS
              IF CH = ΞΞΞΞ THEN                                                      PAS
              BEGIN NEXTCH; BT1 := CH ≠ ΞΞΞΞ END;                                    PAS
              IF ¬BT1 THEN                                                           PAS
                 IF K = ALFALENG THEN                                                PAS
                 BEGIN ERROR(3); BT1 := TRUE END ELSE                                PAS
                 BEGIN K := K + 1; A[K] := CH END;                                   PAS
           UNTIL BT1;                                                                PAS
           IF K =1 THEN  ↱ CHAR CONST ↳                                              PAS
           BEGIN CL:=4; IVAL:= INT (A[1]) END                                        PAS
           ELSE  ↱ ALFA CONST ↳                                                      PAS
           BEGIN IF K ≥ KK THEN KK := K ELSE                                         PAS
              REPEAT A[KK] := Ξ Ξ ; KK := KK - 1 UNTIL KK = K ;                      PAS
              PACK (A,1,AA);                                                         PAS
              CL:=3; IVAL:= INT(AA);                                                 PAS
           END;                                                                      PAS
        END  ↱ Ξ ↳  ELSE                                                             PAS
        BEGIN NO:= SYMNO[CH]; CL:=SYMCL[CH]; IVAL:=0;                                PAS
            ↱ TEST FOR TWO-CHARACTER SYMBOL ↳                                        PAS
           IF CH = Ξ:Ξ THEN                                                          PAS
           BEGIN NEXTCH;                                                             PAS
              IF CH=Ξ=Ξ THEN                                                         PAS
              BEGIN NO:= 20; NEXTCH END;                                             PAS
           END ELSE                                                                  PAS
           IF CH =Ξ.Ξ THEN                                                           PAS
           BEGIN NEXTCH;                                                             PAS
              IF CH = Ξ.Ξ THEN  ↱ : ↳                                                PAS
              BEGIN NO:=19; NEXTCH END;                                              PAS
           END ELSE                                                                  PAS
           IF CH = Ξ↱Ξ THEN  ↱ SKIP COMMENT ↳                                        PAS
           BEGIN NEXTCH;                                                             PAS
              IF CH = Ξ$Ξ THEN                                                       PAS
              REPEAT NEXTCH ;                                                        PAS
              IF CH ≠ Ξ↳Ξ THEN                                                       PAS
              BEGIN CH1 := CH ; NEXTCH ;                                             PAS
                 IF CH1 = ΞAΞ THEN ASSCHECK := CH = Ξ+Ξ ELSE                         PAS
                 IF CH1 = ΞCΞ THEN PRCODE := CH = Ξ+Ξ ELSE                           PAS
                 IF CH1 = ΞDΞ THEN DIVCHECK := CH = Ξ+Ξ ELSE                         PAS
                 IF CH1 = ΞOΞ THEN STOFLCHECK := CH = Ξ+Ξ ELSE                       PAS
                 IF CH1 = ΞRΞ THEN ROUNDING := CH = Ξ+Ξ ELSE                         PAS
                 IF CH1 = ΞTΞ THEN                                                   PAS
                 BEGIN ASSCHECK := CH = Ξ+Ξ ;                                        PAS
```

LIST OF CONTROL, ACTIVE, AND/OR INACTIVE CARDS IN PASCSYM 000012

```
                    DIVCHECK := CH = Ξ+Ξ ; STOFLCHECK := CH = Ξ+Ξ ;                PAS
                    ROUNDING := CH = Ξ+Ξ ; INXCHECK := CH = Ξ+Ξ                    PAS
                  END ELSE                                                         PAS
                  IF CH1 = ΞXΞ THEN INXCHECK := CH = Ξ+Ξ ;                         PAS
                  IF CH ≠ Ξ↓Ξ THEN NEXTCH                                          PAS
                END ;                                                              PAS
                UNTIL CH ≠ Ξ,Ξ ;                                                   PAS
                WHILE CH ≠ Ξ↓Ξ DO NEXTCH;                                          PAS
                NEXTCH;  GOTO 1 ;                                                  PAS
              END ELSE                                                             PAS
            NEXTCH;                                                                PAS
        END  ↱ SPECIAL CHARACTER ↳ ;                                               PAS
    END;                                                                           PAS
  END ↱ INSYMBOL ↳ ;                                                               PAS
                                                                                   PAS
  ↱ ---------------------------------------------------------------------- ↳      PAS
                                                                                   PAS
    PROCEDURE SRCHREC(P:CTP);                                                      PAS
      ↱ SEARCHES ONE BLOCK, RETURNS CTPTR ↳                                        PAS
    BEGIN CTPTR := P;                                                              PAS
      WHILE CTPTR ≠ NIL DO                                                         PAS
        IF CTPTR↑.NAME = AVAL THEN GOTO 1                                          PAS
        ELSE CTPTR := CTPTR↑.NXTEL;                                                PAS
  1:END  ↱ SRCHREC ↳ ;                                                             PAS
                                                                                   PAS
    PROCEDURE SEARCH;                                                              PAS
      ↱ SEARCHES CONTEXTTABLE, RETURNS CTPTR AND DISX = INDEX TO                   PAS
        DISPLAY ↳                                                                  PAS
      VAR I : SHRTINT;                                                             PAS
    BEGIN FOR I := TOP DOWNTO 0 DO                                                 PAS
      BEGIN CTPTR := DISPLAY[I].FNAME;                                             PAS
        WHILE CTPTR ≠ NIL DO                                                       PAS
          IF CTPTR↑.NAME = AVAL THEN GOTO 1                                        PAS
          ELSE CTPTR := CTPTR↑.NXTEL;                                              PAS
      END;                                                                         PAS
  1:  DISX := I;                                                                   PAS
    END  ↱ SEARCH ↳ ;                                                              PAS
                                                                                   PAS
    PROCEDURE INCONST (VAR V : INTEGER; P : CTP; CONST NXT : CTP) ;                PAS
      ↱ INPUT PARAMETER : NXT. SEARCHING IS TO BEGIN AT NXT                        PAS
        OUTPUT PARAMETER : V = VALUE                                               PAS
                           P = TYPE POINTER , P = NIL IF ERROR ↳                   PAS
      VAR SIGN : BOOLEAN;  PT : CTP;                                               PAS
    BEGIN SIGN := FALSE;  P := NIL;                                                PAS
      IF NO = 7 THEN                                                               PAS
      BEGIN SIGN := CL = 2;                                                        PAS
        IF CL ≤ 2 THEN                                                             PAS
        BEGIN INSYMBOL;  IF NO = 1 THEN ERROR(3)  END;                             PAS
      END;                                                                         PAS
      IF NO = 2 THEN                                                               PAS
      BEGIN CASE CL OF                                                             PAS
  1:    P := INTPTR;                                                               PAS
```

LIST OF CONTROL, ACTIVE, AND/OR INACTIVE CARDS IN PASCSYM

```
2:       P := REALPTR;                                                    PAS
3:       P := ALFAPTR;                                                    PAS
4:       P := CHARPTR;                                                    PAS
        END;                                                              PAS
         IF SIGN THEN V := -IVAL ELSE V := +IVAL;                         PAS
         INSYMBOL;                                                        PAS
      END ELSE                                                            PAS
      IF NO = 1 THEN                                                      PAS
      BEGIN PT := CTPTR;                                                  PAS
         SRCHREC(NXT);                                                    PAS
         IF CTPTR = NIL THEN SEARCH;                                      PAS
         IF CTPTR = NIL THEN ERROR(12) ELSE                               PAS
         WITH CTPTR↑ DO                                                   PAS
         BEGIN IF (KLASS ≠ KONST)∨(CONKIND = FORMAL) THEN ERROR(63)       PAS
            ELSE                                                          PAS
            BEGIN P := CONTYPE;  V := VALUES END;                         PAS
         END;                                                             PAS
         CTPTR := PT;  INSYMBOL;                                          PAS
      END ELSE ERROR(3);                                                  PAS
   END ↱ INCONST ↳ ;                                                      PAS
                                                                          PAS
↱ ------------------------------------------------------------------ ↳   PAS
                                                                          PAS
↱ CODE GENERATING ROUTINES ↳                                              PAS
                                                                          PAS
   PROCEDURE NOOP;                                                        PAS
   BEGIN WHILE CP < 4 DO                                                  PAS
      BEGIN APPEND(BUF,15,NOP);                                           PAS
         CP := CP + 1;                                                    PAS
      END ;                                                               PAS
   END  ↱ NOOP ↳  ;                                                       PAS
                                                                          PAS
   PROCEDURE GEN15(OP,I,J,K : SHRTINT);                                   PAS
   BEGIN LASTOP := OP ; LASTI := I ;                                      PAS
      IF CP ≠ 4 THEN                                                      PAS
      BEGIN CP := CP + 1; APPEND(BUF,6,OP)  END ELSE                      PAS
      BEGIN CODE[CA] := BUF;  BUF := OP;  CP := 1;                        PAS
         CA := CA + 1;  IC := IC + 1;                                     PAS
         IF CA > CODMAX THEN                                              PAS
         BEGIN ERROR(90);  CA := 1;                                       PAS
         END;                                                             PAS
      END;                                                                PAS
      BUF := ((8*BUF + I)*8 + J)*8 + K ;                                  PAS
END   ↱ GEN15 ↳ ;                                                         PAS
                                                                          PAS
   PROCEDURE GEN30(OP,I,J,K : SHRTINT);                                   PAS
   BEGIN LASTOP := OP ; LASTI := I ;                                      PAS
      IF CP = 3 THEN ↱NOOP↳                                               PAS
      BEGIN APPEND(BUF,15,NOP);  CP := 4 END;                             PAS
      IF CP < 4 THEN                                                      PAS
      BEGIN CP := CP + 2;  APPEND(BUF,6,OP)  END ELSE                     PAS
      BEGIN CODE[CA] := BUF;  BUF := OP;  CP := 2;                        PAS
```

LIST OF CONTROL, ACTIVE, AND/OR INACTIVE CARDS IN PASCSYM 000014

```
        CA := CA + 1;  IC := IC + 1;                                    PAS
        IF CA > CODMAX THEN                                             PAS
        BEGIN ERROR(90);  CA := 1;                                      PAS
        END;                                                            PAS
      END;                                                              PAS
      BUF := (8*BUF + I)*8 + J ;                                        PAS
      IF K ≥ 0 THEN APPEND(BUF,18,K) ELSE APPEND(BUF,18,777777B+K) ;    PAS
  END  ↱ GEN30 ↳ ;                                                      PAS
                                                                        PAS
    PROCEDURE BXIXJ(I, J : SHRTINT) ;                                   PAS
    ↱TO AVOID B XI XJ INSTRUCTIONS WHENEVER POSSIBLE↳                   PAS
    BEGIN IF (J = LASTI)∧(LASTOP ≥ 10B)∧(LASTOP ≤ 45B)                  PAS
       ∧¬(LASTOP IN [20B,21B,43B]) THEN                                 PAS
       BEGIN IF BUF ≥ 0 THEN BUF := BUF - LASTI*64 + I*64 ELSE          PAS
          BEGIN BUF := - BUF ;                                          PAS
             BUF := BUF - (7 - LASTI)*64 + (7 - I)*64 ;                 PAS
             BUF := - BUF                                               PAS
          END ;                                                         PAS
          LASTI := I                                                    PAS
       END ELSE GEN15(10B,I,J,0)                                        PAS
    END ↱BXIXJ↳ ;                                                       PAS
                                                                        PAS
    PROCEDURE INS(FADR : ADDRESS ; FCP, FCA : SHRTINT) ;                PAS
    BEGIN IF CA ≠ FCA THEN INSERT(FADR,(4-FCP)*15,CODE[FCA])            PAS
                      ELSE INSERT(FADR,(CP-FCP)*15,BUF)                 PAS
    END ;                                                               PAS
                                                                        PAS
  ↱ ---------------------------------------------------------------- ↳  PAS
                                                                        PAS
  ↱ ROUTINES TO PRINT CODE AND JUMPTABLE AND TO OUTPUT CODE ↳           PAS
                                                                        PAS
    PROCEDURE PRTCOMP;                                                  PAS
       ↱ GLOBAL ARRAYS : JB1,JB2,JX1,JX2,REL,FS,R1,SH ↳                 PAS
       VAR II,JJ,KK : CHAR;                                             PAS
          C,CC,S,IT3, OP,I,J,K : SHRTINT;                               PAS
          M,IT4 : SHRTINT;                                              PAS
                                                                        PAS
       PROCEDURE OUTCH(C : CHAR) ;                                      PAS
       BEGIN WRITE(C) END ;                                             PAS
                                                                        PAS
       PROCEDURE SPACE ;                                                PAS
       BEGIN WRITE(Ξ Ξ : 3) END ;                                       PAS
                                                                        PAS
    BEGIN                                                               PAS
       IC := IC - CA - 1;                                               PAS
       M := -77B;  IT4 := -7B;                                          PAS
       FOR IT3 := 1 TO CA DO                                            PAS
       BEGIN BUF := -CODE[IT3];  OUTCH(EOL);                            PAS
          OUTCH(Ξ Ξ) ; OUTCH(Ξ Ξ) ; WRITE(IC+IT3 : 6 OCT);  S := 0;     PAS
          REPEAT IT := BUF;  APPEND(IT,S+6,M);  OP := -IT;              PAS
             C := OP MOD 8;                                             PAS
             IT := BUF;  APPEND(IT,S+9,IT4);  I := -IT;                 PAS
```

000015

LIST OF CONTROL, ACTIVE, AND/OR INACTIVE CARDS IN PASCSYM

```
            IT := BUF;  APPEND(IT,S+12,IT4);  J := -IT;                                  PAS
            IT := BUF;                                                                   PAS
            IF (OP < 8) v ((OP ≥ 40) ^ (C ≤ 2)) THEN                                     PAS
            BEGIN S := S+30;  APPEND(IT,S,MASK)  END ELSE                                PAS
            BEGIN S := S+15;  APPEND(IT,S,IT4)  END;                                     PAS
            K := -IT;                                                                    PAS
                                                                                         PAS
      CC := OP DIV 8;                                                                    PAS
      II :=CHR(I+INT(Ξ0Ξ)); JJ := CHR(J+INT(Ξ0Ξ)); KK :=CHR(K+INT(Ξ0Ξ));                 PAS
      OUTCH(EOL); WRITE(BLANK);                                                          PAS
      IF CC < 5 THEN                                                                     PAS
      CASE CC OF                                                                         PAS
0:       IF OP = 3 THEN                                                                  PAS
         BEGIN OUTCH(JX1[I]);  OUTCH(JX2[I]);  OUTCH(Ξ Ξ);                               PAS
            SPACE;  OUTCH(ΞXΞ);  OUTCH(JJ);  OUTCH(Ξ,Ξ);                                 PAS
            WRITE(K : 6 OCT);                                                            PAS
         END ELSE                                                                        PAS
         BEGIN OUTCH(JB1[OP]);  OUTCH(JB2[OP]);  OUTCH(Ξ Ξ);                             PAS
            SPACE;                                                                       PAS
            IF OP > 3 THEN                                                               PAS
            BEGIN OUTCH(ΞBΞ);  OUTCH(II);  OUTCH(REL[OP]);                               PAS
               OUTCH(ΞBΞ);  OUTCH(JJ);  OUTCH(Ξ,Ξ);  WRITE(K : 6 OCT);                   PAS
            END ELSE                                                                     PAS
            BEGIN IF OP = 2 THEN                                                         PAS
               BEGIN OUTCH(ΞBΞ) ; OUTCH(II) ; OUTCH(Ξ+Ξ) END ;                           PAS
               WRITE(K : 6 OCT)                                                          PAS
            END;                                                                         PAS
         END  ↱ 0 ↳ ;                                                                    PAS
1:       BEGIN OUTCH(ΞBΞ);  OUTCH(ΞXΞ);  OUTCH(II);  SPACE;                              PAS
            IF OP > 11 THEN                                                              PAS
            BEGIN OUTCH(Ξ¬Ξ);  II := JJ; JJ := KK; KK := II  END;                        PAS
            OUTCH(ΞXΞ);  OUTCH(JJ);                                                      PAS
            C := OP MOD 4;                                                               PAS
            IF C ≠ 0 THEN                                                                PAS
            BEGIN OUTCH(REL[C]);  OUTCH(ΞXΞ);  OUTCH(KK)  END;                           PAS
         END  ↱ 1 ↳ ;                                                                    PAS
2:       BEGIN OUTCH(SH[OP]);  OUTCH(ΞXΞ);  OUTCH(II);  SPACE;                           PAS
            IF OP < 18 THEN                                                              PAS
            BEGIN OUTCH(JJ);  OUTCH(KK)  END ELSE                                        PAS
            BEGIN OUTCH(ΞBΞ);  OUTCH(JJ);  OUTCH(Ξ,Ξ);                                   PAS
               OUTCH(ΞXΞ);  OUTCH(KK);                                                   PAS
            END;                                                                         PAS
         END  ↱ 2 ↳ ;                                                                    PAS
3:       BEGIN OUTCH(SH[OP DIV 2]);  OUTCH(ΞXΞ);  OUTCH(II);                             PAS
            SPACE;  OUTCH(ΞXΞ);  OUTCH(JJ);                                              PAS
          IF ODD(OP) THEN OUTCH(Ξ-Ξ) ELSE OUTCH(Ξ+Ξ);                                    PAS
            OUTCH(ΞXΞ);  OUTCH(KK);                                                      PAS
         END  ↱ 3 ↳ ;                                                                    PAS
4:       BEGIN C := OP MOD 4;                                                            PAS
            IF OP = 38 THEN                                                              PAS
            BEGIN OUTCH(ΞNΞ);  OUTCH(Ξ0Ξ)  END ELSE                                      PAS
            IF C = 3 THEN                                                                PAS
```

```
            BEGIN IF OP = 35 THEN OUTCH(ΞMΞ) ELSE OUTCH(ΞCΞ);                    PAS
               OUTCH(ΞXΞ);  OUTCH(II);  SPACE;                                   PAS
               IF OP = 35 THEN OUTCH(JJ) ELSE OUTCH(ΞXΞ) ;                       PAS
               OUTCH(KK);                                                        PAS
            END ELSE                                                             PAS
            BEGIN OUTCH(FS[C]);  OUTCH(ΞXΞ);  OUTCH(II);                         PAS
               SPACE;  OUTCH(ΞXΞ);  OUTCH(JJ);                                   PAS
               IF OP < 36 THEN OUTCH(Ξ*Ξ) ELSE OUTCH(Ξ/Ξ);                       PAS
               OUTCH(ΞXΞ);  OUTCH(KK);                                           PAS
            END;                                                                 PAS
         END ↱ 4 ↳ ;                                                             PAS
      END ↱ CASE ↳ ELSE                                                          PAS
      BEGIN                                                                      PAS
         OUTCH(ΞSΞ);  OUTCH(FS[CC]);  OUTCH(II);                                 PAS
         SPACE;  OUTCH(R1[C]);  OUTCH(JJ);                                       PAS
         IF C IN [5,7] THEN OUTCH(Ξ-Ξ) ELSE OUTCH(Ξ+Ξ) ;                         PAS
         IF C ≤ 2 THEN WRITE(K : 6 OCT) ELSE                                     PAS
         BEGIN OUTCH(ΞBΞ);  OUTCH(KK)  END;                                      PAS
      END;                                                                       PAS
                                                                                 PAS
         UNTIL S ≥ 60;                                                           PAS
      END;                                                                       PAS
      IC := IC + CA + 1;  OUTCH(EOL);                                            PAS
   END ↱ PRTCOMP ↳ ;                                                             PAS
                                                                                 PAS
   PROCEDURE PRJMPTAB ;                                                          PAS
    ↱ PRINTS OUT JUMPTAB.                                                        PAS
     LOADPOINT OF JMPTAB IN PARMLIST.LPJMPTAB ↳                                  PAS
      VAR I : SHRTINT;                                                           PAS
   BEGIN IF JMPIX > 0 THEN WRITE(EOL,Ξ1Ξ,EOL) ;                                  PAS
      FOR I := 0 TO JMPIX - 1 DO                                                 PAS
      WRITE(Ξ Ξ : 2,PARMLIST.LPJMPTAB+I:6 OCT,Ξ Ξ,JMPTAB[I]:20 OCT,EOL)          PAS
   END ↱PRJMPTAB↳ ;                                                              PAS
                                                                                 PAS
   PROCEDURE WRITOUT ;                                                           PAS
    ↱CALLED AT PROCEDURE END, UPDATING ITS CODE (INSERTING ADDRESSES             PAS
     OF CONSTANTS) AND WRITING OUT CODE TOGETHER WITH THE CONSTANTS              PAS
     USED IN THIS PROCEDURE↳                                                     PAS
      VAR IT2 : SHRTINT;                                                         PAS
   BEGIN                                                                         PAS
      NOOP ; CODE[CA] := BUF ;                                                   PAS
      FOR IT := 1 TO LCX DO                                                      PAS
      WITH CSTTB[IT] DO                                                          PAS
      BEGIN                                                                      PAS
         CA := CA + 1 ;                                                          PAS
         IF CA > CODMAX THEN                                                     PAS
         BEGIN ERROR(90) ; CA := 1 END ELSE                                      PAS
         BEGIN                                                                   PAS
            CODE[CA] := VALU ; IT1 := INX ;                                      PAS
            REPEAT                                                               PAS
               WITH UNDLAB[IT1] DO                                               PAS
               BEGIN INS(IC,LFTSH,PLACE) ; IT2 := IT1 ; IT1 := SUCC END          PAS
```

```
            UNTIL IT1 = 0 ;                                                    PAS
            UNDLAB[IT2].SUCC := CHNIX ;                                        PAS
            CHNIX := INX ; IC := IC + 1                                        PAS
         END                                                                   PAS
      END ;                                                                    PAS
      FOR IT := 1 TO CA DO                                                     PAS
      BEGIN PASCLGO↑ := CODE[IT] ; PUT(PASCLGO) END ;                          PAS
      IF PRCODE THEN                                                           PAS
      BEGIN                                                                    PAS
         IC := IC - LCX ; CA := CA - LCX ; PRTCOMP ;                           PAS
         FOR IT := 1 TO LCX DO                                                 PAS
         BEGIN                                                                 PAS
            CA := CA+1 ; WRITE(Ξ Ξ:2,IC:6 OCT,Ξ Ξ,CODE[CA]:20 OCT,EOL) ;       PAS
            IC := IC + 1                                                       PAS
         END ;                                                                 PAS
         IF ¬EOLFLAG THEN                                                      PAS
         WRITE(Ξ Ξ:CHCNT+8) ;                                                  PAS
      END                                                                      PAS
   END  ↱ WRITOUT ↓  ;                                                         PAS
                                                                               PAS
↱ ------------------------------------------------------------------ ↓        PAS
                                                                               PAS
   PROCEDURE MULOPT(VAL1:SHRTINT; VAR EXP1,EXP2:SHRTINT; OPT:OPTPWR);          PAS
     ↱MULOPT DECIDES IF VAL = 2**EXP1              : OPT = PUREP               PAS
                         = 2**EXP1*(2**EXP2+1)     : OPT = POSP                PAS
                         = 2**EXP1*(2**EXP2-1)     : OPT = NEGP                PAS
                           OTHER                   : OPT = NOOPT ↓             PAS
      VAR E1,E2 : SHRTINT;                                                     PAS
         VAL : SHRTINT;                                                        PAS
   BEGIN EXP1 := 0 ; EXP2 := 0 ;                                               PAS
      IF VAL1 > 0 THEN                                                         PAS
      BEGIN E1 := 0;                                                           PAS
         VAL := VAL1 ;                                                         PAS
         WHILE ¬ODD(VAL) DO                                                    PAS
         BEGIN VAL := VAL DIV 2;  E1 := E1 + 1  END;                           PAS
         IF VAL = 1 THEN                                                       PAS
         BEGIN OPT := PUREP;  EXP1 := E1  END ELSE                             PAS
         BEGIN VAL := VAL DIV 2;  E2 := 1;                                     PAS
            IF ODD(VAL) THEN                                                   PAS
            BEGIN                                                              PAS
               REPEAT VAL := VAL DIV 2;  E2 := E2 + 1;                         PAS
               UNTIL ¬ODD(VAL);                                                PAS
               IF VAL > 0 THEN OPT := NOOPT ELSE                               PAS
               BEGIN OPT := NEGP;                                              PAS
                  EXP2 := E2;  EXP1 := E1;                                     PAS
               END;                                                            PAS
            END ELSE                                                           PAS
            BEGIN                                                              PAS
               REPEAT VAL := VAL DIV 2;  E2 := E2 + 1;                         PAS
               UNTIL ODD(VAL);                                                 PAS
               IF VAL > 1 THEN OPT := NOOPT  ELSE                              PAS
               BEGIN OPT := POSP;                                              PAS
```

LIST OF CONTROL, ACTIVE, AND/OR INACTIVE CARDS IN PASCSYM

000018

```
                    EXP2 := E2;  EXP1 := E1;                                    PAS
                END;                                                            PAS
            END;                                                                PAS
        END;                                                                    PAS
      END ELSE OPT := NOOPT;                                                    PAS
    END ↱MULOPT↴;                                                               PAS
                                                                                PAS
    FUNCTION LOG2(VAL1:INTEGER) : SHRTINT;                                      PAS
      VAR E : SHRTINT;  VAL : INTEGER;                                          PAS
    BEGIN E := 1;                                                               PAS
      VAL := VAL1;                                                              PAS
      WHILE VAL > 0 DO                                                          PAS
      BEGIN VAL := VAL DIV 2;  E := E + 1  END;                                 PAS
      LOG2 := E;                                                                PAS
    END ↱LOG2↴;                                                                 PAS
                                                                                PAS
                                                                                PAS
    PROCEDURE SKIP(FNO : SHRTINT) ;                                             PAS
    BEGIN                                                                       PAS
      WHILE (ERRCL[NO] = IRRELSY)∧(FNO ≠ NO) DO                                 PAS
      IF (NO = 38)∧(CL = 2) THEN  ↱RECORD↴                                      PAS
      BEGIN REPEAT INSYMBOL; SKIP(49);                                          PAS
        UNTIL ¬(NO IN [16,26]);                                                 PAS
        IF NO = 22 THEN INSYMBOL ;                                              PAS
      END ELSE INSYMBOL;                                                        PAS
    END  ↱ SKIP ↴  ;                                                            PAS
                                                                                PAS
↱ ------------------------------------------------------------------------ ↴   PAS
                                                                                PAS
    PROCEDURE PACKANDNORM(FRP : RG3) ;                                          PAS
    BEGIN GEN15(27B,FRP,0,FRP) ; GEN15(24B,FRP,0,FRP) END ;                     PAS
                                                                                PAS
    PROCEDURE JUMPTO(FADR : ADDRESS) ;                                          PAS
    BEGIN NOOP ; GEN30(71B,7,0,IC+1) ; GEN30(04B,0,0,FADR) END ;                PAS
                                                                                PAS
    PROCEDURE LDCST(FVAL : INTEGER) ;                                           PAS
     ↱ ENTERS CONSTANT WITH VALUE FVAL INTO CONSTANTTABLE CSTTB IFF             PAS
      NOT YET PRESENT, ELSE CHAINS OCCURRENCES OF FVAL IN CODE                  PAS
      (THROUGH UNDLAB)   ↴                                                      PAS
      VAR IT : SHRTINT;                                                         PAS
    BEGIN                                                                       PAS
      RP := RP + 1 ; IF RP = 6 THEN ERROR(33) ;                                 PAS
      GEN30(51B,RP,0,0) ;                                                       PAS
      FOR IT := 1 TO LCX DO                                                     PAS
      WITH CSTTB[IT] DO                                                         PAS
      IF VALU = FVAL THEN                                                       PAS
      BEGIN                                                                     PAS
        IF CHNIX = 0 THEN ERROR(75) ELSE                                        PAS
        BEGIN                                                                   PAS
          WITH UNDLAB[CHNIX] DO                                                 PAS
          BEGIN                                                                 PAS
            IT1 := SUCC ; SUCC := INX ;                                         PAS
```

000019

LIST OF CONTROL, ACTIVE, AND/OR INACTIVE CARDS IN PASCSYM

```
                INX := CHNIX ; PLACE := CA ; LFTSH := CP                                PA
              END ;                                                                     PA
              CHNIX := IT1                                                              PA
            END ;                                                                       PA
            GOTO 10                                                                     PA
        END ;                                                                           PA
        IF LCX = CSTMAX THEN BEGIN ERROR(84) ; GOTO 10 END ;                            PA
        IF CHNIX = 0 THEN BEGIN ERROR(75) ; GOTO 10 END ;                               PA
        LCX := LCX + 1 ;                                                                PA
        WITH CSTTB[LCX] DO                                                              PA
        BEGIN VALU := FVAL ; INX := CHNIX END ;                                         PA
        WITH UNDLAB[CHNIX] DO                                                           PA
        BEGIN IT1 := SUCC ; SUCC := 0 ; PLACE := CA ; LFTSH := CP END ;                 PA
        CHNIX := IT1 ;                                                                  PA
10:END  ↱ LDCST ↳  ;                                                                    PA
                                                                                        PA
    PROCEDURE MULTCODE(FRP : RG3) ;                                                     PA
    ↱GENERATES CODE FOR REAL MULTIPLICATION, ASSUMING FIRST OPERAND                     PA
    IN X-RP, THE OTHER IN X-FRP; RESULT IN X-RP↳                                        PA
    BEGIN IF ROUNDING THEN GEN15(41B,RP,RP,FRP)                                         PA
                      ELSE GEN15(40B,RP,RP,FRP)                                         PA
    END ↱MULTCODE↳ ;                                                                    PA
                                                                                        PA
    PROCEDURE CHECKBNDS(FRP: RG3; FMIN,FMAX: INTEGER; FADR: ADDRESS) ;                  PA
    ↱DONΞT USE WITH FRP IN [0,7]↳                                                       PA
    BEGIN GEN30(71B,7,0,IC) ;                                                           PA
       IF FMIN ≠ 0 THEN                                                                 PA
       BEGIN IF ABS(FMIN) ≥ TWOTO17 THEN LDCST(FMIN) ELSE                               PA
          BEGIN RP := RP + 1 ; GEN30(71B,RP,0,FMIN) END ;                               PA
          GEN15(37B,0,FRP,RP) ; RP := RP - 1 ;                                          PA
       END ;                                                                            PA
       IF ABS(FMAX) ≥ TWOTO17 THEN LDCST(FMAX) ELSE                                     PA
       BEGIN RP := RP + 1 ; GEN30(71B,RP,0,FMAX) END ;                                  PA
       GEN15(37B,RP,RP,FRP) ;                                                           PA
       IF FMIN ≠ 0 THEN GEN15(12B,0,RP,0) ELSE GEN15(12B,0,RP,FRP) ;                    PA
       GEN30(03B,3,0,FADR) ; RP := RP - 1 ;                                             PA
    END ↱CHECKBNDS↳ ;                                                                   PA
                                                                                        PA
    PROCEDURE LOADBASE(FRP, FLEVEL : RG3) ;                                             PA
      VAR I : SHRTINT ;                                                                 PA
    BEGIN GEN15(56B,FRP,5,0) ;                                                          PA
      FOR I := 1 TO LEVEL - FLEVEL - 1 DO GEN15(53B,FRP,FRP,0) ;                        PA
    END ↱LOADBASE↳ ;                                                                    PA
                                                                                        PA
    PROCEDURE TRANSFER(FATTR : ATTR ; FRP : SHRTINT) ;                                  PA
    ↱DEPENDING ON THE VALUE OF FRP, TRANSFER GENERATES CODE FOR EITHER                  PA
     LOADING THE EXPRESSION DESCRIBED BY FATTR INTO REGISTER X-FRP                      PA
     OR FOR STORING X-FRP AT LOCATION DESCRIBED BY FATTR (TAKING INTO                   PA
     ACCOUNT CDC 6000 HARDWARE FEATURES)↳                                               PA
       VAR LRP, REG : SHRTINT;  AT : ADDRESS;  BT : BOOLEAN;                            PA
    BEGIN WITH FATTR DO                                                                 PA
      IF TYPTR ≠ NIL THEN                                                               PA
```

LIST OF CONTROL, ACTIVE, AND/OR INACTIVE CARDS IN PASCSYM 000020

```
    CASE KIND OF                                                                PA
    VARBL: BEGIN IF (FRP > 5)^PCKD THEN                                         PA
             BEGIN PCKD := FALSE ; TRANSFER(FATTR,1) ;                          PA
               IT := BITADR + BITSZ ; BT := IT ≠ WORDLENGTH ;                   PA
               IF BT THEN GEN15(20B,1,0,IT) ;                                   PA
               GEN15(43B,0,0,WORDLENGTH-BITSZ) ; GEN15(11B,1,0,1) ;             PA
               GEN15(15B,FRP,FRP,0) ; GEN15(12B,FRP,FRP,1) ;                    PA
               IF BT THEN GEN15(20B,FRP,0,WORDLENGTH-IT) ;                      PA
               GEN15(54B,FRP,1,0)                                               PA
             END ELSE                                                           PA
             BEGIN LRP := FRP ;                                                 PA
               IF ACCESS = DRCT THEN                                            PA
               BEGIN IF FRP < 5 THEN                                            PA
                 BEGIN RP := RP + 1 ; LRP := RP END ELSE                        PA
                 IF FRP = 5 THEN ERROR(33) ;                                    PA
                 IF BREG IN [0,LEVEL] THEN                                      PA
                 BEGIN IF BREG ≠ 0 THEN REG := 5 ELSE REG := 0 ;                PA
                   GEN30(51B,LRP,REG,DPLMT) ;                                   PA
                 END ELSE                                                       PA
                 BEGIN LOADBASE(5,BREG) ; GEN30(52B,LRP,5,DPLMT) END ;          PA
               END ELSE                                                         PA
               IF ACCESS = INXD THEN                                            PA
               BEGIN IF BREG IN [0,LEVEL] THEN                                  PA
                 BEGIN IF BREG ≠ 0 THEN REG := 5 ELSE REG := 0 ;                PA
                   IF DPLMT = 0 THEN GEN15(53B,LRP,RP,REG) ELSE                 PA
                   BEGIN IF REG = 0 THEN GEN30(52B,LRP,RP,DPLMT) ELSE           PA
                     BEGIN GEN30(62B,7,RP,DPLMT) ;                              PA
                       GEN15(56B,LRP,7,5)                                       PA
                     END                                                        PA
                   END                                                          PA
                 END ELSE                                                       PA
                 BEGIN GEN30(62B,7,RP,DPLMT) ;                                  PA
                   LOADBASE(5,BREG) ; GEN15(53B,LRP,5,7)                        PA
                 END                                                            PA
               END ELSE                                                         PA
               BEGIN IF DPLMT = 0 THEN GEN15(53B,LRP,RP,0)                      PA
                                  ELSE GEN30(52B,LRP,RP,DPLMT)                  PA
               END ;                                                            PA
               IF PCKD THEN                                                     PA
               BEGIN IF BITADR ≠ 0 THEN GEN15(20B,LRP,0,BITADR) ;               PA
                 GEN15(21B,LRP,0,WORDLENGTH-BITSZ) ; PCKD := FALSE              PA
               END                                                              PA
             END                                                                PA
           END ;                                                                PA
    SVAL : BEGIN IF FRP < 5 THEN                                                PA
             BEGIN RP := RP + 1 ; LRP := RP END ELSE                            PA
             IF FRP > 5 THEN LRP := FRP ELSE ERROR(33) ;                        PA
             IF VAL = 0 THEN GEN15(13B,LRP,0,0)                                 PA
                        ELSE GEN30(71B,LRP,0,VAL)                               PA
           END ;                                                                PA
    LVAL : BEGIN IF CTERM ≠ 0 THEN                                              PA
             BEGIN GEN30(71B,0,0,CTERM) ; GEN15(36B,FRP,RP,0) END ELSE          PA
```

```
            IF RP ≠ FRP THEN BXIXJ(FRP,RP)                                    PA
          END ;                                                               PA
   LCOND: BEGIN IF ARITH THEN                                                 PA
            BEGIN IF JMP ≤ 1 THEN                                             PA
              BEGIN GEN15(43B,FRP,0,0) ; AT := IC + 1 ;                       PA
                GEN30(03B,JMP,0,AT) ; GEN30(71B,FRP,0,1) ; NOOP ;             PA
              END ELSE                                                        PA
              BEGIN GEN15(43B,0,0,1) ;                                        PA
                IF JMP = 2 THEN GEN15(11B,FRP,0,RP)                           PA
                           ELSE GEN15(15B,FRP,0,RP) ;                         PA
                GEN15(20B,FRP,0,1)                                            PA
              END                                                             PA
            END ELSE                                                          PA
            BEGIN IF JMP ≠ 0 THEN                                             PA
              BEGIN GEN15(43B,0,0,59) ; GEN15(17B,FRP,0,RP) END ELSE          PA
              IF RP ≠ FRP THEN BXIXJ(FRP,RP)                                  PA
            END                                                               PA
          END                                                                 PA
    END ↱CASE KIND↳                                                           PA
  END ↱TRANSFER↳ ;                                                            PA
                                                                              PA
  PROCEDURE LOADADR(FATTR : ATTR ; FRP : SHRTINT) ;                           PA
  ↱GENERATES CODE FOR LOADING THE ADDRESS OF THE VARIABLE DESCRIBED            PA
   BY FATTR INTO X-FRP↳                                                       PA
    VAR REG : SHRTINT ;                                                       PA
  BEGIN WITH FATTR DO                                                         PA
    IF TYPTR ≠ NIL THEN                                                       PA
    IF PCKD THEN ERROR(87) ELSE                                               PA
    CASE ACCESS OF                                                            PA
    DRCT  : IF BREG IN [0,LEVEL] THEN                                         PA
            BEGIN IF BREG ≠ 0 THEN REG := 5 ELSE REG := 0 ;                   PA
              GEN30(71B,FRP,REG,DPLMT)                                        PA
            END ELSE                                                          PA
            BEGIN LOADBASE(5,BREG) ; GEN30(72B,FRP,5,DPLMT) END ;             PA
    INDRCT: IF DPLMT ≠ 0 THEN GEN30(72B,FRP,RP,DPLMT) ELSE                    PA
            IF RP ≠ FRP THEN BXIXJ(FRP,RP) ;                                  PA
    INXD  : IF BREG IN [0,LEVEL] THEN                                         PA
            BEGIN IF BREG ≠ 0 THEN REG := 5 ELSE REG := 0 ;                   PA
              IF DPLMT = 0 THEN GEN15(73B,FRP,RP,REG) ELSE                    PA
              BEGIN IF REG = 0 THEN GEN30(72B,FRP,RP,DPLMT) ELSE              PA
                BEGIN GEN30(62B,7,RP,DPLMT) ; GEN15(76B,FRP,7,5) END ;        PA
              END                                                             PA
            END ELSE                                                          PA
            BEGIN GEN30(62B,7,RP,DPLMT) ;                                     PA
              LOADBASE(5,BREG) ; GEN15(73B,FRP,5,7)                           PA
            END ;                                                             PA
    END                                                                       PA
  END ↱LOADADR↳ ;                                                             PA
                                                                              PA
   PROCEDURE GENJP(FJPADDR : ADDRESS) ;                                       PA
    ↱ ACCORDING TO THE GLOBAL ATTRIBUTE RECORD GATTR (DEFINING AN             PA
      EXPRESSION), THIS PROCEDURE PERFORMS CODE GENERATION FOR A              PA
```

LIST OF CONTROL, ACTIVE, AND/OR INACTIVE CARDS IN PASCSYM 000022

```
      CONDITIONAL JUMP (ADDRESS = FJPADDR), DEPENDING ON THE TRUTH                PA
      VALUE OF THE (NECESSARILY) BOOLEAN EXPRESSION ↓                             PA
    BEGIN                                                                         PA
      WITH GATTR DO                                                               PA
      IF TYPTR ≠ BOOLPTR THEN ERROR(57) ELSE                                      PA
      BEGIN                                                                       PA
        IF KIND ≤ SVAL THEN TRANSFER(GATTR,RP) ;                                  PA
        IF KIND = LCOND THEN                                                      PA
          IF (JMP ≤ 1) ∧ ARITH THEN GEN30(03B,JMP,0,FJPADDR) ELSE                 PA
          GEN30(03B,JMP,1,FJPADDR)                                                PA
        ELSE GEN30(03B,0,RP,FJPADDR)                                              PA
      END                                                                         PA
    END ↱ GENJP ↓ ;                                                               PA
                                                                                  PA
  PROCEDURE ADDRESSVAR(FCTP : CTP ; VAR FATTR : ATTR) ;                           PA
  ↱GENERATES CODE TO ADDRESS THE QUANTITY WITH CTPTR = FCTP AND                   PA
   BUILDS UP ITS ATTRIBUTES IN FATTR↓                                             PA
  BEGIN WITH FCTP↑, FATTR DO                                                      PA
    BEGIN KIND := VARBL ;                                                         PA
      IF KLASS = VARS THEN                                                        PA
      BEGIN TYPTR := VTYPE ; PCKD := FALSE ;                                      PA
        IF VKIND = ACTUAL THEN                                                    PA
        BEGIN ACCESS := DRCT ; DPLMT := VADDR ; BREG := VLEVEL ;                  PA
        END ELSE                                                                  PA
        BEGIN RP := RP + 1 ; IF RP = 6 THEN ERROR(33) ;                           PA
          IF VLEVEL = LEVEL THEN GEN30(51B,RP,5,VADDR) ELSE                       PA
          BEGIN LOADBASE(RP,VLEVEL) ; GEN30(52B,RP,RP,VADDR) END ;                PA
          ACCESS := INDRCT ; DPLMT := 0 ;                                         PA
        END ;                                                                     PA
      END ELSE                                                                    PA
      IF KLASS = FIELD THEN                                                       PA
      BEGIN TYPTR := FLDTYPE ;                                                    PA
        WITH DISPLAY[DISX] DO                                                     PA
        BEGIN IF OCCUR = CWITH THEN                                               PA
          BEGIN ACCESS := DRCT ; BREG := CLEV ; DPLMT := FLDADDR + CDSPL          PA
          END ELSE                                                                PA
          BEGIN RP := RP + 1 ; IF RP = 6 THEN ERROR(33) ;                         PA
            GEN30(51B,RP,5,VDSPL) ;                                               PA
            ACCESS := INDRCT ; DPLMT := FLDADDR ;                                 PA
          END ;                                                                   PA
          IF BITWIDTH ≠ 0 THEN                                                    PA
          BEGIN PCKD := TRUE ; BITADR := BITDISPL ; BITSZ := BITWIDTH             PA
          END ELSE PCKD := FALSE                                                  PA
        END                                                                       PA
      END ELSE                                                                    PA
      BEGIN TYPTR := PROCTYPE ; ACCESS := DRCT ; BREG := PROCLEVEL + 1 ;          PA
        DPLMT := 2 ; PCKD := FALSE ;                                              PA
        IF PROCKIND = FORMAL THEN ERROR(81) ELSE                                  PA
        IF LEVEL ≠ BREG THEN ERROR(85)                                            PA
      END                                                                         PA
    END ↱WITH FCTP↑, FATTR↓ ;                                                     PA
  END ↱ADDRESSVAR↓ ;                                                              PA
```

LIST OF CONTROL, ACTIVE, AND/OR INACTIVE CARDS IN PASCSYM

```
                                                                                PA
                                                                                PA
↱ ------------------------------------------------------------------------ ↓    PA
                                                                                PA
  PROCEDURE STATEMENT  ; FORWARD ;                                              PA
                                                                                PA
  PROCEDURE EXPRESSION ; FORWARD ;                                              PA
                                                                                PA
  PROCEDURE PASSPARAMS ; FORWARD ;                                              PA
                                                                                PA
  PROCEDURE VARIABLE ;                                                          PA
    VAR LATTR : ATTR ;                                                          PA
  BEGIN ADDRESSVAR(CTPTR,LATTR) ;                                               PA
    INSYMBOL ;                                                                  PA
    WHILE NO IN [11,17,18] DO ↱[,.,↑↓                                           PA
    IF NO = 11 THEN ↱[↓                                                         PA
    BEGIN                                                                       PA
      REPEAT WITH LATTR DO                                                      PA
        IF TYPTR ≠ NIL THEN                                                     PA
        IF TYPTR↑.FORM ≠ ARRAYS THEN                                            PA
        BEGIN ERROR(34) ; TYPTR := NIL END ;                                    PA
        INSYMBOL ; EXPRESSION ;                                                 PA
        IF GATTR.TYPTR ≠ NIL THEN                                               PA
        BEGIN IF GATTR.TYPTR↑.FORM > SYMBOLIC THEN ERROR(35) ;                  PA
          IF LATTR.TYPTR ≠ NIL THEN                                             PA
          BEGIN PT := LATTR.TYPTR↑.INXTYPE ;                                    PA
            IF ((GATTR.TYPTR↑.FORM = SYMBOLIC)∨(PT↑.FORM = SYMBOLIC))∧          PA
            (GATTR.TYPTR ≠ PT) THEN ERROR(36) ;                                 PA
            WITH GATTR DO                                                       PA
            IF KIND = SVAL THEN                                                 PA
            BEGIN IF (LATTR.TYPTR↑.LO > VAL)∨(LATTR.TYPTR↑.HI < VAL)            PA
              THEN ERROR(99) ;                                                  PA
              IT := VAL                                                         PA
            END ELSE                                                            PA
            IF KIND = LVAL THEN IT := CTERM ELSE                                PA
            BEGIN TRANSFER(GATTR,RP) ; IT := 0 END ;                            PA
            WITH LATTR, TYPTR↑ DO                                               PA
            BEGIN DPLMT := DPLMT + (IT - LO)*AELTYPE↑.SIZE ;                    PA
              IF GATTR.KIND ≠ SVAL THEN                                         PA
              BEGIN IF INXCHECK THEN CHECKBNDS(RP,LO-IT,HI-IT,INXERR) ;         PA
                IF OPTTYP = NOOPT THEN                                          PA
                BEGIN GEN30(71B,6,0,AELTYPE↑.SIZE) ;                            PA
                  GEN15(42B,RP,RP,6)                                            PA
                END ELSE                                                        PA
                BEGIN IF EXP1 ≠ 0 THEN GEN15(20B,RP,0,EXP1) ;                   PA
                  IF OPTTYP ≠ PUREP THEN                                        PA
                  BEGIN GEN15(10B,0,RP,0) ; GEN15(20B,RP,0,EXP2) ;              PA
                    IF OPTTYP = POSP THEN GEN15(36B,RP,RP,0)                    PA
                                     ELSE GEN15(37B,RP,RP,0)                    PA
                  END                                                           PA
                END ;                                                           PA
                IF ACCESS = DRCT THEN ACCESS := INXD ELSE                       PA
```

LIST OF CONTROL, ACTIVE, AND/OR INACTIVE CARDS IN PASCSYM 000024

```
              BEGIN RP := RP - 1 ; GEN15(36B,RP,RP,RP+1) END                    PAS
            END                                                                 PAS
          END ;                                                                 PAS
        END ↱IF LATTR.TYPTR ≠ NIL↓                                              PAS
      END ↱IF GATTR.TYPTR ≠ NIL↓ ;                                              PAS
      IF LATTR.TYPTR ≠ NIL THEN LATTR.TYPTR := LATTR.TYPTR↑.AELTYPE ;           PAS
    UNTIL NO ≠ 15 ↱,↓ ;                                                         PAS
    IF NO ≠ 12 THEN ↱]↓ ERROR(37) ELSE INSYMBOL                                 PAS
  END ↱IF NO = 11↓ ELSE                                                         PAS
  IF NO = 17 THEN ↱.↓                                                           PAS
  BEGIN INSYMBOL ;                                                              PAS
    IF NO = 1 THEN ↱ID↓                                                         PAS
    BEGIN IF LATTR.TYPTR ≠ NIL THEN                                             PAS
      BEGIN IF LATTR.TYPTR↑.FORM = RECORDS THEN                                 PAS
        WITH LATTR DO                                                           PAS
        BEGIN SRCHREC(TYPTR↑.FSTFLD) ;                                          PAS
          IF CTPTR = NIL THEN                                                   PAS
          BEGIN ERROR(39) ; CTPTR := UNDECPTR END ;                             PAS
          WITH CTPTR↑ DO                                                        PAS
          BEGIN TYPTR := FLDTYPE ; DPLMT := DPLMT + FLDADDR ;                   PAS
            IF BITWIDTH ≠ 0 THEN                                                PAS
            BEGIN BITADR := BITOISPL ; BITSZ := BITWIDTH ;                      PAS
              PCKD := TRUE                                                      PAS
            END ELSE PCKD := FALSE                                              PAS
          END                                                                   PAS
        END ELSE                                                                PAS
        BEGIN ERROR(38) ; LATTR.TYPTR := NIL END ;                              PAS
      END ;                                                                     PAS
      INSYMBOL                                                                  PAS
    END ELSE                                                                    PAS
    BEGIN ERROR(41) ; LATTR.TYPTR := NIL END                                    PAS
  END ↱IF NO = 17↓ ELSE ↱↑↓                                                     PAS
  BEGIN WITH LATTR DO                                                           PAS
    IF TYPTR ≠ NIL THEN                                                         PAS
    BEGIN TRANSFER(LATTR,RP) ; ACCESS := INDRCT ; DPLMT := 0 ;                  PAS
      IF TYPTR↑.FORM = FILES THEN TYPTR := TYPTR↑.FELTYPE ELSE                  PAS
      IF TYPTR↑.FORM = POINTER THEN TYPTR := TYPTR↑.ELTYPE ELSE                 PAS
      BEGIN ERROR(40) ; TYPTR := NIL END                                        PAS
    END ;                                                                       PAS
    INSYMBOL                                                                    PAS
  END ↱↑↓ ;                                                                     PAS
  GATTR := LATTR                                                                PAS
END ↱VARIABLE↓ ;                                                                PAS
                                                                                PAS
PROCEDURE FACTOR ;                                                              PAS
  VAR LATTR : ATTR ; LPTR : CTP ; LRP, LB6 : SHRTINT ;                          PAS
      AT : ADDRESS;  LPSVAL, IT2 : INTEGER;                                     PAS
                                                                                PAS
  PROCEDURE ELEMENT ;                                                           PAS
  ↱WORKS UP THE NEXT ELEMENT IN THE SET BEEING ANALYSED↓                        PAS
  BEGIN                                                                         PAS
    EXPRESSION ;                                                                PAS
```

LIST OF CONTROL, ACTIVE, AND/OR INACTIVE CARDS IN PASCSYM 000025

```
      IF GATTR.TYPTR ≠ NIL THEN                                                  PAS
      IF GATTR.TYPTR↑.FORM > SYMBOLIC THEN ERROR(35) ELSE                        PAS
      IF (GATTR.TYPTR = REALPTR)∨(GATTR.TYPTR = ALFAPTR) THEN                    PAS
      ERROR(62) ELSE                                                             PAS
      BEGIN IF LATTR.TYPTR ≠ NIL THEN                                            PAS
        IF ((LATTR.TYPTR↑.FORM ≠ NUMERIC)∨(GATTR.TYPTR↑.FORM ≠                   PAS
        NUMERIC))∧(LATTR.TYPTR ≠ GATTR.TYPTR) THEN ERROR(73) ;                   PAS
        LATTR.TYPTR := GATTR.TYPTR ;                                             PAS
        IF GATTR.KIND = SVAL THEN INSERT(1,GATTR.VAL,LPSVAL) ELSE                PAS
        BEGIN TRANSFER(GATTR,RP) ; GEN15(63B,7,RP,0) ;                           PAS
          IF LATTR.KIND = SVAL THEN                                              PAS
          BEGIN GEN30(71B,RP,0,1) ; GEN15(22B,RP,7,RP) ;                         PAS
            LATTR.KIND := LVAL ; LATTR.CTERM := 0                                PAS
          END ELSE                                                               PAS
          BEGIN GEN30(71B,0,0,1) ; GEN15(22B,0,7,0) ;                            PAS
            RP := RP - 1 ; GEN15(12B,RP,RP,0) ;                                  PAS
          END                                                                    PAS
        END                                                                      PAS
      END ;                                                                      PAS
    END ↱ELEMENT↳ ;                                                              PAS
                                                                                 PAS
  BEGIN IF NO = 1 THEN                                    ↱IDENTIFIER↳           PAS
    BEGIN SEARCH ;                                                               PAS
      IF CTPTR = NIL THEN                                                        PAS
      BEGIN ERROR(31) ; CTPTR := UNDECPTR END ;                                  PAS
      CASE CTPTR↑.KLASS OF                                                       PAS
TYPES: BEGIN ERROR(45) ; GATTR.TYPTR := NIL ; INSYMBOL END ;                     PAS
KONST: WITH GATTR, CTPTR↑ DO                                                     PAS
        BEGIN TYPTR := CONTYPE ;                                                 PAS
          IF CONKIND = ACTUAL THEN                                               PAS
          BEGIN IF ABS(VALUES) ≥ TWOTO17 THEN                                    PAS
            BEGIN LOCST(VALUES) ; KIND := LVAL ; CTERM := 0 END ELSE             PAS
            BEGIN KIND := SVAL ; VAL := VALUES END                               PAS
          END ELSE ↱FORMAL↳                                                      PAS
          BEGIN KIND := VARBL ; ACCESS := DRCT ;                                 PAS
            BREG := CLEVEL ; DPLMT := CADDR ; PCKD := FALSE ;                    PAS
          END ;                                                                  PAS
          INSYMBOL                                                               PAS
        END ;                                                                    PAS
PROC : BEGIN INSYMBOL ;                                                          PAS
          IF (CTPTR↑.PROCTYPE = NIL)∨(CTPTR↑.PROCTYPE = CTPTR) THEN              PAS
          BEGIN ERROR(46) ; GATTR.TYPTR := NIL END ELSE                          PAS
          IF NO ≠ 9 THEN ↱(↳                                                     PAS
          BEGIN ERROR(79) ; GATTR.TYPTR := NIL END ELSE                          PAS
          IF CTPTR ≤ PREDEFP THEN                          ↱**INLINE FCT**↳      PAS
          BEGIN LPTR := CTPTR ; INSYMBOL ; EXPRESSION ;                          PAS
            IF GATTR.TYPTR ≠ NIL THEN                                            PAS
            CASE LPTR↑.SEGSIZE OF                                                PAS
    ↱ODD↳ 1: BEGIN IF GATTR.TYPTR↑.FORM ≠ NUMERIC THEN ERROR(47) ;               PAS
                TRANSFER(GATTR,RP) ;                                             PAS
                IF GATTR.TYPTR↑.MIN ≥ 0 THEN GEN15(20B,RP,0,59) ELSE             PAS
                BEGIN GEN15(10B,0,RP,0) ; GEN15(20B,0,0,59) ;                    PAS
```

LIST OF CONTROL, ACTIVE, AND/OR INACTIVE CARDS IN PASCSYM 000026

```
                GEN15(13B,RP,RP,0) ;                                    PAS
              END ;                                                     PAS
              WITH GATTR DO                                             PAS
              BEGIN TYPTR := BOOLPTR ; KIND := LCOND ; JMP := 2 ;       PAS
                ARITH := TRUE                                           PAS
              END                                                       PAS
            END ;                                                       PAS
  ↱INT↓ 2: BEGIN IF GATTR.TYPTR↑.FORM > POWER THEN ERROR(44) ;          PAS
              GATTR.TYPTR := INTPTR                                     PAS
            END ;                                                       PAS
  ↱CHR↓ 3: BEGIN IF GATTR.TYPTR↑.FORM ≠ NUMERIC THEN ERROR(47) ;        PAS
              IF ASSCHECK THEN                                          PAS
              BEGIN TRANSFER(GATTR,RP) ;                                PAS
                WITH GATTR DO                                           PAS
                BEGIN KIND := LVAL ; CTERM := 0 END ;                   PAS
                CHECKBNDS(RP,0,63,ASSERR)                               PAS
              END ;                                                     PAS
              GATTR.TYPTR := CHARPTR                                    PAS
            END ;                                                       PAS
  ↱EOF↓ 4: BEGIN IF GATTR.TYPTR↑.FORM ≠ FILES THEN ERROR(44) ;          PAS
              WITH GATTR DO                                             PAS
              BEGIN DPLMT := DPLMT + 1 ; TRANSFER(GATTR,RP) ;           PAS
                TYPTR := BOOLPTR ; KIND := LCOND ; JMP := 2 ;           PAS
                ARITH := TRUE                                           PAS
              END                                                       PAS
            END ;                                                       PAS
  ↱ABS↓ 5: BEGIN IF (GATTR.TYPTR ≠ REALPTR)∧(GATTR.TYPTR↑.FORM ≠        PAS
              NUMERIC) THEN ERROR(44) ;                                 PAS
              TRANSFER(GATTR,RP) ; GEN15(10B,0,RP,0) ;                  PAS
              GEN15(21B,0,0,73B) ; GEN15(13B,RP,0,RP) ;                 PAS
              WITH GATTR DO                                             PAS
              BEGIN IF TYPTR ≠ REALPTR THEN TYPTR := INTPTR ;           PAS
                KIND := LVAL ; CTERM := 0                               PAS
              END                                                       PAS
            END ;                                                       PAS
  ↱SQR↓ 6: BEGIN IF (GATTR.TYPTR ≠ REALPTR)∧(GATTR.TYPTR↑.FORM ≠        PAS
              NUMERIC) THEN ERROR(44) ;                                 PAS
              TRANSFER(GATTR,RP) ;                                      PAS
              WITH GATTR DO                                             PAS
              BEGIN IF TYPTR = REALPTR THEN MULTCODE(RP) ELSE           PAS
                BEGIN GEN15(42B,RP,RP,RP) ; TYPTR := INTPTR END ;       PAS
                KIND := LVAL ; CTERM := 0 ;                             PAS
              END                                                       PAS
            END ;                                                       PAS
  ↱TRC↓ 7: BEGIN IF GATTR.TYPTR ≠ REALPTR THEN ERROR(44) ;              PAS
              TRANSFER(GATTR,RP) ;                                      PAS
              GEN15(26B,RP,7,RP) ; GEN15(22B,RP,7,RP) ;                 PAS
              GEN15(13B,0,0,0) ; GEN15(36B,RP,RP,0) ;                   PAS
              WITH GATTR DO                                             PAS
              BEGIN TYPTR := INTPTR ; KIND := LVAL ; CTERM := 0 END     PAS
            END ;                                                       PAS
  ↱PRE↓ 8,                                                              PAS
```

```
      ↱SUC↓ 9: IF (GATTR.TYPTR↑.FORM > SYMBOLIC)∨(GATTR.TYPTR =                  PAS
               REALPTR)∨(GATTR.TYPTR = ALFAPTR) THEN                             PAS
               BEGIN ERROR(44) ; GATTR.TYPTR := NIL END ELSE                     PAS
               BEGIN TRANSFER(GATTR,RP) ;↴                                       PAS
                 WITH GATTR.TYPTR↑ DO                                            PAS
                 IF FORM = NUMERIC THEN                                          PAS
                 BEGIN IT1 := MIN ; IT2 := MAX END ELSE                          PAS
                 BEGIN IT1 := 0 ; IT2 := FCONST↑.VALUES END ;                    PAS
                 IT := LPTR↑.SEGSIZE ;                                           PAS
                 IF (ABS(IT1) ≥ TWOTO17)∨(ABS(IT2) ≥ TWOTO17) THEN               PAS
                 BEGIN GEN30(71B,0,0,1) ;                                        PAS
                   GEN15(47B-IT,RP,RP,0) ;   ↱36B OR 37B↓                       PAS
                 END ELSE GEN30(72B,RP,RP,2*IT-17) ;  ↱+1 OR -1↓                 PAS
                 IF ASSCHECK^(GATTR.TYPTR ≠ INTPTR) THEN                         PAS
                 CHECKBNDS(RP,IT1,IT2,ASSERR) ;                                  PAS
                 WITH GATTR DO                                                   PAS
                 BEGIN KIND := LVAL ; CTERM := 0 END                             PAS
               END                                                               PAS
             END ↱CASE LPTR↑.SEGSIZE OF↓ ;                                       PAS
             IF NO ≠ 10 THEN ↱)↓                                                 PAS
             BEGIN ERROR(48) ; GATTR.TYPTR := NIL END ELSE INSYMBOL              PAS
           END ↱IF CTPTR ≤ PREDEFP↓ ELSE              ↱**EXTERNAL FCT**↓       PAS
           BEGIN FOR IT := 1 TO RP DO                 ↱**USERDEF. FCT**↓       PAS
             BEGIN BXIXJ(6,IT) ; GEN30(51B,6,5,LC+TCT) ;                       PAS
               TCT := TCT + 1 ; IF TCT > TMAX THEN TMAX := TCT ;               PAS
             END ;                                                             PAS
             IF CTPTR ≤ EXTPTR THEN  ↱ SET BIT TO INFORM MONITOR ↓             PAS
             WITH PARMLIST DO EXTFLAGS := EXTFLAGS∨[CTPTR↑.SEGSIZE] ;          PAS
             LRP := RP ; LB6 := B6DPL ; LPTR := CTPTR↑.PROCTYPE ;              PAS
             IF LB6 ≠ 3 THEN GEN30(61B,6,6,LB6) ; RP := 0 ;                    PAS
             PASSPARAMS ;                                                      PAS
             IF LB6 ≠ 3 THEN GEN30(61B,6,6,-LB6) ;                             PAS
             B6DPL := LB6 ;                                                    PAS
             FOR IT := LRP DOWNTO 1 DO                                         PAS
             BEGIN TCT := TCT - 1 ; GEN30(51B,IT,5,LC+TCT) END ;               PAS
             IF LRP = 5 THEN ERROR(33) ELSE                                    PAS
             BEGIN RP := LRP + 1 ;                                             PAS
               IF LB6 = 3 THEN IT := 2 ELSE IT := LB6 + 2 ;                    PAS
               GEN30(51B,RP,6,IT) ;                                            PAS
             END ;                                                             PAS
             WITH GATTR DO                                                     PAS
             BEGIN TYPTR := LPTR ; KIND := LVAL ; CTERM := 0 END               PAS
           END                                                                 PAS
         END ;                                                                 PAS
VARS ,                                                                         PAS
FIELD : VARIABLE ;                                                             PAS
       END ↱CASE CTPTR↑.KLASS OF↓ ;                                            PAS
     END ↱IF NO = 1↓ ELSE                                                      PAS
     IF NO = 2 THEN                                        ↱CONSTANT↓          PAS
     BEGIN WITH GATTR DO                                                       PAS
       BEGIN IF ABS(IVAL) ≥ TWOTO17 THEN                                       PAS
         BEGIN LDCST(IVAL) ; KIND := LVAL ; CTERM := 0 END ELSE                PAS
```

LIST OF CONTROL, ACTIVE, AND/OR INACTIVE CARDS IN PASCSYM 000028

```
        BEGIN KIND := SVAL ; VAL := IVAL END ;                                  PAS
        CASE CL OF                                                              PAS
       1: TYPTR := INTPTR  ;                                                    PAS
       2: TYPTR := REALPTR ;                                                    PAS
       3: TYPTR := ALFAPTR ;                                                    PAS
       4: TYPTR := CHARPTR ;                                                    PAS
        END ;                                                                   PAS
      END ;                                                                     PAS
      INSYMBOL ;                                                                PAS
    END ↱IF NO = 2↓ ELSE                                                        PAS
    IF NO = 36 THEN                                          ↱NIL↓              PAS
    BEGIN WITH GATTR DO                                                         PAS
      BEGIN TYPTR := NILPTR ; KIND := SVAL ; VAL := NILVAL END ;                PAS
      INSYMBOL                                                                  PAS
    END ELSE                                                                    PAS
    IF NO = 5 THEN                                           ↱ ¬ ↓              PAS
    BEGIN INSYMBOL ; FACTOR ;                                                   PAS
      WITH GATTR DO                                                             PAS
      IF TYPTR ≠ NIL THEN                                                       PAS
      IF TYPTR = BOOLPTR THEN                                                   PAS
      BEGIN IF KIND = LCOND THEN                                                PAS
        BEGIN IF JMP ≤ 1 THEN JMP := 1 - JMP ELSE JMP := 5 - JMP ;              PAS
        END ELSE                                                                PAS
        BEGIN TRANSFER(GATTR,RP) ; GEN15(43B,0,0,59) ;                          PAS
          GEN15(17B,RP,0,RP) ; KIND := LVAL ; CTERM := 0                        PAS
        END                                                                     PAS
      END ELSE ERROR(50)                                                        PAS
    END ↱IF NO = 5↓ ELSE                                                        PAS
    IF NO = 9 THEN                                           ↱ ( ↓              PAS
    BEGIN INSYMBOL ; EXPRESSION ;                                               PAS
      IF NO ≠ 10 THEN ↱)↓                                                       PAS
      BEGIN ERROR(48) ; GATTR.TYPTR := NIL END ELSE INSYMBOL                    PAS
    END ELSE                                                                    PAS
    IF NO = 11 THEN                                          ↱ [ ↓              PAS
    BEGIN INSYMBOL ;                                                            PAS
      IF NO = 12 THEN ↱]↓                                                       PAS
      WITH GATTR DO                                                             PAS
      BEGIN TYPTR := LAMPTR ; KIND := SVAL ;                                    PAS
        VAL := 0 ; INSYMBOL                                                     PAS
      END ELSE                                                                  PAS
      BEGIN WITH LATTR DO                                                       PAS
        BEGIN TYPTR := NIL ; KIND := SVAL END ; LPSVAL := 0 ;                   PAS
        ELEMENT ;                                                               PAS
        WHILE NO = 15 DO ↱,↓                                                    PAS
        BEGIN INSYMBOL ; ELEMENT END ;                                          PAS
        IF LPSVAL ≠ 0 THEN                                                      PAS
        IF LPSVAL ≥ TWOTO17 THEN                                                PAS
        BEGIN LDCST(LPSVAL) ;                                                   PAS
          IF LATTR.KIND = LVAL THEN                                             PAS
          BEGIN RP := RP - 1 ; GEN15(12B,RP,RP,RP+1) END ELSE                   PAS
          LATTR.KIND := LVAL ; LATTR.CTERM := 0                                 PAS
        END ELSE                                                                PAS
```

LIST OF CONTROL, ACTIVE, AND/OR INACTIVE CARDS IN PASCSYM 000029

```
        IF LATTR.KIND = SVAL THEN LATTR.VAL := LPSVAL ELSE                    PAS
        BEGIN GEN30(71B,0,0,LPSVAL) ; GEN15(12B,RP,RP,0) END ;                PAS
        IF LATTR.TYPTR ≠ NIL THEN                                             PAS
        IF LATTR.TYPTR↑.FORM = NUMERIC THEN LATTR.TYPTR := PNUMPTR             PAS
        ELSE LATTR.TYPTR := LATTR.TYPTR↑.PWSET ;                              PAS
        IF NO ≠ 12 THEN ↱]↳                                                   PAS
        BEGIN ERROR(37) ; LATTR.TYPTR := NIL END ELSE INSYMBOL ;              PAS
        GATTR := LATTR                                                        PAS
      END ;                                                                   PAS
    END ↱IF NO = 11↳ ELSE                                                     PAS
    BEGIN ERROR(42) ; GATTR.TYPTR := NIL END                                  PAS
  END ↱FACTOR↳ ;                                                              PAS
                                                                              PAS
  PROCEDURE TERM ;                                                            PAS
  VAR LATTR : ATTR ; AT : ADDRESS ; BT1,BT2,BT3,BT4 : BOOLEAN ;               PAS
      LEXP1, LEXP2, LMOPCL : SHRTINT ; LOPT : OPTPWR ;                        PAS
  PROCEDURE CHECKDIV(FRP : RG3) ;                                             PAS
    ↱DONΞT USE WITH FRP = 7↳                                                  PAS
    BEGIN IF GATTR.KIND = SVAL THEN                                           PAS
      BEGIN IF GATTR.VAL = 0 THEN ERROR(102) ;                                PAS
      END ELSE                                                                PAS
      IF DIVCHECK THEN                                                        PAS
      BEGIN GEN30(71B,7,0,IC) ; GEN30(03B,0,FRP,DIVERR) END ;                 PAS
    END ↱DIVCHECK↳ ;                                                          PAS
  BEGIN FACTOR ;                                                              PAS
    IF NO = 6 THEN ↱MULOP↳                                                    PAS
    BEGIN LOPT := NOOPT;                                                      PAS
      IF (GATTR.KIND = SVAL)∧(CL = 1) THEN                                    PAS
      MULOPT(GATTR.VAL,LEXP1,LEXP2,LOPT) ;                                    PAS
      IF LOPT = NOOPT THEN                                                    PAS
      BEGIN TRANSFER(GATTR,RP) ;                                              PAS
        WITH LATTR DO                                                         PAS
        BEGIN TYPTR := GATTR.TYPTR ; KIND := LVAL ; CTERM := 0 END ;          PAS
      END ELSE LATTR := GATTR ;                                               PAS
      REPEAT LMOPCL := CL ; INSYMBOL ; FACTOR ;                               PAS
        IF (LATTR.TYPTR ≠ NIL)∧(GATTR.TYPTR ≠ NIL) THEN                       PAS
        BEGIN CASE LMOPCL OF                                                  PAS
↱ * ↳ 1: BEGIN BT1 := LATTR.TYPTR↑.FORM = NUMERIC ;                           PAS
            BT2 := GATTR.TYPTR↑.FORM = NUMERIC ;                              PAS
            BT3 := LATTR.TYPTR = REALPTR ;                                    PAS
            BT4 := GATTR.TYPTR = REALPTR ;                                    PAS
            IF LOPT ≠ NOOPT THEN                                              PAS
            BEGIN TRANSFER(GATTR,RP) ;                                        PAS
              IF BT4 THEN                                                     PAS
              BEGIN TRANSFER(LATTR,RP) ; PACKANDNORM(RP) ;                    PAS
                LATTR.TYPTR := REALPTR ;                                      PAS
                LOPT := NOOPT ; BT3 := TRUE ;                                 PAS
              END ;                                                           PAS
              WITH LATTR DO                                                   PAS
              BEGIN KIND := LVAL ; CTERM := 0 END ;                           PAS
            END ELSE                                                          PAS
            BEGIN IF (GATTR.KIND = SVAL)∧¬BT3 THEN                            PAS
```

LIST OF CONTROL, ACTIVE, AND/OR INACTIVE CARDS IN PASCSYM 000830

```
            MULOPT(GATTR.VAL,LEXP1,LEXP2,LOPT) ;                                PA
            IF LOPT = NOOPT THEN TRANSFER(GATTR,RP) ;                           PA
          END ;                                                                 PA
          IF LOPT = NOOPT THEN BEGIN RP1 := RP ; RP := RP - 1 END ;             PA
          IF BT3^BT4 THEN MULTCODE(RP1) ELSE                                    PA
          IF BT1^BT2 THEN                                                       PA
          BEGIN IF LOPT = NOOPT THEN                                            PA
            BEGIN GEN15(42B,RP,RP,RP1) ; GEN15(13B,0,0,0) ;                     PA
              GEN15(36B,RP,RP,0)                                                PA
            END ELSE                                                            PA
            BEGIN IF LEXP1 ≠ 0 THEN GEN15(20B,RP,0,LEXP1) ;                     PA
              IF LOPT ≠ PUREP THEN                                              PA
              BEGIN GEN15(10B,0,RP,0) ; GEN15(20B,RP,0,LEXP2) ;                 PA
                IF LOPT = POSP THEN GEN15(36B,RP,RP,0)                          PA
                               ELSE GEN15(37B,RP,RP,0) ;                        PA
              END                                                               PA
            END ;                                                               PA
            LATTR.TYPTR := INTPTR ;                                             PA
          END ELSE                                                              PA
          IF (BT1^BT4)∨(BT2^BT3) THEN                                           PA
          BEGIN IF BT1 THEN PACKANDNORM(RP) ELSE PACKANDNORM(RP1) ;             PA
            MULTCODE(RP1) ; LATTR.TYPTR := REALPTR                              PA
          END ELSE ERROR(50)                                                    PA
        END ;                                                                   PA
↱ / ↓ 2: BEGIN TRANSFER(GATTR,RP) ;                                             PA
          IF GATTR.TYPTR↑.FORM = NUMERIC THEN                                   PA
          BEGIN PACKANDNORM(RP) ; GATTR.TYPTR := REALPTR END ;                  PA
          RP := RP - 1 ;                                                        PA
          IF LATTR.TYPTR↑.FORM = NUMERIC THEN                                   PA
          BEGIN PACKANDNORM(RP) ; LATTR.TYPTR := REALPTR END ;                  PA
          IF (GATTR.TYPTR = REALPTR)^(LATTR.TYPTR = REALPTR) THEN               PA
          BEGIN CHECKDIV(RP+1) ;                                                PA
            IF ROUNDING THEN GEN15(45B,RP,RP,RP+1)                              PA
                        ELSE GEN15(44B,RP,RP,RP+1)                              PA
          END ELSE ERROR(50) ;                                                  PA
        END ;                                                                   PA
↱ ^ ↓ 3: BEGIN TRANSFER(GATTR,RP) ; RP := RP - 1 ;                              PA
          IF (LATTR.TYPTR = GATTR.TYPTR)^((LATTR.TYPTR↑.FORM =                  PA
          POWER)∨(LATTR.TYPTR = BOOLPTR)) THEN GEN15(11B,RP,RP,RP+1)            PA
          ELSE ERROR(50)                                                        PA
        END ;                                                                   PA
↱DIV↓ 4: BEGIN IF GATTR.KIND = SVAL THEN                                        PA
            MULOPT(GATTR.VAL,LEXP1,LEXP2,LOPT) ;                                PA
          IF (LATTR.TYPTR↑.FORM = NUMERIC)^(GATTR.TYPTR↑.FORM =                 PA
          NUMERIC) THEN                                                         PA
          BEGIN IF LOPT = PUREP THEN                                            PA
            BEGIN IF LEXP1 ≠ 0 THEN GEN15(21B,RP,0,LEXP1) ;                     PA
              IF LATTR.TYPTR↑.MIN < 0 THEN                                      PA
              BEGIN GEN15(13B,0,0,0) ; GEN15(36B,RP,RP,0) END                   PA
            END ELSE                                                            PA
            BEGIN TRANSFER(GATTR,RP) ; CHECKDIV(RP) ;                           PA
              PACKANDNORM(RP) ; RP := RP - 1 ;                                  PA
```

LIST OF CONTROL, ACTIVE, AND/OR INACTIVE CARDS IN PASCSYM

```
                GEN15(27B,RP,0,RP) ;                                                    PAS
                GEN15(44B,RP,RP,RP+1) ; GEN15(26B,RP,7,RP) ;                            PAS
                GEN15(22B,RP,7,RP) ; GEN15(13B,0,0,0) ;                                 PAS
               GEN15(36B,RP,RP,0) ;                                                     PAS
              END ;                                                                     PAS
              LATTR.TYPTR := INTPTR                                                     PAS
            END ELSE ERROR(50)                                                          PAS
          END ;                                                                         PAS
 ↱MOD↓ 5: BEGIN IF GATTR.KIND = SVAL THEN                                               PAS
            MULOPT(GATTR.VAL,LEXP1,LEXP2,LOPT) ;                                        PAS
            IF (LATTR.TYPTR↑.FORM = NUMERIC)∧(GATTR.TYPTR↑.FORM =                       PAS
            NUMERIC) THEN                                                               PAS
            BEGIN IF LOPT = PUREP THEN                                                  PAS
              BEGIN GEN15(10B,0,RP,0) ; GEN15(21B,0,0,LEXP1) ;                          PAS
                GEN15(20B,0,0,LEXP1) ; GEN15(37B,RP,RP,0) ;                             PAS
              END ELSE                                                                  PAS
              BEGIN TRANSFER(GATTR,RP) ;                                                PAS
                CHECKDIV(RP) ; GEN15(27B,6,0,RP) ; GEN15(24B,6,0,6) ;                   PAS
                RP1 := RP ; RP := RP - 1 ;                                              PAS
                GEN15(27B,0,0,RP) ; GEN15(44B,6,0,6) ;                                  PAS
                GEN15(26B,6,7,6) ; GEN15(22B,6,7,6) ;                                   PAS
                GEN15(42B,6,RP1,6) ; GEN15(37B,RP,RP,6) ;                               PAS
              END ;                                                                     PAS
              LATTR.TYPTR := INTPTR                                                     PAS
            END ELSE ERROR(50)                                                          PAS
          END ;                                                                         PAS
          END ↱CASE LMOPCL OF↓                                                          PAS
        END ;                                                                           PAS
        LOPT := NOOPT                                                                   PAS
      UNTIL NO ≠ 6 ; ↱MULOP↓                                                            PAS
      GATTR := LATTR                                                                    PAS
    END ↱IF NO = 6↓                                                                     PAS
  END ↱TERM↓ ;                                                                          PAS
                                                                                        PAS
  PROCEDURE SIMPLEEXP ;                                                                 PAS
  VAR LATTR : ATTR ; LADOPCL : SHRTINT ; LFG,BT1,BT2 : BOOLEAN ;                        PAS
  BEGIN LFG := FALSE ;                                                                  PAS
    IF NO = 7 THEN ↱ADDOP↓                                                              PAS
    BEGIN IF CL = 2 THEN LFG := TRUE ELSE IF CL = 3 THEN ERROR(51) ;                    PAS
      INSYMBOL ;                                                                        PAS
    END ;                                                                               PAS
    TERM ;                                                                              PAS
    IF LFG∨(NO = 7) THEN                                                                PAS
    BEGIN WITH LATTR DO                                                                 PAS
      BEGIN TYPTR := GATTR.TYPTR ; KIND := LVAL ; CTERM := 0 ;                          PAS
        IF TYPTR ≠ NIL THEN                                                             PAS
        BEGIN TRANSFER(GATTR,RP) ;                                                      PAS
          IF LFG THEN                                                                   PAS
          BEGIN GEN15(13B,0,0,0) ;                                                      PAS
            IF (TYPTR↑.FORM = NUMERIC)∨(TYPTR = REALPTR) THEN                           PAS
            GEN15(37B,RP,0,RP) ELSE ERROR(50)                                           PAS
          END                                                                           PAS
```

```
      END                                                                    PAS
    END ;                                                                    PAS
    WHILE NO = 7 DO                                                          PAS
    BEGIN LADOPCL := CL ; INSYMBOL ; TERM ;                                  PAS
      IF (LATTR.TYPTR ≠ NIL)∧(GATTR.TYPTR ≠ NIL) THEN                        PAS
      BEGIN BT1 := LATTR.TYPTR↑.FORM = NUMERIC ;                             PAS
        BT2 := GATTR.TYPTR↑.FORM = NUMERIC ;                                 PAS
        IF BT1∧BT2∧(GATTR.KIND = SVAL) THEN                                  PAS
        BEGIN WITH LATTR DO                                                  PAS
          BEGIN TYPTR := INTPTR ;                                            PAS
            CASE LADOPCL OF                                                  PAS
            1: CTERM := CTERM + GATTR.VAL ;                                  PAS
            2: CTERM := CTERM - GATTR.VAL ;                                  PAS
            3: ERROR(50)                                                     PAS
            END                                                              PAS
          END                                                                PAS
        END ELSE                                                             PAS
        BEGIN TRANSFER(GATTR,RP) ; RP1 := RP ; RP := RP - 1 ;                PAS
          IF BT1∧BT2 THEN                                                    PAS
          BEGIN LATTR.TYPTR := INTPTR ;                                      PAS
            CASE LADOPCL OF                                                  PAS
            1: GEN15(36B,RP,RP,RP1) ;                                        PAS
            2: GEN15(37B,RP,RP,RP1) ;                                        PAS
            3: ERROR(50)                                                     PAS
            END                                                              PAS
          END ELSE                                                           PAS
          BEGIN IF BT1 THEN                                                  PAS
            BEGIN PACKANDNORM(RP) ; LATTR.TYPTR := REALPTR                   PAS
            END ELSE                                                         PAS
            IF BT2 THEN                                                      PAS
            BEGIN PACKANDNORM(RP1) ; GATTR.TYPTR := REALPTR                  PAS
            END ;                                                            PAS
            IF (LATTR.TYPTR = REALPTR)∧(GATTR.TYPTR = REALPTR) THEN          PAS
            BEGIN IF LADOPCL ≤ 2 THEN                                        PAS
              BEGIN IF ROUNDING THEN GEN15(33B+LADOPCL,RP,RP,RP1)            PAS
                                ELSE GEN15(27B+LADOPCL,RP,RP,RP1) ;          PAS
                GEN15(24B,RP,0,RP)                                           PAS
              END ELSE ERROR(50)                                             PAS
            END ELSE                                                         PAS
            IF LATTR.TYPTR = GATTR.TYPTR THEN                                PAS
            BEGIN IF (LATTR.TYPTR = BOOLPTR)∧(LADOPCL = 3) THEN              PAS
              GEN15(12B,RP,RP,RP1) ELSE                                      PAS
              IF LATTR.TYPTR↑.FORM = POWER THEN                              PAS
              CASE LADOPCL OF                                                PAS
              1: GEN15(13B,RP,RP,RP1) ; ↗NON-STANDARD PASCAL↘                PAS
              2: GEN15(15B,RP,RP,RP1) ;                                      PAS
              3: GEN15(12B,RP,RP,RP1) ;                                      PAS
              END ELSE ERROR(50)                                             PAS
            END ELSE ERROR(50)                                               PAS
          END                                                                PAS
        END                                                                  PAS
      END                                                                    PAS
```

LIST OF CONTROL, ACTIVE, AND/OR INACTIVE CARDS IN PASCSYM 000033

```
      END ↑WHILE NO = 7↓ ;                                                    PAS
      GATTR := LATTR                                                          PAS
    END ↑IF LFG∨(NO = 7)↓                                                     PAS
  END ↑SIMPLEEXP↓ ;                                                           PAS
                                                                              PAS
  PROCEDURE EXPRESSION ;                                                      PAS
  VAR LATTR : ATTR ; LRELOPCL, LCST : SHRTINT ;                               PAS
    LOF,BT1,BT2 : BOOLEAN ; AT : ADDRESS ;                                    PAS
  BEGIN SIMPLEEXP ;                                                           PAS
    IF NO = 8 THEN ↑RELOP↓                                                    PAS
    BEGIN LOF := TRUE ;                                                       PAS
      WITH GATTR DO                                                           PAS
      IF (KIND = SVAL)∧((CL = 7)∨(VAL = 0)∧(TYPTR ≠ BOOLPTR)∧                 PAS
      ¬(CL IN [1,3])) THEN                                                    PAS
      BEGIN LOF := FALSE ; LCST := VAL END ELSE                               PAS
      BEGIN IF TYPTR ≠ NIL THEN                                               PAS
        IF TYPTR↑.FORM IN [ARRAYS,RECORDS] THEN                               PAS
        BEGIN IF ACCESS = DRCT THEN                                           PAS
          BEGIN RP := RP + 1 ; IF RP = 6 THEN ERROR(33) END ;                 PAS
          DPLMT := DPLMT - 1 ; LOADADR(GATTR,RP) ;                            PAS
        END ELSE TRANSFER(GATTR,RP)                                           PAS
      END ;                                                                   PAS
      WITH LATTR DO                                                           PAS
      BEGIN TYPTR := GATTR.TYPTR ; KIND := LCOND END ;                        PAS
      LRELOPCL := CL ; INSYMBOL ; SIMPLEEXP ;                                 PAS
      IF (LATTR.TYPTR ≠ NIL)∧(GATTR.TYPTR ≠ NIL) THEN                         PAS
      BEGIN WITH GATTR DO                                                     PAS
        IF (KIND = SVAL)∧(VAL = 0)∧(TYPTR ≠ BOOLPTR)∧                         PAS
        ¬(LRELOPCL IN [2,4]) ∧ LOF THEN LOF := FALSE ELSE                     PAS
        BEGIN IF TYPTR↑.FORM IN [RECORDS,ARRAYS] THEN                         PAS
          BEGIN DPLMT := DPLMT - 1 ; LOADADR(GATTR,6) ;                       PAS
            IF ACCESS = DRCT THEN RP := RP + 1 ;                              PAS
          END ELSE TRANSFER(GATTR,RP)                                         PAS
        END ;                                                                 PAS
        IF LOF THEN                                                           PAS
        BEGIN RP1 := RP ; RP := RP - 1 END ;                                  PAS
        IF LRELOPCL = 7 THEN ↑IN↓                                             PAS
        BEGIN IF (GATTR.TYPTR↑.FORM = POWER)∧(GATTR.TYPTR↑.ELSET =            PAS
          LATTR.TYPTR)∨(GATTR.TYPTR = PNUMPTR)∧(LATTR.TYPTR↑.FORM =            PAS
          NUMERIC) THEN                                                       PAS
          BEGIN IF LOF THEN                                                   PAS
            BEGIN GEN15(63B,7,RP,0) ; GEN15(23B,RP,7,RP1) ;                   PAS
              GEN15(20B,RP,0,59) ;                                            PAS
            END ELSE GEN15(20B,RP,0,59-LCST) ;                                PAS
            WITH LATTR DO                                                     PAS
            BEGIN JMP := 2 ; ARITH := TRUE END ;                              PAS
          END ELSE ERROR(50)                                                  PAS
        END ↑IF LRELOPCL = 7↓ ELSE                                            PAS
        BEGIN IF (LATTR.TYPTR = BOOLPTR)∧(GATTR.TYPTR = BOOLPTR) THEN         PAS
          BEGIN CASE LRELOPCL OF                                              PAS
           1: BEGIN GEN15(15B,RP,RP1,RP) ; LATTR.JMP := 0 END ;               PAS
           2: BEGIN GEN15(15B,RP,RP,RP1) ; LATTR.JMP := 1 END ;               PAS
```

LIST OF CONTROL, ACTIVE, AND/OR INACTIVE CARDS IN PASCSYM 000034

```
          3: BEGIN GEN15(15B,RP,RP1,RP) ; LATTR.JMP := 1 END ;                    PA
          4: BEGIN GEN15(15B,RP,RP,RP1) ; LATTR.JMP := 0 END ;                    PA
          5: BEGIN GEN15(13B,RP,RP,RP1) ; LATTR.JMP := 0 END ;                    PA
          6: BEGIN GEN15(13B,RP,RP,RP1) ; LATTR.JMP := 1 END                      PA
           END ↑CASE↓ ;                                                           PA
           LATTR.ARITH := FALSE                                                   PA
         END ↑IF↓ ELSE                                                            PA
         BEGIN BT1 := LATTR.TYPTR↑.FORM = NUMERIC ;                               PA
           BT2 := GATTR.TYPTR↑.FORM = NUMERIC ;                                   PA
           IF BT1∧(GATTR.TYPTR = REALPTR) THEN                                    PA
           BEGIN IF LOF THEN PACKANDNORM(RP) ;                                    PA
             LATTR.TYPTR := REALPTR                                               PA
           END ;                                                                  PA
           IF BT2∧(LATTR.TYPTR = REALPTR) THEN                                    PA
           BEGIN IF LOF THEN PACKANDNORM(RP1) ;                                   PA
             GATTR.TYPTR := REALPTR                                               PA
           END ;                                                                  PA
           IF BT1∧BT2∨(LATTR.TYPTR = GATTR.TYPTR)∧(LATTR.TYPTR↑.FORM               PA
           ≤ POINTER)∨(LATTR.TYPTR↑.FORM = POINTER)∧(GATTR.TYPTR =                 PA
           NILPTR)∨(GATTR.TYPTR↑.FORM = POINTER)∧(LATTR.TYPTR =                    PA
           NILPTR) THEN                                                           PA
           BEGIN IF LATTR.TYPTR = REALPTR THEN IT := 31B ELSE IT := 37B ;         PA
              CASE LRELOPCL OF                                                    PA
          1: BEGIN IF LOF THEN GEN15(IT,RP,RP,RP1) ; LATTR.JMP := 2               PA
             END ;                                                                PA
          2: BEGIN IF LOF THEN GEN15(IT,RP,RP1,RP) ; LATTR.JMP := 3               PA
             END ;                                                                PA
          3: BEGIN IF LOF THEN GEN15(IT,RP,RP,RP1) ; LATTR.JMP := 3               PA
             END ;                                                                PA
          4: BEGIN IF LOF THEN GEN15(IT,RP,RP1,RP) ; LATTR.JMP := 2               PA
             END ;                                                                PA
          5: BEGIN IF LOF THEN GEN15(37B,0,RP,RP1)                                PA
                          ELSE BXIXJ(0,RP) ;                                      PA
               LATTR.JMP := 0                                                     PA
             END ;                                                                PA
          6: BEGIN IF LOF THEN GEN15(37B,0,RP,RP1)                                PA
                          ELSE BXIXJ(0,RP) ;                                      PA
               LATTR.JMP := 1                                                     PA
             END ;                                                                PA
             END ↑CASE↓ ;                                                       PA
             LATTR.ARITH := TRUE                                                PA
           END ↑IF BT1∧BT2∨(...↓ ELSE                                           PA
           IF (LATTR.TYPTR↑.FORM = POWER)∧(GATTR.TYPTR↑.FORM =                   PA
           POWER) THEN                                                          PA
           BEGIN IF (LATTR.TYPTR = GATTR.TYPTR)∨(LATTR.TYPTR = LAMPTR)           PA
             ∨(GATTR.TYPTR = LAMPTR) THEN                                       PA
             BEGIN CASE LRELOPCL OF                                             PA
           1, 4: ERROR(88) ;                                                    PA
             2: BEGIN IF LOF THEN GEN15(15B,0,RP,RP1) ELSE ERROR(89) ;          PA
                  LATTR.JMP := 1                                                PA
                END ;                                                           PA
             3: BEGIN IF LOF THEN GEN15(15B,0,RP1,RP) ELSE ERROR(89) ;          PA
```

000035

LIST OF CONTROL, ACTIVE, AND/OR INACTIVE CARDS IN PASCSYM

```
                    LATTR.JMP := 1                                                  PAS
                  END ;                                                             PAS
            5, 6: BEGIN IF LOF THEN GEN15(13B,0,RP,RP1)                             PAS
                                ELSE BXIXJ(0,RP) ;                                  PAS
                    LATTR.JMP := LRELOPCL - 5                                       PAS
                  END ;                                                             PAS
                END ↱CASE↓ ;                                                        PAS
                LATTR.ARITH := TRUE                                                 PAS
              END ELSE ERROR(50)                                                    PAS
            END ↱IF (LATTR...↓ ELSE                                                 PAS
            IF (LATTR.TYPTR = GATTR.TYPTR)∧((LATTR.TYPTR↑.FORM =                    PAS
            RECORDS)∨(LATTR.TYPTR↑.FORM = ARRAYS)) THEN                             PAS
            BEGIN IF LRELOPCL ≤ 4 THEN ERROR(94) ELSE                               PAS
              BEGIN IF RP ≥ 4 THEN ERROR(33) ;                                      PAS
                GEN30(61B,7,0,GATTR.TYPTR↑.SIZE) ; NOOP ; AT := IC ;                PAS
                GEN15(53B,RP1,6,7) ; GEN15(53B,RP1+1,RP,7) ;                        PAS
                GEN15(37B,0,RP1+1,RP1) ; GEN30(03B,1,0,IC+2) ;                      PAS
                GEN30(61B,7,7,-1) ; GEN30(05B,7,0,AT) ; NOOP ;                      PAS
                WITH LATTR DO                                                       PAS
                BEGIN KIND := LCOND ; ARITH := TRUE ;                               PAS
                  JMP := LRELOPCL - 5                                               PAS
                END ;                                                               PAS
              END                                                                   PAS
            END ELSE ERROR(50)                                                      PAS
          END ↱IF ≠ BOOLPTRS↓ ;                                                     PAS
        END ↱IF LRELOPCL ≤ 6↓                                                       PAS
      END ↱IF TYPTRS ≠ NIL↓ ;                                                       PAS
      LATTR.TYPTR := BOOLPTR ;                                                      PAS
      GATTR := LATTR                                                                PAS
    END ↱IF NO = 8↓                                                                 PAS
  END ↱EXPRESSION↓ ;                                                                PAS
                                                                                    PAS
↱ ------------------------------------------------------------------ ↓             PAS
                                                                                    PAS
  PROCEDURE PASSPARAMS ;                                                            PAS
  ↱PERFORMS CODE GENERATION FOR PASSING PARAMETERS TO CALLED                        PAS
   PROCEDURE AND JUMPING TO ITS ENTRY↓                                              PAS
  VAR LPL,LPC,LDSP,IT2 : SHRTINT ; LPA,AT : ADDRESS; LPK,BT : BOOLEAN;              PAS
      LFP : CTP ;                                                                   PAS
  BEGIN WITH CTPTR↑ DO                                                              PAS
    BEGIN LPA := PROCADDR ; LPL := PROCLEVEL ;                                      PAS
      LFP := FORMALS ; LPK := PROCKIND = ACTUAL                                     PAS
    END ;                                                                           PAS
    LDSP := 3 ; LPC := 0 ;                                                          PAS
    IF NO = 9 THEN ↱(↓                                                              PAS
    BEGIN REPEAT                                                                    PAS
      BEGIN IF LFP = NIL THEN                                                       PAS
        BEGIN IF LPK THEN                                                           PAS
          BEGIN ERROR(72) ; SKIP(49) ; GOTO 1 END ;                                 PAS
          BT := FALSE                                                               PAS
        END ELSE                                                                    PAS
        BT := LFP↑.KLASS = PROC ;                                                   PAS
```

LIST OF CONTROL, ACTIVE, AND/OR INACTIVE CARDS IN PASCSYM 000036

```
        INSYMBOL ;                                                              PAS
        IF BT THEN ↱PROC/FCT TO BE PASSED↓                                      PAS
        BEGIN IF NO ≠ 1 THEN ↱ID↓ ERROR(11) ELSE                                PAS
          BEGIN SEARCH ;                                                        PAS
            IF CTPTR = NIL THEN                                                 PAS
            BEGIN ERROR(31) ; CTPTR := UNDECPTR END ;                           PAS
            IF CTPTR↑.KLASS ≠ PROC THEN ERROR(60) ELSE                          PAS
            BEGIN WITH CTPTR↑ DO                                                PAS
              BEGIN PT := LFP↑.PROCTYPE ;                                       PAS
                IF PROCTYPE ≠ CTPTR THEN ↱PARAM. IS FCT↓                        PAS
                BEGIN IF (PT = LFP)∨(PROCTYPE ≠ PT)∧((PROCTYPE↑.FORM            PAS
                  ≠ NUMERIC)∨(PT↑.FORM ≠ NUMERIC)) THEN ERROR(60)               PAS
                END ELSE ↱PARAM. IS PROC↓                                       PAS
                IF PT ≠ LFP THEN ERROR(60) ;                                    PAS
                IF CTPTR ≤ PREDEFP THEN                                         PAS
                BEGIN ERROR(83) ; SKIP(49) ; GOTO 1 END ELSE                    PAS
                IF CTPTR ≤ EXTPTR THEN  ↱ SET BIT TO INFORM MONITOR ↓           PAS
                WITH PARMLIST DO EXTFLAGS := EXTFLAGS∨[CTPTR↑.SEGSIZE] ;        PAS
                IF PROCTYPE ≠ NIL THEN                                          PAS
                IF PROCKIND = ACTUAL THEN                                       PAS
                BEGIN PT := FORMALS ; IT1 := 0 ; IT2 := 0 ;                     PAS
                  WHILE PT ≠ NIL DO                                             PAS
                  BEGIN IF (PT↑.KLASS = VARS)∧(PT↑.VKIND = FORMAL)              PAS
                    THEN APPEND(IT1,1,1) ELSE APPEND(IT1,1,0) ;                 PAS
                    IT2 := IT2 + 1 ; PT := PT↑.NXTEL                            PAS
                  END ;                                                         PAS
                  APPEND(IT1,17-IT2,0) ; IF IT2 > 17 THEN ERROR(103) ;          PAS
                  GEN30(71B,6,0,PROCADDR) ;                                     PAS
                  IF IT1 ≠ 0 THEN                                               PAS
                  BEGIN GEN30(71B,0,0,IT1) ; GEN15(20B,0,0,43) ;                PAS
                    GEN15(12B,6,6,0) ;                                          PAS
                  END ;                                                         PAS
                  IF PROCLEVEL ≠ 0 THEN                                         PAS
                  BEGIN IF PROCLEVEL = LEVEL THEN GEN15(76B,1,5,0)              PAS
                    ELSE LOADBASE(1,PROCLEVEL) ;                                PAS
                    GEN15(20B,1,0,18) ; GEN15(12B,6,6,1) ;                      PAS
                  END                                                           PAS
                END ELSE                                                        PAS
                BEGIN IF PROCLEVEL = LEVEL THEN                                 PAS
                  GEN30(51B,1,5,PROCADDR) ELSE                                  PAS
                  BEGIN LOADBASE(1,PROCLEVEL) ; GEN30(52B,1,1,PROCADDR)         PAS
                  END ;                                                         PAS
                  GEN15(10B,6,1,0)                                              PAS
                END ;                                                           PAS
              END ↱WITH CTPTR↑↓                                                 PAS
            END ↱IF CTPTR↑.KLASS = PROC↓ ;                                      PAS
            INSYMBOL                                                            PAS
          END ↱NO = 1↓                                                          PAS
        END ELSE ↱EXPRESSION TO BE PASSED↓                                      PAS
        BEGIN                                                                   PAS
        B6DPL := LDSP ; EXPRESSION ;                                            PAS
        IF GATTR.TYPTR ≠ NIL THEN                                               PAS
```

LIST OF CONTROL, ACTIVE, AND/OR INACTIVE CARDS IN PASCSYM 000037

```
      BEGIN IF LPK THEN                                                    PAS
        BEGIN IF (LFP↑.KLASS = KONST)∨(LFP↑.VKIND = ACTUAL) THEN           PAS
          BEGIN IF GATTR.KIND = VARBL THEN                                 PAS
            BEGIN TRANSFER(GATTR,RP) ; GEN15(10B,6,RP,0) END               PAS
            ELSE TRANSFER(GATTR,6) ;                                       PAS
            IF LFP↑.KLASS = KONST THEN PT := LFP↑.CONTYPE                  PAS
                                  ELSE PT := LFP↑.VTYPE ;                  PAS
            IF (GATTR.TYPTR↑.FORM = NUMERIC)∧(PT = REALPTR) THEN           PAS
            BEGIN PACKANDNORM(6) ; GATTR.TYPTR := REALPTR END ;            PAS
          END ELSE                                                         PAS
          BEGIN IF GATTR.KIND = VARBL THEN LOADADR(GATTR,6)                PAS
                                      ELSE ERROR(86) ;                     PAS
            PT := LFP↑.VTYPE                                               PAS
          END ;                                                            PAS
          IF PT ≠ NIL THEN                                                 PAS
          IF GATTR.TYPTR ≠ PT THEN                                         PAS
          IF (GATTR.TYPTR↑.FORM ≠ NUMERIC)∨(PT↑.FORM ≠ NUMERIC) THEN       PAS
          IF (GATTR.TYPTR ≠ NILPTR)∨(PT↑.FORM ≠ POINTER) THEN              PAS
          IF (GATTR.TYPTR ≠ LAMPTR)∨(PT↑.FORM ≠ POWER) THEN                PAS
          ERROR(60) ;                                                      PAS
        END ↱IF LPK↳ ELSE                                                  PAS
        BEGIN IF GATTR.KIND ≠ VARBL THEN TRANSFER(GATTR,6) ELSE            PAS
          BEGIN LOADADR(GATTR,1) ;                                         PAS
            IF LPL = LEVEL THEN GEN30(51B,2,5,LPA) ELSE                    PAS
            BEGIN LOADBASE(2,LPL) ; GEN30(52B,2,2,LPA) END ;               PAS
            IF LPC ≠ 0 THEN GEN15(20B,2,0,LPC) ;                           PAS
            IF CP = 1 THEN AT := IC ELSE AT := IC + 1 ;                    PAS
            GEN30(03B,3,2,AT) ; GEN15(53B,1,1,0) ; NOOP ;                  PAS
            GEN15(10B,6,1,0)                                               PAS
          END                                                              PAS
        END ;                                                              PAS
      END ↱IF GATTR.TYPTR ≠ NIL↳ ;                                         PAS
      END ↱EXP↳ ;                                                          PAS
      GEN30(51B,6,6,LDSP) ;                                                PAS
      LDSP := LDSP + 1 ; LPC := LPC + 1 ; RP := 0 ;                        PAS
      IF LPK THEN LFP := LFP↑.NXTEL                                        PAS
    END                                                                    PAS
    UNTIL NO ≠ 15 ; ↱,↳                                                    PAS
    IF NO ≠ 10 THEN ↱)↳ BEGIN ERROR(48) ; SKIP(49) END ELSE INSYMBOL       PAS
  END ↱IF NO = 9↳ ;                                                        PAS
  IF LFP ≠ NIL THEN ERROR(72) ;                                            PAS
  IF LPK THEN                                                              PAS
  BEGIN IF LPL ≠ 0 THEN                                                    PAS
    BEGIN IF LPL = LEVEL THEN GEN15(76B,6,5,0) ELSE                        PAS
      BEGIN LOADBASE(1,LPL) ; GEN15(10B,6,1,0) END                         PAS
    END ;                                                                  PAS
    GEN30(71B,7,0,IC+1) ; GEN30(04B,0,0,LPA) ;                             PAS
  END ELSE                                                                 PAS
  BEGIN IF LPL = LEVEL THEN GEN30(51B,1,5,LPA) ELSE                        PAS
    BEGIN LOADBASE(1,LPL) ; GEN30(52B,1,1,LPA) END ;                       PAS
    GEN15(63B,7,1,0) ; GEN15(20B,1,0,42) ;                                 PAS
    GEN15(73B,6,1,0) ; GEN30(71B,7,0,IC+1) ;                               PAS
```

LIST OF CONTROL, ACTIVE, AND/OR INACTIVE CARDS IN PASCSYM

```
      GEN30(02B,7,0,0) ;                                                    PA:
    END ;                                                                   PA:
    NOOP ;                                                                  PA:
1:  B6DPL := 3                                                              PA:
  END ↱PASSPARAMS↳ ;                                                        PA:
                                                                            PA:
  PROCEDURE ASSIGN ;                                                        PA:
  VAR LATTR : ATTR ; AT : ADDRESS;                                          PA:
  BEGIN VARIABLE ; LATTR := GATTR ;                                         PA:
    IF NO ≠ 20 THEN ↱:=↳                                                    PA:
    BEGIN IF GATTR.TYPTR ≠ NIL THEN ERROR(52) ; SKIP(20) ;                  PA:
      IF NO ≠ 20 THEN                                                       PA:
      BEGIN IF GATTR.TYPTR = NIL THEN ERROR(52) ; GOTO 1 END                PA:
    END ;                                                                   PA:
    INSYMBOL ; EXPRESSION ;                                                 PA:
    IF GATTR.TYPTR = NIL THEN SKIP(49) ELSE                                 PA:
    IF LATTR.TYPTR ≠ NIL THEN                                               PA:
    BEGIN WITH GATTR DO                                                     PA:
      IF TYPTR↑.FORM ≥ ARRAYS THEN                                          PA:
      BEGIN DPLMT := DPLMT - 1 ; LOADADR(GATTR,2) ; RP := 2 END ELSE        PA:
      IF (KIND = LVAL)∧(CTERM ≠ 0) THEN                                     PA:
      WITH LATTR.TYPTR↑ DO                                                  PA:
      IF (FORM = NUMERIC)∧(ABS(MIN) < TWOTO17)∧(ABS(MAX) < TWOTO17) THEN    PA:
      GEN30(72B,6,RP,CTERM) ELSE TRANSFER(GATTR,6) ELSE                     PA:
      IF KIND = VARBL THEN                                                  PA:
      BEGIN TRANSFER(GATTR,RP) ; GEN15(10B,6,RP,0) END ELSE                 PA:
      TRANSFER(GATTR,6) ;                                                   PA:
      IF GATTR.KIND ≠ SVAL THEN RP := RP - 1 ;                              PA:
      IF (LATTR.TYPTR = REALPTR)∧(GATTR.TYPTR↑.FORM = NUMERIC) THEN         PA:
      PACKANDNORM(6) ELSE                                                   PA:
      IF LATTR.TYPTR ≠ GATTR.TYPTR THEN                                     PA:
      IF (LATTR.TYPTR↑.FORM ≠ NUMERIC)∨(GATTR.TYPTR↑.FORM ≠ NUMERIC)        PA:
      THEN                                                                  PA:
      IF (LATTR.TYPTR↑.FORM ≠ POINTER)∨(GATTR.TYPTR ≠ NILPTR) THEN          PA:
      IF (LATTR.TYPTR↑.FORM ≠ POWER)∨(GATTR.TYPTR ≠ LAMPTR) THEN            PA:
      ERROR(53) ;                                                           PA:
      WITH LATTR.TYPTR↑ DO                                                  PA:
      IF FORM ≤ POWER THEN                                                  PA:
      BEGIN IF (FORM = NUMERIC)∧(LATTR.TYPTR ≠ INTPTR) THEN                 PA:
        BEGIN WITH GATTR DO                                                 PA:
          IF KIND = SVAL THEN                                               PA:
          BEGIN IF (MIN > VAL)∨(MAX < VAL) THEN ERROR(101) ;                PA:
          END ELSE                                                          PA:
          IF ASSCHECK THEN CHECKBNDS(6,MIN,MAX,ASSERR)                      PA:
        END ELSE                                                            PA:
        IF (FORM = SYMBOLIC)∧(LATTR.TYPTR ≠ REALPTR)                        PA:
        ∧(LATTR.TYPTR ≠ ALFAPTR)∧(GATTR.KIND ≠ SVAL)∧ASSCHECK THEN          PA:
        CHECKBNDS(6,0,FCONST↑.VALUES,ASSERR) ;                              PA:
        TRANSFER(LATTR,6)                                                   PA:
      END ELSE                                                              PA:
      IF FORM ≥ CLASSS THEN ERROR(53) ELSE                                  PA:
      BEGIN LATTR.DPLMT := LATTR.DPLMT - 1 ; LOADADR(LATTR,1) ;             PA:
```

```
        GEN30(61B,7,0,SIZE) ; NOOP ; AT := IC ;                                    PAS
        GEN15(53B,3,2,7) ; GEN15(10B,6,3,0) ; GEN15(53B,6,1,7) ;                   PAS
        GEN30(61B,7,7,-1) ; GEN30(05B,0,7,AT)                                      PAS
      END                                                                          PAS
    END ↑IF LATTR.TYPTR = NIL↓ ;                                                   PAS
1:END ↑ASSIGN↓ ;                                                                   PAS
                                                                                   PAS
    PROCEDURE VARIAB ;                                                             PAS
    BEGIN IF NO ≠ 1 THEN                                                           PAS
      BEGIN ERROR(49) ; GATTR.TYPTR := NIL END ELSE                                PAS
      BEGIN SEARCH ;                                                               PAS
        IF CTPTR = NIL THEN                                                        PAS
        BEGIN ERROR(31) ; CTPTR := UNDECPTR END ;                                  PAS
        IF CTPTR↑.KLASS ≤ PROC THEN                                                PAS
        BEGIN ERROR(32) ; INSYMBOL ; GATTR.TYPTR := NIL END                        PAS
        ELSE VARIABLE                                                              PAS
      END                                                                          PAS
    END ↑VARIAB↓ ;                                                                 PAS
                                                                                   PAS
  PROCEDURE TITLE ;                                                                PAS
    CONST TXT = 112B ;                                                             PAS
    VAR I, LCA, LCP : SHRTINT ; AWORD : ALFA ;                                     PAS
      A : ARRAY [1..ALFALENG] OF CHAR ;                                            PAS
  BEGIN WHILE CH = ΞΞ DO NEXTCH ;                                                  PAS
    IF CH ≠ ΞΞΞΞ THEN                                                              PAS
    BEGIN INSYMBOL ; ERROR(44) ; SKIP(49) ; GOTO 10 END ;                          PAS
    GEN30(71B,7,0,0) ; LCA := CA ; LCP := CP ;                                     PAS
    GEN30(71B,1,0,IC+1) ; GEN30(04B,0,0,TXT) ;                                     PAS
    NOOP ; CODE[CA] := BUF ; I := 0 ; NEXTCH ;                                     PAS
    WHILE (CH ≠ ΞΞΞΞ)∧¬EOLFLAG DO                                                  PAS
    BEGIN I := I + 1 ; A[I] := CH ;                                                PAS
      IF I = ALFALENG THEN                                                         PAS
      BEGIN PACK(A,1,AWORD) ; I := 0 ;                                             PAS
        CA := CA + 1 ; IC := IC + 1 ;                                              PAS
        IF CA > CODMAX THEN                                                        PAS
        BEGIN ERROR(90) ; CA := 1 END ;                                            PAS
        CODE[CA] := INT(AWORD)                                                     PAS
      END ;                                                                        PAS
      NEXTCH                                                                       PAS
    END ;                                                                          PAS
    IF EOLFLAG THEN                                                                PAS
    BEGIN ERROR(44) ; INSYMBOL ; SKIP(49) ; GOTO 10 END ;                          PAS
    FOR I := I + 1 TO ALFALENG DO A[I] := EOL ;                                    PAS
    PACK(A,1,AWORD) ; BUF := INT(AWORD) ;                                          PAS
    CA := CA + 1 ; IC := IC + 1 ; INS(IC,LCP,LCA) ;                                PAS
    IF CA > CODMAX THEN                                                            PAS
    BEGIN ERROR(90) ; CA := 1 END ;                                                PAS
    NEXTCH ; INSYMBOL ;                                                            PAS
    IF NO ≠ 10 THEN                                                                PAS
    BEGIN ERROR(48) ; SKIP(49) END ELSE INSYMBOL ;                                 PAS
10:END ↑TITLE↓ ;                                                                   PAS
                                                                                   PAS
```

LIST OF CONTROL, ACTIVE, AND/OR INACTIVE CARDS IN PASCSYM 000040

```
      PROCEDURE GETPUT(LPSW : SHRTINT) ;                                        PA
         CONST EOR = 107B, GETB = 113B, PUTB = 102B, REST = 106B ;              PA
      BEGIN                                                                     PA
         IF NO ≠ 9 THEN  ↱ SY ≠ ( ↳                                            PA
         BEGIN ERROR(79) ; SKIP(49) ; GOTO 10 END ;                             PA
         INSYMBOL ; VARIAB ;                                                    PA
         WITH GATTR DO                                                          PA
         IF TYPTR ≠ NIL THEN                                                    PA
         BEGIN                                                                  PA
            IF TYPTR↑.FORM = FILES THEN                                         PA
            BEGIN                                                               PA
               DPLMT := DPLMT + 1 ;                                             PA
               IF ACCESS = DRCT THEN                                            PA
               BEGIN IF BREG IN [0,LEVEL] THEN                                  PA
                  BEGIN IF BREG ≠ 0 THEN BREG := 5 ;                            PA
                     GEN30(61B,7,BREG,DPLMT)                                    PA
                  END ELSE                                                      PA
                  BEGIN LOADBASE(5,BREG) ; GEN30(62B,7,5,DPLMT) END ;           PA
               END ELSE GEN30(62B,7,RP,DPLMT) ;                                 PA
               CASE LPSW OF                                                     PA
                 ↱EOR↳ 1: JUMPTO(EOR) ;                                         PA
                 ↱GET↳ 2: IF TYPTR↑.FELTYPE = CHARPTR THEN JUMPTO(GETC)         PA
                                                      ELSE JUMPTO(GETB) ;       PA
                 ↱PUT↳ 3: IF TYPTR↑.FELTYPE = CHARPTR THEN JUMPTO(PUTC)         PA
                                                      ELSE JUMPTO(PUTB) ;       PA
                 ↱RST↳ 4: JUMPTO(REST)                                          PA
               END ;                                                            PA
            END ELSE                                                            PA
            IF (TYPTR↑.FORM = POINTER)∧(LPSW=4) THEN                            PA
            BEGIN                                                               PA
               TRANSFER(GATTR,RP) ; GEN15(10B,6,RP,0) ;                         PA
               RP := RP - 1 ; ADDRESSVAR(TYPTR↑.DOMAIN,GATTR) ;                 PA
               TRANSFER(GATTR,6)                                                PA
            END ELSE ERROR(44) ;                                                PA
         END ;                                                                  PA
         IF NO ≠ 10 THEN  ↱ SY ≠ ) ↳                                           PA
         BEGIN ERROR(48) ; SKIP(49) ; END ELSE INSYMBOL ;                       PA
10:   END  ↱ GETPUT ↳  ;                                                        PA
                                                                                PA
      PROCEDURE ALLC ;                                                          PA
         VAR BT1 : BOOLEAN ; LATTR : ATTR ;                                     PA
      BEGIN                                                                     PA
         IF NO ≠ 9 THEN  ↱ SY ≠ ( ↳                                            PA
         BEGIN ERROR(79) ; SKIP(49) ; GOTO 10 END ;                             PA
         INSYMBOL;  VARIAB;  PT := NIL;                                         PA
         WITH GATTR DO                                                          PA
         IF TYPTR ≠ NIL THEN                                                    PA
         WITH TYPTR↑ DO                                                         PA
         IF FORM ≠ POINTER THEN ERROR(44) ELSE                                  PA
         BEGIN                                                                  PA
            ADDRESSVAR(DOMAIN,LATTR) ; TRANSFER(LATTR,RP) ;                     PA
            IT1 := DOMAIN↑.VTYPE↑.SIZE ; PT := ELTYPE ;                         PA
```

LIST OF CONTROL, ACTIVE, AND/OR INACTIVE CARDS IN PASCSYM 000041

```
        END ;                                                                   PAS
        IF PT = NIL THEN                                                        PAS
        BEGIN SKIP(49);  GOTO 10  END;                                          PAS
        IF NO = 15 THEN  ↱ SY = , ↓                                             PAS
        BEGIN                                                                   PAS
           PT := PT↑.RECVAR;                                                    PAS
           REPEAT                                                               PAS
              IF PT = NIL THEN                                                  PAS
              BEGIN ERROR(66);  SKIP(49);  GOTO 10  END;                        PAS
              INSYMBOL;                                                         PAS
              IF NO = 1 THEN                                                    PAS
              BEGIN SEARCH;                                                     PAS
                 IF CTPTR = NIL THEN                                            PAS
                 BEGIN ERROR(31);  SKIP(49);  GOTO 10 END;                      PAS
                 IF CTPTR↑.KLASS ≠ KONST THEN                                   PAS
                 BEGIN ERROR(63);  SKIP(49);  GOTO 10 END;                      PAS
                 WITH PT↑ DO                                                    PAS
                 IF (CASETYPE↑.FORM = SYMBOLIC) ∧                               PAS
                     (CTPTR↑.CONTYPE ≠ CASETYPE)  ∨                             PAS
                   (CASETYPE↑.FORM = NUMERIC) ∧                                 PAS
                     (CTPTR↑.CONTYPE ≠ INTPTR) THEN                             PAS
                 BEGIN ERROR(73);  SKIP(49);  GOTO 10 END;                      PAS
                 IVAL := CTPTR↑.VALUES;                                         PAS
              END ELSE                                                          PAS
              IF (NO ≠ 2) ∨ (CL IN [2,3]) THEN                                  PAS
              BEGIN ERROR(63);  SKIP(49);  GOTO 10 END ELSE                     PAS
              WITH PT↑ DO                                                       PAS
              IF (CL = 1) ∧ (CASETYPE↑.FORM ≠ NUMERIC)                          PAS
                 ∨ (CL = 4) ∧ (CASETYPE ≠ CHARPTR) THEN                         PAS
              BEGIN ERROR(73);  SKIP(49);  GOTO 10 END;                         PAS
              CTPTR := PT↑.VARIANTS;  BT1 := FALSE;                             PAS
              WHILE (CTPTR ≠ NIL) ∧ ¬BT1 DO                                     PAS
              WITH CTPTR↑ DO                                                    PAS
                 IF CASEVAL = IVAL THEN BT1 := TRUE                             PAS
                                   ELSE CTPTR := NXTEL;                         PAS
              IF CTPTR = NIL THEN                                               PAS
              BEGIN IT := PT↑.CASESIZE;  PT := NIL END ELSE                     PAS
              BEGIN IT := CTPTR↑.CASESIZE;  PT := CTPTR↑.VARIANTS END;          PAS
              INSYMBOL;                                                         PAS
           UNTIL NO ≠ 15;                                                       PAS
        END ELSE IT := PT↑.SIZE;                                                PAS
        GEN30(72B,6,RP,IT) ; GEN15(54B,6,RP,0) ;                                PAS
        GEN30(70B,0,RP,IT1) ; GEN15(37B,0,0,6) ;                                PAS
        GEN15(10B,6,RP,0) ; GEN30(03B,2,0,IC + (CP+1) DIV 2) ;                  PAS
        GEN15(54B,6,RP,0) ; GEN30(71B,6,0,NILVAL) ;                             PAS
        NOOP ; RP := RP - 1 ; TRANSFER(GATTR,6) ;                               PAS
        IF NO ≠ 10 THEN  ↱ SY ≠ ) ↓                                             PAS
        BEGIN ERROR(48) ; SKIP(49) END ELSE INSYMBOL ;                          PAS
10:   END  ↱ ALLC ↓  ;                                                          PAS
                                                                                PAS
      PROCEDURE PCK ;                                                           PAS
        VAR LPT : CTP ; LLO, LHI : SHRTINT ;                                    PAS
```

LIST OF CONTROL, ACTIVE, AND/OR INACTIVE CARDS IN PASCSYM 000042

```
      BEGIN                                                                  PAS
         IF NO ≠ 9 THEN  ↱ SY ≠ ( ↳                                          PAS
         BEGIN ERROR(79) ; SKIP(49) ; GOTO 10 END ;                          PAS
         INSYMBOL ; VARIAB ;                                                 PAS
         WITH GATTR DO                                                       PAS
         IF TYPTR ≠ NIL THEN                                                 PAS
         BEGIN                                                               PAS
            WITH TYPTR↑ DO                                                   PAS
            IF (FORM ≠ ARRAYS)∨(AELTYPE ≠ CHARPTR) THEN ERROR(44) ELSE       PAS
            BEGIN LPT := INXTYPE ; DPLMT := DPLMT + (ALFALENG-1) - LO ;      PAS
               LLO := LO ; LHI := HI - (ALFALENG - 1) ;                      PAS
            END ;                                                            PAS
            LOADADR(GATTR,1) ; RP := 1 ;                                     PAS
         END ELSE LPT := INTPTR;                                             PAS
         IF NO ≠ 15 THEN                                                     PAS
         BEGIN ERROR(80) ; SKIP(49) ; GOTO 10 END ;                          PAS
         INSYMBOL ; EXPRESSION ;                                             PAS
         WITH GATTR DO                                                       PAS
         IF TYPTR ≠ NIL THEN                                                 PAS
         BEGIN                                                               PAS
            IF (TYPTR↑.FORM ≠ NUMERIC)∨(LPT↑.FORM ≠ NUMERIC) THEN            PAS
            BEGIN IF TYPTR ≠ LPT THEN ERROR(44) END ELSE                     PAS
            BEGIN                                                            PAS
               TRANSFER(GATTR,2) ; RP := 2 ;                                 PAS
               IF INXCHECK THEN CHECKBNDS(2,LLO,LHI,INXERR) ;                PAS
               GEN15(36B,0,1,2) ;                                            PAS
               GEN30(61B,7,0,-ALFALENG) ; GEN15(13B,6,0,0) ;                 PAS
               NOOP ; IT := IC ;                                             PAS
               GEN30(61B,7,7,1) ; GEN15(53B,1,0,7) ;                         PAS
               GEN15(20B,6,0,6) ; GEN15(12B,6,1,6) ;                         PAS
               GEN30(05B,7,0,IT) ; RP := 0                                   PAS
            END ;                                                            PAS
         END ;                                                               PAS
         IF NO ≠ 15 THEN                                                     PAS
         BEGIN ERROR(80) ; SKIP(49) ; GOTO 10 END ;                          PAS
         INSYMBOL ; VARIAB ;                                                 PAS
         IF GATTR.TYPTR ≠ NIL THEN                                           PAS
            IF GATTR.TYPTR ≠ ALFAPTR THEN ERROR(44)                          PAS
                                      ELSE TRANSFER(GATTR,6) ;               PAS
         IF NO ≠ 10 THEN  ↱ SY ≠ ) ↳                                         PAS
         BEGIN ERROR(48) ;SKIP(49) ; END ELSE INSYMBOL ;                     PAS
   10:   END  ↱ PCK ↳  ;                                                     PAS
                                                                             PAS
      PROCEDURE UNPCK ;                                                      PAS
         VAR LPT : CTP ; LLO, LHI : SHRTINT ;                                PAS
      BEGIN                                                                  PAS
         IF NO ≠ 9 THEN  ↱ SY ≠ ( ↳                                          PAS
         BEGIN ERROR(79) ; SKIP(49) ; GOTO 10 END ;                          PAS
         INSYMBOL ; EXPRESSION ;                                             PAS
         IF GATTR.TYPTR ≠ NIL THEN                                           PAS
            IF GATTR.TYPTR ≠ ALFAPTR THEN ERROR(44)                          PAS
                                      ELSE TRANSFER(GATTR,1) ;               PAS
```

LIST OF CONTROL, ACTIVE, AND/OR INACTIVE CARDS IN PASCSYM 000043

```
        IF NO ≠ 15 THEN                                                          PAS
        BEGIN ERROR(80) ; SKIP(49) ; GOTO 10 END ;                               PAS
        INSYMBOL ; VARIAB ;                                                      PAS
        WITH GATTR DO                                                            PAS
        IF TYPTR ≠ NIL THEN                                                      PAS
        BEGIN                                                                    PAS
           WITH TYPTR↑ DO                                                        PAS
           IF (FORM ≠ ARRAYS)∨(AELTYPE ≠ CHARPTR) THEN ERROR(44) ELSE            PAS
           BEGIN LPT := INXTYPE ; DPLMT := DPLMT + (ALFALENG-1) - LO ;           PAS
           LLO := LO ; LHI := HI - (ALFALENG - 1) ;                              PAS
        END ;                                                                    PAS
           LOADADR(GATTR,2) ; RP := 2 ;                                          PAS
        END ELSE LPT := INTPTR;                                                  PAS
        IF NO ≠ 15 THEN                                                          PAS
        BEGIN ERROR(80) ; SKIP(49) ; GOTO 10 END ;                               PAS
        INSYMBOL ; EXPRESSION ;                                                  PAS
        WITH GATTR DO                                                            PAS
        IF TYPTR ≠ NIL THEN                                                      PAS
        BEGIN                                                                    PAS
           IF (TYPTR↑.FORM ≠ NUMERIC)∨(LPT↑.FORM ≠ NUMERIC) THEN                 PAS
           BEGIN IF TYPTR ≠ LPT THEN ERROR(44) END ELSE                          PAS
           BEGIN                                                                 PAS
              TRANSFER(GATTR,3) ; RP := 3 ;                                      PAS
              IF INXCHECK THEN CHECKBNDS(3,LLO,LHI,INXERR) ;                     PAS
              GEN15(36B,2,2,3) ;                                                 PAS
              GEN30(61B,7,0,-ALFALENG) ; GEN15(43B,0,0,54) ;                     PAS
              NOOP ; IT := IC ;                                                  PAS
              GEN30(61B,7,7,1) ; GEN15(20B,1,0,6) ;                              PAS
              GEN15(15B,6,1,0) ; GEN15(53B,6,2,7) ;                              PAS
              GEN30(05B,7,0,IT) ;                                                PAS
           END                                                                   PAS
        END ;                                                                    PAS
        IF NO ≠ 10 THEN  ↱ SY ≠ ) ↓                                              PAS
        BEGIN ERROR(48) ;SKIP(49) ; END ELSE INSYMBOL ;                          PAS
10:  END  ↱ UNPCK ↓  ;                                                           PAS
                                                                                 PAS
     PROCEDURE INSAPP(LPSW : SHRTINT) ;                                          PAS
     VAR LVALUE : SHRTINT ; LOF : BOOLEAN ;                                      PAS
     BEGIN                                                                       PAS
        IF NO ≠ 9 THEN  ↱ SY ≠ ( ↓                                               PAS
        BEGIN ERROR(79) ; SKIP(49) ; GOTO 10 END ;                               PAS
        INSYMBOL ;                                                               PAS
        IF LPSW = 8 THEN EXPRESSION ELSE VARIAB ;                                PAS
        IF GATTR.TYPTR ≠ NIL THEN                                                PAS
           IF GATTR.TYPTR↑.FORM ≠ NUMERIC THEN ERROR(44)                         PAS
                                          ELSE TRANSFER(GATTR,1) ;               PAS
        IF NO ≠ 15 THEN                                                          PAS
        BEGIN ERROR(80) ; SKIP(49) ; GOTO 10 END ;                               PAS
        INSYMBOL ; EXPRESSION ;                                                  PAS
        LOF := FALSE ;                                                           PAS
        WITH GATTR DO                                                            PAS
        IF TYPTR ≠ NIL THEN                                                      PAS
```

```
            IF TYPTR↑.FORM ≠ NUMERIC THEN ERROR(44) ELSE                    PA
            IF KIND = SVAL THEN                                             PA
            BEGIN LOF := TRUE ; LVALUE := VAL ; RP := 2 END                 PA
            ELSE TRANSFER(GATTR,2) ;                                        PA
         IF NO ≠ 15 THEN                                                    PA
         BEGIN ERROR(80) ; SKIP(49) ; GOTO 10 END ;                         PA
         INSYMBOL ;                                                         PA
         IF LPSW = 8 THEN VARIAB ELSE EXPRESSION ;                          PA
         IF GATTR.TYPTR ≠ NIL THEN                                          PA
            IF GATTR.TYPTR↑.FORM ≠ NUMERIC THEN ERROR(44)                   PA
                                    ELSE TRANSFER(GATTR,3) ;                PA
         IF LOF THEN                                                        PA
         BEGIN IF LVALUE ≠ 0 THEN GEN15(20B,1,0,LVALUE) END ELSE            PA
         BEGIN GEN15(63B,7,2,0) ; GEN15(22B,1,7,1) END ;                    PA
         GEN15(12B,6,1,3) ;                                                 PA
         IF LPSW = 8 THEN GEN15(54B,6,3,0) ELSE GEN15(54B,6,1,0) ;          PA
         IF NO ≠ 10 THEN  ↱ SY ≠ ) ↓                                        PA
         BEGIN ERROR(48) ; SKIP(49) ; END ELSE INSYMBOL ;                   PA
10:   END  ↱ INSAPP ↓  ;                                                    PA
                                                                            PA
      PROCEDURE READIR;                                                     PA
         CONST RD = 110B ;                                                  PA
      BEGIN IF NO ≠ 9 THEN ↱ ( ↓                                            PA
         BEGIN ERROR(79);  SKIP(49);  GOTO 10  END;                         PA
         REPEAT                                                             PA
            INSYMBOL;  VARIAB;                                              PA
            IF GATTR.TYPTR ≠ NIL THEN                                       PA
               IF GATTR.TYPTR = CHARPTR THEN                                PA
               BEGIN ↱ INLINE CODE ↓                                        PA
                  IF GATTR.ACCESS ≠ DRCT THEN                               PA
                  BEGIN GEN15(10B,6,1,0) ; IT := LC + TCT ;                 PA
                     IF TCT = TMAX THEN TMAX := TMAX + 1 ;                  PA
                     GEN30(51B,6,5,IT) ;                                    PA
                  END ;                                                     PA
                  GEN30(61B,7,0,INPT+1);  JUMPTO(GETC) ;                    PA
                  GEN30(51B,1,0,INPT);  GEN15(53B,1,1,0);                   PA
                  GEN15(10B,6,1,0);                                         PA
                  IF GATTR.ACCESS ≠ DRCT THEN GEN30(51B,1,5,IT);            PA
                  TRANSFER(GATTR,6);                                        PA
               END ELSE                                                     PA
               BEGIN LOADADR(GATTR,1) ;                                     PA
                  IF GATTR.TYPTR↑.FORM = NUMERIC THEN                       PA
                     GEN15(66B,7,0,0)  ELSE                                 PA
                  IF GATTR.TYPTR = REALPTR THEN                             PA
                     GEN30(61B,7,0,1)  ELSE ERROR(44) ;                     PA
                  JUMPTO(RD) ;                                              PA
               END;  RP := 0;                                               PA
         UNTIL NO ≠ 15 ↱ , ↓ ;                                              PA
         IF NO ≠ 10 THEN ↱ ) ↓                                              PA
         BEGIN ERROR(48);  SKIP(49)  END ELSE INSYMBOL ;                    PA
 10:  END ↱READIR↓ ;                                                        PA
                                                                            PA
```

LIST OF CONTROL, ACTIVE, AND/OR INACTIVE CARDS IN PASCSYM 000045

```
      PROCEDURE WRITEIR ;                                                    PAS
         CONST WRTI=114B,WRTE=115B,WRTF=116B,WRTA=117B,WRTC=120B,WRTB=121B,  PAS
               WRTO=111B;                                                    PAS
         VAR LPTR : CTP ; LX2 : BOOLEAN ;                                    PAS
         PROCEDURE SETX2ANDJMP(FDEFVAL : INTEGER; FADR : ADDRESS) ;          PAS
         BEGIN IF LX2 THEN GEN30(71B,2,0,FDEFVAL) ;                          PAS
            JUMPTO(FADR)                                                     PAS
         END ;                                                               PAS
         PROCEDURE LOADEXP ;                                                 PAS
         BEGIN INSYMBOL ; EXPRESSION ;                                       PAS
            IF GATTR.TYPTR ≠ NIL THEN                                        PAS
               IF GATTR.TYPTR↑.FORM ≠ NUMERIC THEN ERROR(62) ;               PAS
            TRANSFER(GATTR,RP)                                               PAS
         END ;                                                               PAS
      BEGIN IF NO ≠ 9 THEN ↱(↓                                               PAS
         BEGIN ERROR(79) ; SKIP(49) ; GOTO 10 END ;                          PAS
         REPEAT INSYMBOL ; EXPRESSION ; TRANSFER(GATTR,RP) ;                 PAS
            LPTR := GATTR.TYPTR ;                                            PAS
            IF NO = 19 THEN ↱:↓                                              PAS
            BEGIN LOADEXP ; LX2 := FALSE ;                                   PAS
            END ELSE LX2 := TRUE ; ↱X2 NOT YET SET↓                          PAS
            IF LPTR ≠ NIL THEN                                               PAS
            IF (NO=1)∧(AVAL=ΞOCTΞ)∧(LPTR↑.FORM≤POWER) THEN ↱O-FORMAT↓         PAS
            BEGIN SETX2ANDJMP(20,WRTO) ; INSYMBOL END ELSE                   PAS
            IF LPTR↑.FORM = NUMERIC THEN SETX2ANDJMP(10,WRTI) ELSE           PAS
            IF LPTR = REALPTR THEN                                           PAS
            BEGIN IF NO = 19 THEN ↱:↓                                        PAS
               BEGIN LOADEXP ;                                               PAS
                  SETX2ANDJMP(20,WRTF)                                       PAS
               END ELSE SETX2ANDJMP(20,WRTE) ;                               PAS
            END ELSE                                                         PAS
            IF LPTR = ALFAPTR THEN SETX2ANDJMP(10,WRTA) ELSE                 PAS
            IF LPTR = BOOLPTR THEN SETX2ANDJMP(10,WRTB) ELSE                 PAS
            IF LPTR = CHARPTR THEN                                           PAS
            BEGIN IF LX2 THEN ↱CALL PUTC ONLY↓                               PAS
               BEGIN BXIXJ(6,1) ;                                            PAS
                  GEN30(51B,1,0,OUTPT) ; GEN15(53B,6,1,0) ;                  PAS
                  GEN30(61B,7,0,OUTPT+1) ; JUMPTO(PUTC)                      PAS
               END ELSE JUMPTO(WRTC)                                         PAS
            END ELSE ERROR(44) ;   RP := 0 ;                                 PAS
         UNTIL NO ≠ 15 ; ↱,↓                                                 PAS
         IF NO ≠ 10 THEN ↱)↓                                                 PAS
         BEGIN ERROR(48) ; SKIP(49) END ELSE INSYMBOL ;                      PAS
10:   END ↱WRITEIR↓ ;                                                       PAS
                                                                            PAS
      PROCEDURE IFSTAT ;                                                    PAS
      VAR LCA1, LCA2, LCP1, LCP2: SHRTINT ;                                 PAS
      BEGIN                                                                 PAS
         INSYMBOL ; EXPRESSION ;                                            PAS
         IF GATTR.TYPTR ≠ NIL THEN GENJP(0) ;                               PAS
         LCA1 := CA ; LCP1 := CP ;                                          PAS
         IF NO ≠ 24 THEN  ↱ SY ≠ THEN ↓                                     PAS
```

LIST OF CONTROL, ACTIVE, AND/OR INACTIVE CARDS IN PASCSYM

000046

```
        BEGIN                                                                   PA
          IF GATTR.TYPTR ≠ NIL THEN ERROR(56) ; SKIP(24) ;                      PA
          IF NO ≠ 24 THEN                                                       PA
          BEGIN                                                                 PA
            IF GATTR.TYPTR = NIL THEN ERROR(56) ;                               PA
            IF ERRCL[NO] = ENDSY THEN GOTO 30 ; GOTO 20 ;                       PA
          END                                                                   PA
        END ;                                                                   PA
       INSYMBOL ;                                                               PA
20:     STATEMENT ;                                                             PA
30:     IF NO ≠ 25 THEN  ↱ SY ≠ ELSE ↓                                          PA
        BEGIN NOOP;  INS(IC,LCP1,LCA1)  END ELSE                                PA
        BEGIN                                                                   PA
          GEN30(04B,0,0,0) ;                                                    PA
          LCA2 := CA ; LCP2 := CP ;                                             PA
          NOOP ;                                                                PA
          INS(IC,LCP1,LCA1);                                                    PA
          INSYMBOL ; STATEMENT ;                                                PA
          NOOP;  INS(IC,LCP2,LCA2);                                             PA
        END                                                                     PA
      END  ↱ IFSTAT ↓  ;                                                        PA
                                                                                PA
                                                                                PA
  PROCEDURE CASESTAT ;                                                          PA
  TYPE PTR = ↑LCSLABS ;                                                         PA
  VAR LCSLABS : CLASS   30   OF PACKED RECORD NEXT : PTR ;                      PA
                                              CSLAB : SHRTINT ;                 PA
                                              ADDR : ADDRESS ;                  PA
                                            END ;                               PA
      LCA, LCP, LCA1, LCP1, LCA2, LCP2, LMIN, LMAX : SHRTINT ;                  PA
        LPTR, PT1, PT2 : PTR ; LCTP : CTP ; LERR : BOOLEAN ;                    PA
                                                                                PA
    PROCEDURE ACASE ;                                                           PA
    VAR LCA,LCP : SHRTINT ; LERRFG : BOOLEAN ; PT1,PT2,PT3 : PTR ;              PA
    BEGIN LERRFG := TRUE ; NOOP ;                                               PA
      REPEAT IF NO IN [15,27] THEN ↱, OF↓ INSYMBOL ;                            PA
        IF NO = 1 THEN ↱ID↓                                                     PA
        BEGIN SEARCH ;                                                          PA
          IF CTPTR = NIL THEN                                                   PA
          BEGIN ERROR(31) ; CTPTR := UNDECPTR END ;                             PA
          WITH CTPTR↑ DO                                                        PA
          BEGIN IF KLASS ≠ KONST THEN                                           PA
            BEGIN IF LERRFG THEN ERROR(61) ; GOTO 1 END ;                       PA
            IF CONTYPE ≠ NIL THEN                                               PA
            BEGIN IF CONKIND = FORMAL THEN                                      PA
              BEGIN ERROR(61) ; GOTO 1 END ;                                    PA
              IF LCTP = NIL THEN LCTP := CONTYPE ELSE                           PA
              IF LCTP ≠ CONTYPE THEN ERROR(73) ;                                PA
              IT := VALUES ;                                                    PA
            END                                                                 PA
          END                                                                   PA
        END ↱IF NO = 1↓ ELSE                                                    PA
```

```
        IF NO = 2 THEN ↱CONST↳                                                  PAS
        BEGIN CASE CL OF                                                        PAS
          1: PT := INTPTR ;                                                     PAS
          2: PT := REALPTR ;                                                    PAS
          3: PT := ALFAPTR ;                                                    PAS
          4: PT := CHARPTR ;                                                    PAS
          END ;                                                                 PAS
          IF LCTP = NIL THEN LCTP := PT ELSE                                    PAS
          IF LCTP ≠ PT THEN ERROR(73) ;                                         PAS
          IT := IVAL ;                                                          PAS
        END ↱IF NO = 2↳ ELSE                                                    PAS
        BEGIN IF LERRFG THEN ERROR(61) ; GOTO 1 END ;                           PAS
        IF IT ≥ TWOTO17 THEN ERROR(100) ;                                       PAS
        LERRFG := FALSE ; PT1 := LPTR ; PT2 := NIL ;                            PAS
        IF ¬ERR THEN                                                            PAS
        WHILE PT1 ≠ NIL DO                                                      PAS
        BEGIN IF PT1↑.CSLAB ≥ IT THEN                                           PAS
          BEGIN IF PT1↑.CSLAB = IT THEN ERROR(77) ; GOTO 3 END ;                PAS
          PT2 := PT1 ; PT1 := PT1↑.NEXT ;                                       PAS
        END ;                                                                   PAS
        LMAX := IT ;                                                            PAS
  3:    ALLOC(PT3) ;                                                            PAS
        IF PT3 ≠ NIL THEN                                                       PAS
        BEGIN WITH PT3↑ DO                                                      PAS
          BEGIN NEXT := PT1 ; CSLAB := IT ; ADDR := IC END ;                    PAS
          IF PT2 = NIL THEN                                                     PAS
          BEGIN LPTR := PT3 ; LMIN := IT END ELSE PT2↑.NEXT := PT3 ;            PAS
        END ELSE ERROR(71) ;                                                    PAS
        INSYMBOL ;                                                              PAS
      UNTIL NO ≠ 15 ; ↱,↳                                                      PAS
      IF NO ≠ 19 THEN ↱:↳ ERROR(64) ELSE INSYMBOL ;                             PAS
1:    STATEMENT ; GEN30(04B,0,0,0) ; LCA := CA ; LCP := CP ;                    PAS
      IF NO = 16 THEN ↱;↳                                                       PAS
      BEGIN INSYMBOL ; IF NO ≠ 22 THEN ACASE END ELSE                           PAS
      IF ERRCL[NO] = BEGSY THEN                                                 PAS
      BEGIN ERROR(58) ; GOTO 1 END ELSE                                         PAS
      IF NO ≠ 22 THEN ERROR(68) ;                                               PAS
      INS(IC+LMAX-LMIN+1,LCP,LCA) ;                                             PAS
    END ↱ACASE↳ ;                                                               PAS
                                                                                PAS
  BEGIN ↱CASESTAT↳                                                              PAS
    LMIN := 0 ; LMAX := 0 ; LPTR := NIL ; LERR := ERR ; ERR := FALSE ;          PAS
    INSYMBOL ; EXPRESSION ;                                                     PAS
    LCTP := GATTR.TYPTR ;                                                       PAS
    IF LCTP ≠ NIL THEN                                                          PAS
    BEGIN IF LCTP↑.FORM = NUMERIC THEN LCTP := INTPTR ELSE                      PAS
      IF (LCTP↑.FORM > SYMBOLIC)∨(LCTP = REALPTR)∨(LCTP = ALFAPTR) THEN         PAS
      BEGIN ERROR(62) ; LCTP := NIL END ;                                       PAS
      TRANSFER(GATTR,RP) ;                                                      PAS
    END ;                                                                       PAS
    IF INXCHECK THEN                                                            PAS
    BEGIN GEN30(71B,7,0,IC) ; GEN30(71B,2,0,0) ;                                PAS
```

LIST OF CONTROL, ACTIVE, AND/OR INACTIVE CARDS IN PASCSYM

000048

```
      LCA1 := CA ; LCP1 := CP ; GEN15(37B,0,1,2) ;                                    PA
      GEN30(71B,2,0,0) ; LCA2 := CA ; LCP2 := CP ;                                    PA
      GEN15(37B,2,2,1) ; GEN15(12B,2,2,0) ;                                           PA
      GEN30(03B,3,2,INXERR)                                                           PA
    END ;                                                                             PA
    GEN15(63B,7,1,0) ; GEN30(02B,7,0,0) ; LCA := CA ; LCP := CP ;                     PA
    IF NO ≠ 27 THEN ↱OF↳                                                              PA
    BEGIN IF LCTP ≠ NIL THEN ERROR(65) ; SKIP(27) ;                                   PA
      IF NO ≠ 27 THEN                                                                 PA
      IF LCTP = NIL THEN ERROR(65) ;                                                  PA
    END ;                                                                             PA
    ACASE ; INS(IC-LMIN,LCP,LCA) ; PT1 := LPTR ;                                      PA
    IF INXCHECK THEN                                                                  PA
    BEGIN INS(LMIN,LCP1,LCA1) ; INS(LMAX,LCP2,LCA2) ; END ;                           PA
    IT := LMIN ; IT1 := IC + LMAX - LMIN + 1 ; NOOP ;                                 PA
    IF ¬ERR THEN                                                                      PA
    WHILE PT1 ≠ NIL DO                                                                PA
    BEGIN WHILE PT1↑.CSLAB ≠ IT DO                                                    PA
      BEGIN IF INXCHECK THEN                                                          PA
        BEGIN GEN30(71B,7,0,IC) ; GEN30(04B,0,0,INXERR) END ELSE                      PA
        BEGIN GEN30(04B,0,0,IT1) ; NOOP END ;                                         PA
        IT := IT + 1                                                                  PA
      END ;                                                                           PA
      GEN30(04B,0,0,PT1↑.ADDR) ; NOOP ;                                               PA
      PT1 := PT1↑.NEXT ; IT := IT + 1 ;                                               PA
    END ;                                                                             PA
    IF NO = 22 THEN ↱END↳ INSYMBOL ;                                                  PA
    ERR := ERR ∨ LERR ;                                                               PA
  END ↱CASESTAT↳ ;                                                                    PA
                                                                                      PA
      PROCEDURE REPEATSTAT ;                                                          PA
      VAR LJPADDR: ADDRESS ;                                                          PA
      BEGIN                                                                           PA
         NOOP ; LJPADDR := IC ;                                                       PA
         REPEAT                                                                       PA
         BEGIN                                                                        PA
            INSYMBOL ;                                                                PA
20:         STATEMENT ;                                                             PA
            IF ERRCL[NO] = BEGSY THEN BEGIN ERROR(58) ; GOTO 20 END ;                 PA
            IF NO = 25 THEN  ↱ELSE↳                                                   PA
            BEGIN ERROR(54) ; INSYMBOL ; GOTO 20 END                                  PA
         END                                                                          PA
         UNTIL NO ≠ 16 ;  ↱ SY ≠ ; ↳                                                 PA
         IF NO ≠ 29 THEN  ↱ SY ≠ UNTIL ↳  ERROR(67) ELSE                              PA
         BEGIN                                                                        PA
            INSYMBOL ; EXPRESSION ;                                                   PA
            IF GATTR.TYPTR ≠ NIL THEN GENJP(LJPADDR) ELSE SKIP(49)                    PA
         END                                                                          PA
      END  ↱ REPEATSTAT ↳  ;                                                          PA
                                                                                      PA
      PROCEDURE WHILESTAT ;                                                           PA
      VAR LJPADDR : ADDRESS;  LCA,LCP : SHRTINT;                                      PA
```

LIST OF CONTROL, ACTIVE, AND/OR INACTIVE CARDS IN PASCSYM 000049

```
      BEGIN                                                                    PAS
         NOOP ; LJPADDR := IC ;                                                PAS
         INSYMBOL ; EXPRESSION ;                                               PAS
         IF GATTR.TYPTR ≠ NIL THEN GENJP(0) ;                                  PAS
         LCA := CA ; LCP := CP ;                                               PAS
         IF NO ≠ 31 THEN  ↱ SY ≠ DO ↳                                          PAS
         BEGIN                                                                 PAS
            IF GATTR.TYPTR ≠ NIL THEN ERROR(59) ; SKIP(31) ;                   PAS
            IF NO ≠ 31 THEN                                                    PAS
            BEGIN                                                              PAS
               IF GATTR.TYPTR = NIL THEN ERROR(59) ;                           PAS
               IF ERRCL[NO] = BEGSY THEN GOTO 20 ; GOTO 10                     PAS
            END                                                                PAS
         END ;                                                                 PAS
         INSYMBOL ;                                                            PAS
20:      STATEMENT ;                                                           PAS
         GEN30(04B,0,0,LJPADDR) ;                                              PAS
10:      NOOP;  INS(IC,LCP,LCA);                                               PAS
      END  ↱ WHILESTAT ↳  ;                                                    PAS
                                                                               PAS
      PROCEDURE FORSTAT ;                                                      PAS
      VAR LATTR : ATTR ; LCLASS, LCA, LCP : SHRTINT ;                          PAS
         LJPADDR : ADDRESS ; LOF : BOOLEAN ; LUPBND : SHRTINT ;                PAS
                                                                               PAS
      PROCEDURE CHTYPE ;                                                       PAS
      BEGIN                                                                    PAS
         WITH GATTR DO                                                         PAS
         IF TYPTR ≠ NIL THEN                                                   PAS
         IF (TYPTR↑.FORM > SYMBOLIC)∨(TYPTR = REALPTR)∨                        PAS
         (TYPTR = ALFAPTR) THEN                                                PAS
         BEGIN ERROR(62) ; TYPTR := NIL END                                    PAS
      END ;                                                                    PAS
                                                                               PAS
      PROCEDURE CHTYPES ;                                                      PAS
      BEGIN                                                                    PAS
         WITH GATTR DO                                                         PAS
         IF (TYPTR ≠ NIL)∧(LATTR.TYPTR ≠ NIL) THEN                             PAS
         IF((TYPTR↑.FORM = SYMBOLIC)∨(LATTR.TYPTR↑.FORM = SYMBOLIC))∧          PAS
         (TYPTR ≠ LATTR.TYPTR) THEN                                            PAS
         BEGIN ERROR(73) ; TYPTR := NIL END                                    PAS
      END ;                                                                    PAS
                                                                               PAS
      BEGIN LCA := 0 ;                                                         PAS
         INSYMBOL ;                                                            PAS
         IF NO ≠ 1 THEN                                                        PAS
         BEGIN ERROR(49) ; GATTR.TYPTR := NIL END ELSE                         PAS
         BEGIN SEARCH ;                                                        PAS
            IF CTPTR = NIL THEN                                                PAS
            BEGIN ERROR(31) ; CTPTR := UNDECPTR END ;                          PAS
            IF CTPTR↑.KLASS ≤ PROC THEN                                        PAS
            BEGIN ERROR(32) ; INSYMBOL END ELSE VARIABLE                       PAS
         END ;                                                                 PAS
```

LIST OF CONTROL, ACTIVE, AND/OR INACTIVE CARDS IN PASCSYM 000050

```
      CHTYPE ;                                                                  PAS
      IF GATTR.TYPTR ≠ NIL THEN                                                 PAS
      WITH GATTR DO                                                             PAS
      IF ACCESS ≠ DRCT THEN                                                     PAS
      BEGIN ERROR(69) ; TYPTR := NIL END ELSE                                   PAS
      IF ¬(BREG IN [0,LEVEL]) THEN ERROR(69) ;                                  PAS
      LATTR := GATTR ;                                                          PAS
      IF NO ≠ 20 THEN  ↱ SY ≠ := ↳                                              PAS
      BEGIN                                                                     PAS
         IF GATTR.TYPTR ≠ NIL THEN ERROR(52) ; SKIP(20);                        PAS
         IF NO ≠ 20 THEN                                                        PAS
         BEGIN                                                                  PAS
            IF GATTR.TYPTR = NIL THEN ERROR(52) ;                               PAS
            IF ERRCL[NO] = BEGSY THEN GOTO 20 ; GOTO 10                         PAS
         END                                                                    PAS
      END ;                                                                     PAS
      INSYMBOL ; EXPRESSION ;                                                   PAS
      CHTYPE ; CHTYPES ;                                                        PAS
      TRANSFER(GATTR,RP) ;                                                      PAS
      IF NO ≠ 33 THEN  ↱ SY ≠ TO/DOWNTO ↳                                       PAS
      BEGIN                                                                     PAS
         IF GATTR.TYPTR ≠ NIL THEN ERROR(70) ; SKIP(33) ;                       PAS
         IF NO ≠ 33 THEN                                                        PAS
         BEGIN                                                                  PAS
            IF GATTR.TYPTR = NIL THEN ERROR(70) ;                               PAS
            IF ERRCL[NO] = BEGSY THEN GOTO 20 ; GOTO 10                         PAS
         END                                                                    PAS
      END ;                                                                     PAS
      LCLASS := CL ;                                                            PAS
      INSYMBOL ; EXPRESSION ;                                                   PAS
      CHTYPE ; CHTYPES ;                                                        PAS
      IF (GATTR.TYPTR ≠ NIL)∧(LATTR.TYPTR ≠ NIL) THEN                           PAS
      BEGIN                                                                     PAS
         TRANSFER(GATTR,RP) ;                                                   PAS
         IF GATTR.KIND = SVAL THEN                                              PAS
         BEGIN LOF := TRUE ; LUPBND := GATTR.VAL END ELSE                       PAS
         BEGIN GEN15(10B,7,RP,0) ;                                              PAS
            GEN30(51B,7,5,LC+TCT) ; TCT := TCT + 1 ;                            PAS
            IF TCT > TMAX THEN TMAX := TCT ; LOF := FALSE ;                     PAS
         END ;                                                                  PAS
         GEN15(10B,6,1,0) ; NOOP ; LJPADDR := IC ;                              PAS
         IF LCLASS = 1 THEN ↱STEP +1↳ GEN15(37B,1,2,6)                          PAS
                       ELSE ↱STEP -1↳ GEN15(37B,1,6,2) ;                        PAS
         GEN30(03B,3,1,0) ; LCA := CA ; LCP := CP ; RP := 0 ;                   PAS
         IF ASSCHECK THEN                                                       PAS
         WITH LATTR.TYPTR↑ DO                                                   PAS
         IF FORM = NUMERIC THEN                                                 PAS
         BEGIN IF LATTR.TYPTR ≠ INTPTR THEN                                     PAS
            CHECKBNDS(6,MIN,MAX,ASSERR)                                         PAS
         END ELSE CHECKBNDS(6,0,FCONST↑.VALUES,ASSERR) ;                        PAS
         TRANSFER(LATTR,6) ;                                                    PAS
      END ;                                                                     PAS
```

LIST OF CONTROL, ACTIVE, AND/OR INACTIVE CARDS IN PASCSYM 000051

```
        IF NO ≠ 31 THEN  ↱ SY ≠ DO ↳                                          PAS
        BEGIN                                                                  PAS
           IF GATTR.TYPTR ≠ NIL THEN ERROR(59) ; SKIP(31) ;                    PAS
           IF NO ≠ 31 THEN                                                     PAS
           BEGIN                                                               PAS
              IF GATTR.TYPTR = NIL THEN ERROR(59) ;                            PAS
              IF ERRCL[NO] = BEGSY THEN GOTO 20 ; GOTO 10                      PAS
           END                                                                 PAS
        END ;                                                                  PAS
        INSYMBOL ;                                                             PAS
20:     STATEMENT ;                                                            PAS
        IF LATTR.TYPTR ≠ NIL THEN                                              PAS
        BEGIN TRANSFER(LATTR,RP) ;                                             PAS
           WITH LATTR.TYPTR↑ DO                                                PAS
           IF (FORM = NUMERIC)∧((ABS(MIN) ≥ TWOTO17)∨(ABS(MAX) ≥               PAS
           TWOTO17)) THEN                                                      PAS
           BEGIN GEN30(71B,0,0,1) ;                                            PAS
              GEN15(35B+LCLASS,6,1,0) ;  ↱ +1 OR -1↳                          PAS
           END ELSE GEN30(72B,6,1,3-2*LCLASS) ;                                PAS
           IF LOF THEN GEN30(71B,2,0,LUPBND) ELSE                              PAS
           BEGIN TCT := TCT - 1 ; GEN30(51B,2,5,LC+TCT) END ;                  PAS
           GEN30(04B,0,0,LJPADDR) ; NOOP ;                                     PAS
           INS(IC,LCP,LCA);                                                    PAS
        END ;                                                                  PAS
10:  END  ↱ FORSTAT ↳  ;                                                       PAS
                                                                               PAS
     PROCEDURE GOTOSTAT ;                                                      PAS
     BEGIN INSYMBOL ;                                                          PAS
           IF (NO = 1)∧(AVAL = ΞEXITΞ) THEN                                    PAS
           BEGIN INSYMBOL ;                                                    PAS
              IF (NO ≠ 2)∨(CL ≠ 1) THEN                                        PAS
              BEGIN ERROR(61) ; SKIP(49) END ELSE                              PAS
              BEGIN FOR IT := CEXTABIX DOWNTO 1 DO                             PAS
                 IF EXTAB[IT].EXVAL = IVAL THEN                                PAS
                 BEGIN IT1 := EXTAB[IT].JMPTABIX ;                             PAS
                    IF JMPTAB[IT1] ≠ 0 THEN                                    PAS
                    BEGIN IF JMPTAB[IT1] ≠ LEVEL THEN                          PAS
                       BEGIN LOADBASE(1,JMPTAB[IT1]) ; GEN15(63B,5,1,0)        PAS
                       END                                                     PAS
                    END ELSE                                                   PAS
                    BEGIN GEN30(51B,1,0,PARMLIST.IC0) ;                        PAS
                       GEN15(63B,5,1,0)                                        PAS
                    END ;                                                      PAS
                    GEN30(04B,0,0,PARMLIST.LPJMPTAB+IT1) ; GOTO 1              PAS
                 END ;                                                         PAS
                 ERROR(43) ;                                                   PAS
1:               INSYMBOL                                                      PAS
              END                                                              PAS
           END ELSE                                                            PAS
           IF (NO ≠ 2)∨(CL ≠ 1) THEN                                           PAS
           BEGIN ERROR(61) ; SKIP(49) END ELSE                                 PAS
           BEGIN IF IVAL ≥ TWOTO17 THEN                                        PAS
```

LIST OF CONTROL, ACTIVE, AND/OR INACTIVE CARDS IN PASCSYM 000052

```
            BEGIN ERROR(100) ; GOTO 20 END ;                                    PAS
             ↱ SEARCH THROUGH LABELTABLE OF CURRENT BLOCK ↓                     PAS
            FOR IT := 1 TO CLABIX DO                                            PAS
            WITH LABTAB[IT] DO                                                  PAS
            BEGIN                                                               PAS
               IF LABVAL = IVAL THEN  ↱ LABEL ALREADY OCCURED ↓                 PAS
               BEGIN                                                            PAS
                   ↱ IF DECL. OCC. GENERATE CODE ELSE CHAIN OCC. ↓               PAS
                  IF FLD2 = 0 THEN GEN30(04B,0,0,FLD3) ELSE                     PAS
                  BEGIN                                                         PAS
                     GEN30(04B,0,0,0) ;                                         PAS
                     IF CHNIX = 0 THEN                                          PAS
                     BEGIN ERROR(75) ; GOTO 20 END ;                            PAS
                     WITH UNDLAB[CHNIX] DO                                      PAS
                     BEGIN                                                      PAS
                        IT1 := SUCC ; SUCC := FLD3 ;                            PAS
                        FLD3 := CHNIX ; PLACE := CA ;                           PAS
                        LFTSH := CP ;                                           PAS
                     END ;                                                      PAS
                     CHNIX := IT1                                               PAS
                  END ;                                                         PAS
                  GOTO 20                                                       PAS
               END                                                              PAS
            END ;                                                               PAS
             ↱ LABEL NOT YET MET, ENTER IT INTO LABELTABLE ↓                    PAS
            GEN30(04B,0,0,0) ;                                                  PAS
            IF CLABIX = MAXLABS THEN BEGIN ERROR(74) ; GOTO 20 END ;            PAS
            IF CHNIX = 0 THEN BEGIN ERROR(75) ; GOTO 20 END ;                   PAS
            CLABIX := CLABIX + 1 ;                                              PAS
            WITH LABTAB[CLABIX], UNDLAB[CHNIX] DO                               PAS
            BEGIN                                                               PAS
               IT1 := SUCC ; LABVAL := IVAL ;                                   PAS
               SUCC := 0 ; FLD2 := CHNIX ;                                      PAS
               FLD3 := CHNIX ; PLACE := CA ;                                    PAS
               LFTSH := CP                                                      PAS
            END ;                                                               PAS
            CHNIX := IT1 ;                                                      PAS
20:         INSYMBOL ;                                                          PAS
         END ;                                                                  PAS
   END  ↱ GOTOSTAT ↓  ;                                                         PAS
                                                                                PAS
   PROCEDURE WITHSTAT ;                                                         PAS
   VAR LTPCT, LB7CT: SHRTINT ;                                                  PAS
   BEGIN                                                                        PAS
      LTPCT := 0 ; LB7CT := 0 ;                                                 PAS
      REPEAT                                                                    PAS
         INSYMBOL ;                                                             PAS
         VARIAB ;                                                               PAS
         WITH GATTR DO                                                          PAS
         IF TYPTR ≠ NIL THEN                                                    PAS
         IF TYPTR↑.FORM ≠ RECORDS THEN ERROR(38) ELSE                           PAS
         BEGIN                                                                  PAS
```

```
            TOP := TOP + 1 ; LTPCT := LTPCT + 1 ;                                  PAS
            IF TOP > DISPLIMIT THEN ERROR(82) ELSE                                 PAS
            WITH DISPLAY[TOP] DO                                                   PAS
            BEGIN                                                                  PAS
               FNAME := TYPTR↑.FSTFLD ;                                            PAS
               IF (ACCESS = DRCT)∧(BREG IN [0,LEVEL]) THEN                         PAS
               BEGIN OCCUR := CWITH;                                               PAS
                  CDSPL := DPLMT;  CLEV := BREG;                                   PAS
               END ELSE                                                            PAS
               BEGIN                                                               PAS
                  LOADADR(GATTR,6) ; IT := LC + TCT ;                              PAS
                  GEN30(51B,6,5,IT) ;                                              PAS
                  OCCUR := VWITH ; VDSPL := IT ;                                   PAS
                  TCT := TCT + 1 ; LB7CT := LB7CT + 1 ;                            PAS
                  IF TCT > TMAX THEN TMAX := TCT ;                                 PAS
               END                                                                 PAS
            END                                                                    PAS
         END                                                                       PAS
      UNTIL NO ≠ 15 ;  ↱ SY ≠ , ↳                                                 PAS
      IF NO ≠ 31 THEN                                                              PAS
      BEGIN                                                                        PAS
         IF GATTR.TYPTR ≠ NIL THEN ERROR(59) ; SKIP(31) ;                          PAS
         IF NO ≠ 31 THEN                                                           PAS
         BEGIN                                                                     PAS
            IF GATTR.TYPTR = NIL THEN ERROR(59) ;                                  PAS
         IF ERRCL[NO] = BEGSY THEN GOTO 20 ; GOTO 10                               PAS
         END                                                                       PAS
      END ;                                                                        PAS
      INSYMBOL ;                                                                   PAS
20:   STATEMENT ;                                                                  PAS
10:   TOP := TOP - LTPCT ; TCT := TCT - LB7CT ;                                    PAS
   END  ↱ WITHSTAT ↳  ;                                                            PAS
                                                                                   PAS
                                                                                   PAS
↱ ------------------------------------------------------------------ ↳            PAS
                                                                                   PAS
PROCEDURE COMPSTAT ;                                                               PAS
BEGIN REPEAT BEGIN                                                                 PAS
      INSYMBOL ;                                                                   PAS
1:    STATEMENT ;                                                                  PAS
      IF ERRCL[NO] = BEGSY THEN                                                    PAS
   BEGIN ERROR(58) ; GOTO 1 END ;                                                  PAS
   IF NO = 25 THEN  ↱ELSE↳                                                         PAS
   BEGIN ERROR(54) ; INSYMBOL ; GOTO 1 END                                         PAS
   END                                                                             PAS
 UNTIL NO ≠ 16 ↱;↳ ;                                                               PAS
 IF NO ≠ 22 THEN ↱END↳ ERROR(68) ELSE INSYMBOL ;                                   PAS
END ↱COMPSTAT↳ ;                                                                   PAS
                                                                                   PAS
PROCEDURE STATEMENT ;                                                              PAS
   VAR LPSW : SHRTINT;                                                             PAS
BEGIN                                                                              PAS
```

LIST OF CONTROL, ACTIVE, AND/OR INACTIVE CARDS IN PASCSYM 000054

```
    IF (NO = 2)^(CL = 1) THEN ↱LABEL↳                                          PA
    BEGIN NOOP ;                                                               PA
      IF IVAL ≥ TWOTO17 THEN                                                   PA
      BEGIN ERROR(100) ; GOTO 1 END ;                                          PA
      FOR IT := 1 TO CLABIX DO                                                 PA
      WITH LABTAB[IT] DO                                                       PA
      IF LABVAL = IVAL THEN ↱FOUND↳                                            PA
      BEGIN IF FLD2 = 0 THEN ↱MULTIDEF.↳ ERROR(77) ELSE ↱FIXUP↳                PA
        BEGIN IT1 := FLD3 ;                                                    PA
          REPEAT WITH UNDLAB[IT1] DO                                           PA
            BEGIN INS(IC,LFTSH,PLACE) ; IT1 := SUCC END                        PA
          UNTIL IT1 = 0 ;                                                      PA
          IT1 := FLD2 ; UNDLAB[IT1].SUCC := CHNIX ;                            PA
          CHNIX := FLD3 ; FLD2 := 0 ; FLD3 := IC ;                             PA
        END ;                                                                  PA
        GOTO 1                                                                 PA
      END ↱IF, WITH, FOR↳ ;                                                    PA
      ↱NEW LABEL↳                                                              PA
      IF CLABIX = MAXLABS THEN ERROR(74) ELSE                                  PA
      BEGIN CLABIX := CLABIX + 1 ;                                             PA
        WITH LABTAB[CLABIX] DO                                                 PA
        BEGIN LABVAL := IVAL ; FLD2 := 0 ; FLD3 := IC END ;                    PA
      END ;                                                                    PA
1:    INSYMBOL ;                                                               PA
      IF NO ≠ 19 THEN ↱:↳                                                      PA
      BEGIN ERROR(64) ; SKIP(49) END ELSE INSYMBOL ;                           PA
    END ↱IF (NO=2)^(CL=1)↳ ;                                                   PA
    RP := 0 ;                                                                  PA
    CASE SPLITSTAT[NO] OF                                                      PA
    ↱PASS ↳  1:  ↱ENDSY OR IRRELSY↳ ;                                          PA
    ↱IDENT ↳ 2:  BEGIN SEARCH ;                                                PA
                   IF CTPTR = NIL THEN                                         PA
                   BEGIN ERROR(31) ; CTPTR := UNDECPTR END ;                   PA
                   WITH CTPTR↑ DO                                              PA
                   IF KLASS ≤ KONST THEN ERROR(55) ELSE                        PA
                   IF (KLASS = PROC)^                                          PA
                   ((PROCTYPE = CTPTR)v(PROCTYPE = NIL)) THEN                  PA
                   BEGIN ↱ PROCCALL ↳                                          PA
                     IF PROCTYPE = CTPTR THEN                                  PA
                     BEGIN INSYMBOL ; IF CTPTR ≤ PREDEFP THEN                  PA
                       BEGIN LPSW := CTPTR↑.SEGSIZE ;                          PA
                        CASE LPSW OF                                           PA
                     0:    TITLE ;                                             PA
                 1,2,3,4:  GETPUT(LPSW) ;                                      PA
                     5:    ALLC ;                                              PA
                     6:    PCK ;                                               PA
                     7:    UNPCK ;                                             PA
                   8,9:    INSAPP(LPSW) ;                                      PA
                    10:    READIR ;                                            PA
                    11:    WRITEIR ;                                           PA
                       END ;                                                   PA
                       END ELSE  PASSPARAMS ;                                  PA
```

```
                    END ELSE SKIP(49)                                          PAS
                  END ↱PROCCALL↳ ELSE  ASSIGN ;                                PAS
                END ;                                                          PAS
    ↱BEGIN ↳  3:  COMPSTAT ;                                                   PAS
    ↱IF    ↳  4:  IFSTAT ;                                                     PAS
    ↱CASE  ↳  5:  CASESTAT ;                                                   PAS
    ↱REPEAT↳  6:  REPEATSTAT ;                                                 PAS
    ↱WHILE ↳  7:  WHILESTAT ;                                                  PAS
    ↱FOR   ↳  8:  FORSTAT ;                                                    PAS
    ↱GOTO  ↳  9:  GOTOSTAT ;                                                   PAS
    ↱WITH  ↳ 10:  WITHSTAT ;                                                   PAS
    END ;                                                                      PAS
    IF ERRCL[NO] = IRRELSY THEN                                                PAS
    BEGIN ERROR(54) ; SKIP(49) END ;                                           PAS
    RP := 0                                                                    PAS
  END ↱STATEMENT↳ ;                                                            PAS
                                                                               PAS
↱ ------------------------------------------------------------------ ↳         PAS
                                                                               PAS
  PROCEDURE TYPEDECL(VAR TL:SHRTINT; P1:CTP);                                  PAS
  VAR I,L,LL,J,CV,E1,E2,BDISPL : SHRTINT;                                      PAS
     OPT : OPTPWR;  DISPL : ADDRESS;  PACKFLAG,LERR : BOOLEAN;                 PAS
     LASTFLD,PP,P,NXTF,NXTC,NXTA,RTYP : CTP;                                   PAS
                                                                               PAS
     PROCEDURE SKIPT(FNO : SHRTINT) ;                                          PAS
     BEGIN                                                                     PAS
        WHILE (TERRCL[NO] = IRRELSY) ^ (FNO ≠ NO) DO INSYMBOL;                 PAS
     END ↱SKIPT↳;                                                              PAS
                                                                               PAS
     PROCEDURE TYPERR(I : SHRTINT);                                            PAS
     BEGIN TL := 0;  P1 := NIL;                                                PAS
        ERROR(I);  SKIPT(49);                                                  PAS
     END ↱TYPERR↳;                                                             PAS
                                                                               PAS
     PROCEDURE SUBRANGE(VAR VAL1,VAL2 : INTEGER; N1 : CTP;                     PAS
        CONST P : CTP) ;                                                       PAS
        ↱THE FIRST SYMBOL OF SUBRANGE HAS BEEN READ.                           PAS
           THE PROCEDURE RETURNS THE TWO BOUND-VALUES IN VAL1,VAL2,            PAS
           AND RETURNS  N1 = POINTER TO TYPE OF CONSTANTS.                     PAS
           ERRORS : TYPES DO NOT AGREE, TYPE IS NOT INTEGER,CHAR,              PAS
              OR SYMBOLIC,  VAL1 > VAL2.                                       PAS
           P INDICATES BEGINNING OF SEARCH FOR FIRST SYMBOL IF IT IS           PAS
           AN ID.    ↳                                                        PAS
        VAR N2 : CTP;                                                          PAS
     BEGIN INCONST(VAL1,N1,P);                                                 PAS
        IF (N1=NIL) ∨ (N1=REALPTR) ∨ (N1=ALFAPTR) THEN                         PAS
           ERR := TRUE ELSE                                                    PAS
        BEGIN IF NO ≠ 19 THEN ERROR(10) ELSE INSYMBOL ;                        PAS
           INCONST(VAL2,N2,NEXT);                                              PAS
           IF N1 ≠ N2 THEN ERR := TRUE ELSE                                    PAS
           IF VAL1 > VAL2 THEN ERROR(25);                                      PAS
        END;                                                                   PAS
```

LIST OF CONTROL, ACTIVE, AND/OR INACTIVE CARDS IN PASCSYM 000356

```
    END ↱SUBRANGE↳;                                                          PA
                                                                             PA
    PROCEDURE SUBTYPE(VAR I,J : SHRTINT; P : CTP);                           PA
      ↱EITHER A TYPE-ID FOR A SUBRANGE OR AN EXPLICIT SUBRANGE               PA
        ARE PROCESSED.                                                       PA
        RETURNS I = LOWBOUND, J = HIGHBOUND,                                 PA
          P = POINTER TO TYPE OF CONSTANTS ↳                                 PA
    BEGIN                                                                    PA
      IF NO = 1 THEN                                                         PA
      BEGIN SRCHREC(NEXT);                                                   PA
        IF CTPTR = NIL THEN SEARCH;                                          PA
        IF CTPTR = NIL THEN ERROR(12) ELSE                                   PA
        BEGIN IF CTPTR↑.KLASS = TYPES THEN                                   PA
          BEGIN IF CTPTR↑.FORM > SYMBOLIC THEN ERROR(13) ELSE                PA
            BEGIN                                                            PA
              CASE CTPTR↑.FORM OF                                            PA
NUMERIC:        BEGIN I := CTPTR↑.MIN;  J := CTPTR↑.MAX;                     PA
                  P := INTPTR;                                               PA
                END;                                                         PA
SYMBOLIC:       BEGIN IF (CTPTR = REALPTR)∨(CTPTR = ALFAPTR) THEN            PA
                  ERROR(6) ELSE                                              PA
                  BEGIN I := 0 ; J := CTPTR↑.FCONST↑.VALUES ;                PA
                    P := CTPTR ;                                             PA
                  END                                                        PA
                END;                                                         PA
              END ↱CASE↳;                                                    PA
              INSYMBOL;                                                      PA
            END;                                                             PA
          END ↱TYPES↳ ELSE                                                   PA
          IF CTPTR↑.KLASS = KONST THEN                                       PA
            SUBRANGE(I,J,P,CTPTR) ELSE ERROR(63) ;                           PA
        END;                                                                 PA
      END ↱ID↳ ELSE                                                          PA
      IF NO IN [2,7] THEN SUBRANGE(I,J,P,NIL) ELSE                           PA
        ERROR(1) ;                                                           PA
    END ↱SUBTYPE↳;                                                           PA
                                                                             PA
    PROCEDURE SCALDECL(N : CTP);                                             PA
    BEGIN SUBRANGE(I,J,PP,N);                                                PA
      IF ¬ERR THEN                                                           PA
      IF PP↑.FORM = SYMBOLIC THEN                                            PA
      BEGIN ERROR(28);  P1 := NIL  END ELSE                                  PA
      BEGIN ALLOC(P,TYPES,NUMERIC);                                          PA
        WITH P↑ DO                                                           PA
        BEGIN NAME := BLANK;  NXTEL := NIL;  KLASS := TYPES;                 PA
          SIZE := 1;  FORM := NUMERIC;                                       PA
          MIN := I;  MAX := J;                                               PA
          IF ABS(I) > ABS(J) THEN BITS := LOG2(ABS(I)) ELSE                  PA
                                  BITS := LOG2(ABS(J)) ;                     PA
        END;                                                                 PA
        TL := 1;  P1 := P;                                                   PA
      END ELSE P1 := NIL                                                     PA
```

```
    END ↑SCALDECL↓;                                                          PAS
                                                                             PAS
  PROCEDURE FIELDLIST(VAR MAXSIZE:SHRTINT; VARPTR,NXTF:CTP);                 PAS
    VAR MXL,L,LL,MINSIZE,CASEBITS,I : SHRTINT;                               PAS
       P,PP,PP1,PP2,NXTC,NXT,CPTR : CTP;                                     PAS
       TAGFLAG : BOOLEAN;                                                    PAS
                                                                             PAS
    PROCEDURE RECERR(I : SHRTINT);                                           PAS
    BEGIN ERROR(I);  SKIPT(49); END;                                         PAS
                                                                             PAS
    PROCEDURE ADJUST;                                                        PAS
       ↑ MOVE LAST FIELD TO RIGHT. IF IT IS THE ONLY FIELD,                  PAS
          CHANGE TO NONPACKED.                                               PAS
          IF LAST FIELD IS TAGFIELD THEN DO NOT MOVE.                        PAS
          INCREASE DISPL, RESET BDISPL  ↓                                    PAS
    BEGIN IF ~TAGFLAG THEN                                                   PAS
       WITH LASTFLD↑ DO                                                      PAS
       BEGIN IF BITDISPL = 0 THEN ↑ONLY ONE FIELD IN WORD↓                   PAS
             BITWIDTH := 0 ELSE                                              PAS
             BITDISPL := WORDLENGTH - BITWIDTH;                              PAS
       END;                                                                  PAS
       DISPL := DISPL + 1;  BDISPL := 0;                                     PAS
    END ↑ADJUST↓;                                                            PAS
                                                                             PAS
  BEGIN                                                                      PAS
    TAGFLAG := TRUE;    NXT := NXTF;                                         PAS
    REPEAT                                                                   PAS
       IF NO = 26 THEN ↑CASE↓ GOTO 2 ;                                       PAS
       I := 0;                                                               PAS
1:     IF NO ≠ 1 THEN                                                        PAS
       BEGIN RECERR(11);                                                     PAS
          IF TERRCL[NO] = BEGSY THEN GOTO 11;  GOTO 12;                      PAS
       END;                                                                  PAS
       SRCHREC(NXT);                                                         PAS
       IF CTPTR ≠ NIL THEN ERROR(5) ELSE                                     PAS
       BEGIN ALLOC(P,FIELD); I := I + 1;                                     PAS
          WITH P↑ DO                                                         PAS
          BEGIN NAME := AVAL;  NXTEL := NXT;  KLASS := FIELD;                PAS
             FLDTYPE := NIL;                                                 PAS
          END;  NXT := P;                                                    PAS
       END;                                                                  PAS
       INSYMBOL;                                                             PAS
       IF NO = 15 THEN ↑,↓                                                   PAS
       BEGIN INSYMBOL;  GOTO 1 END;                                          PAS
       IF NO ≠ 19 THEN ↑NOT :↓ ERROR(10) ELSE INSYMBOL;                      PAS
11:    TYPEDECL(L,P);                                                        PAS
       IF P ≠ NIL THEN                                                       PAS
       IF P↑.FORM > RECORDS THEN ERROR(30) ELSE                              PAS
       BEGIN IF PACKFLAG THEN                                                PAS
          BEGIN IF I > 1 THEN ↑REVERSE POINTERS↓                             PAS
             BEGIN PP := NXT;                                                PAS
               FOR I := I DOWNTO 1 DO                                        PAS
```

LIST OF CONTROL, ACTIVE, AND/OR INACTIVE CARDS IN PASCSYM

000053

```
                    BEGIN PP1 := PP↑.NXTEL;                                    PA
                      PP↑.NXTEL := PP2;  PP2 := PP;                            PA
                      PP := PP1;                                               PA
                    END;                                                       PA
                    NXT↑.NXTEL := PP;  NXT := PP2;                             PA
                  END ↱REVERSE↓ ELSE PP := NXT↑.NXTEL;                         PA
                  PP1 := NXT;                                                  PA
                  IF P↑.FORM ≤ POWER THEN                                      PA
                  BEGIN                                                        PA
                    CASE P↑.FORM OF                                            PA
NUMERIC:              LL := P↑.BITS;                                           PA
SYMBOLIC:             LL := P↑.BITSIZE;                                        PA
POINTER:              LL := 18 ;                                               PA
POWER:                LL := P↑.PWBITS;                                         PA
                    END;                                                       PA
                    REPEAT                                                     PA
                      IF BDISPL + LL > WORDLENGTH THEN ADJUST;                 PA
                      IF LL = WORDLENGTH THEN                                  PA
                      BEGIN PP1↑.BITWIDTH := 0;                                PA
                        PP1↑.FLDADDR := DISPL;  DISPL := DISPL + 1;            PA
                      END ELSE                                                 PA
                      BEGIN PP1↑.BITWIDTH := LL;                               PA
                        PP1↑.BITDISPL := BDISPL;                               PA
                        PP1↑.FLDADDR := DISPL;  BDISPL := BDISPL + LL;         PA
                      END;                                                     PA
                      PP1↑.FLDTYPE := P;                                       PA
                      LASTFLD := PP1;  TAGFLAG := FALSE;                       PA
                      PP1 := PP1↑.NXTEL;                                       PA
                    UNTIL PP1 = PP;                                            PA
                  END ↱FORM ≤ POINTER↓ ELSE                                    PA
                  BEGIN IF BDISPL ≠ 0 THEN ADJUST;                             PA
                    TAGFLAG := FALSE;                                          PA
                    REPEAT                                                     PA
                    PP1↑.BITWIDTH := 0;  PP1↑.FLDTYPE := P;                    PA
                      PP1↑.FLDADDR := DISPL;                                   PA
                      DISPL := DISPL + L;  LASTFLD := PP1;                     PA
                      PP1 := PP1↑.NXTEL;                                       PA
                    UNTIL PP1 = PP;                                            PA
                  END;                                                         PA
                END ↱PACKFLAG↓ ELSE                                            PA
                BEGIN LL := DISPL + I*L;  DISPL := LL;                         PA
                  PP := NXT;                                                   PA
                  FOR I := I DOWNTO 1 DO                                       PA
                  BEGIN PP↑.FLDTYPE := P;  LL := LL - L;                       PA
                    PP↑.FLDADDR := LL;  PP↑.BITWIDTH := 0;                     PA
                    PP := PP↑.NXTEL;                                           PA
                  END;                                                         PA
              END;                                                             PA
            END ↱FORM ≤ RECORDS↓;                                              PA
12:     IF NO = 16 THEN INSYMBOL;                                              PA
      UNTIL (TERRCL[NO] = ENDSY) ∧ (NO ≠ 26);                                  PA
      IF BDISPL ≠ 0 THEN ADJUST;                                               PA
```

LIST OF CONTROL, ACTIVE, AND/OR INACTIVE CARDS IN PASCSYM

000059

```
      MAXSIZE := DISPL;  VARPTR := NIL;                                    PAS
      GOTO 9;                                                              PAS
 2:   ↱CASE↳                                                               PAS
      INSYMBOL;                                                            PAS
      IF NO ≠ 1 THEN ERROR(11) ELSE                                        PAS
      BEGIN SRCHREC(NXT);                                                  PAS
         IF CTPTR ≠ NIL THEN ERROR(5) ELSE                                 PAS
         BEGIN ALLOC(P,FIELD);                                             PAS
            WITH P↑ DO                                                     PAS
            BEGIN NAME := AVAL;  NXTEL := NXT;  KLASS := FIELD;            PAS
               FLDTYPE := NIL;                                             PAS
            END;  NXT := P;                                                PAS
         END;                                                              PAS
         INSYMBOL;                                                         PAS
      END ↱NO = 1↳;                                                        PAS
         IF NO ≠ 19 THEN ERROR(10) ELSE INSYMBOL;                          PAS
         IF NO ≠ 1 THEN ERROR(11) ELSE                                     PAS
         BEGIN SRCHREC(NEXT);                                              PAS
            IF CTPTR = NIL THEN SEARCH;                                    PAS
            IF CTPTR = NIL THEN ERROR(12) ELSE                             PAS
            IF (CTPTR↑.KLASS≠TYPES) ∨ (CTPTR↑.FORM>SYMBOLIC) THEN          PAS
               ERROR(7);                                                   PAS
            INSYMBOL;                                                      PAS
               IF NO ≠ 27 THEN ERROR(14);                                  PAS
               CPTR := CTPTR;                                              PAS
               IF CPTR ≠ NIL THEN                                          PAS
               IF PACKFLAG THEN                                            PAS
               BEGIN                                                       PAS
                    CASE CPTR↑.FORM OF                                     PAS
   NUMERIC:           LL := CPTR↑.BITS;                                    PAS
   SYMBOLIC:          LL := CPTR↑.BITSIZE;                                 PAS
                    END;                                                   PAS
                    IF BDISPL + LL > WORDLENGTH THEN ADJUST;               PAS
                    P↑.BITDISPL := BDISPL;  P↑.BITWIDTH := LL;             PAS
                    P↑.FLDADDR := DISPL;  BDISPL := BDISPL + LL;           PAS
                 LASTFLD := P;  CASEBITS := BDISPL;                        PAS
               END ↱PACKFLAG↳ ELSE                                         PAS
               BEGIN P↑.FLDADDR := DISPL;                                  PAS
                  P↑.BITWIDTH := 0;                                        PAS
               END;                                                        PAS
               P↑.FLDTYPE := CPTR;  DISPL := DISPL + 1;                    PAS
               MINSIZE := DISPL;  MAXSIZE := DISPL;                        PAS
               NXTC := NIL;  INSYMBOL;                                     PAS
               REPEAT  I := 0;                                             PAS
                  REPEAT                                                   PAS
                    IF (NO>2) ∨ (NO=2) ∧ ¬(CL IN [1,4]) THEN               PAS
                    BEGIN RECERR(63);  GOTO 3 END ;                        PAS
                    IF CPTR ≠ NIL THEN                                     PAS
                      IF NO = 1 THEN                                       PAS
                      BEGIN SRCHREC(NEXT);                                 PAS
                        IF CTPTR = NIL THEN SEARCH;                        PAS
                        IF CTPTR = NIL THEN ERROR(12) ELSE                 PAS
```

LIST OF CONTROL, ACTIVE, AND/OR INACTIVE CARDS IN PASCSYM 000060

```
                    WITH CTPTR↑ DO                                          PAS
                      IF KLASS ≠ KONST THEN ERROR(63) ELSE                  PAS
                      IF (CPTR↑.FORM = SYMBOLIC) ∧                          PAS
                          (CONTYPE ≠ CPTR)  ∨                               PAS
                        (CPTR↑.FORM = NUMERIC) ∧                            PAS
                          (CONTYPE ≠ INTPTR) THEN ERROR(73) ELSE            PAS
                      IT := VALUES;                                         PAS
                  END ↱NO = 1↴ ELSE                                         PAS
                  IF (CL = 1) ∧ (CPTR↑.FORM ≠ NUMERIC)                      PAS
                    ∨ (CL = 4) ∧ (CPTR ≠ CHARPTR) THEN                      PAS
                    ERROR(73) ELSE                                          PAS
                  IT := IVAL;  ↱END CPTR ≠ NIL↴                             PAS
                  ALLOC(P,TAGFIELD);                                        PAS
                  WITH P↑ DO                                                PAS
                  BEGIN NAME := BLANK;  NXTEL := NXTC;                      PAS
                    KLASS := TAGFIELD;                                      PAS
                    TAGVAL := TRUE;  CASEVAL := IT;                         PAS
                  END;  NXTC := P;  I := I + 1;                             PAS
                  INSYMBOL;                                                 PAS
3:                IF NO ≠ 19 THEN ERROR(10) ELSE INSYMBOL;                  PAS
                UNTIL NO > 2;                                               PAS
                IF NO = 9 THEN  ↱ ( FIELDLIST ) ↴                           PAS
                BEGIN IF PACKFLAG THEN                                      PAS
                  BEGIN DISPL := MINSIZE - 1;                               PAS
                    BDISPL := CASEBITS;                                     PAS
                  END ELSE DISPL := MINSIZE;                                PAS
                  INSYMBOL;  FIELDLIST(MXL,PP,NXT);                         PAS
                  IF NO ≠ 10 THEN ERROR(9) ELSE INSYMBOL;                   PAS
                END ELSE                                                    PAS
                BEGIN PP := NIL;  MXL := MINSIZE END;                       PAS
                P := NXTC;                                                  PAS
                FOR I := I DOWNTO 1 DO                                      PAS
                WITH P↑ DO                                                  PAS
                BEGIN CASESIZE := MXL;  VARIANTS := PP;                     PAS
                  P := NXTEL;                                               PAS
                END;  IF MAXSIZE < MXL THEN MAXSIZE := MXL;                 PAS
                IF NO = 16 THEN INSYMBOL;                                   PAS
              UNTIL NO > 2;                                                 PAS
              ALLOC(P,TAGFIELD);                                            PAS
              WITH P↑ DO                                                    PAS
              BEGIN NAME := BLANK;  NXTEL := NIL;  KLASS := TAGFIELD;       PAS
                CASESIZE := MINSIZE;  VARIANTS := NXTC;                     PAS
                TAGVAL := FALSE;  CASETYPE := CPTR;                         PAS
              END;                                                          PAS
              VARPTR := P;                                                  PAS
          END ↱NO = 1↴;                                                     PAS
9:    NXTF := NXT;                                                        PAS
  END ↱ FIELDLIST ↴;                                                      PAS
                                                                          PAS
  ↱TYPEDECL↴                                                              PAS
  BEGIN PACKFLAG := FALSE;                                                PAS
    IF NO = 42 THEN  ↱PACKED↴                                             PAS
```

LIST OF CONTROL, ACTIVE, AND/OR INACTIVE CARDS IN PASCSYM

000061

```
      BEGIN PACKFLAG := TRUE;  INSYMBOL END;                                    PAS
      IF NO = 1 THEN                                                           PAS
      BEGIN SRCHREC(NEXT);                                                     PAS
         IF CTPTR = NIL THEN SEARCH;                                           PAS
         IF CTPTR = NIL THEN                                                   PAS
         BEGIN ERROR(12);  P1 := NIL;  SKIPT(16) END ELSE                      PAS
         BEGIN IF CTPTR↑.KLASS = TYPES THEN ↱TYPE-ID↳                          PAS
            BEGIN TL := CTPTR↑.SIZE;  P1 := CTPTR;  INSYMBOL END ELSE          PAS
            IF CTPTR↑.KLASS = KONST THEN SCALDECL(CTPTR) ELSE                  PAS
            TYPERR(11);                                                        PAS
         END;                                                                  PAS
      END ↱ID↳ ELSE                                                            PAS
      IF NO = 9 THEN  ↱ SYMBOLIC ↳                                             PAS
      BEGIN CV := 0; LERR := ERR ; ERR := FALSE;ALLOC(P,TYPES,SYMBOLIC);       PAS
         WITH P↑ DO                                                            PAS
         BEGIN NAME := BLANK;  NXTEL := NIL;  KLASS := TYPES;                  PAS
            FORM := SYMBOLIC;                                                  PAS
         END;  RTYP := P;  NXTC := NIL;                                        PAS
         REPEAT INSYMBOL;                                                      PAS
            IF NO ≠ 1 THEN                                                     PAS
            BEGIN ERROR(11);  SKIPT(15);                                       PAS
               GOTO 2;                                                         PAS
            END;                                                               PAS
            SRCHREC(NEXT);                                                     PAS
               IF CTPTR ≠ NIL THEN ERROR(8) ELSE                               PAS
               BEGIN ALLOC(P,KONST,ACTUAL);                                    PAS
                  WITH P↑ DO                                                   PAS
                  BEGIN NAME := AVAL;  NXTEL := NEXT;  KLASS := KONST;         PAS
                     CONTYPE := RTYP;  CONKIND := ACTUAL;                      PAS
                     VALUES := CV;  SUCC := NXTC;                              PAS
                  END;  CV := CV + 1;  NEXT := P;  NXTC := P;                  PAS
               END;                                                            PAS
               INSYMBOL;                                                       PAS
2:       UNTIL NO ≠ 15;                                                        PAS
            ALLOC(P,TYPES,POWER) ;                                             PAS
            WITH P↑ DO                                                         PAS
            BEGIN NAME := BLANK;  NXTEL := NIL;  KLASS := TYPES;               PAS
               SIZE := 1;  FORM := POWER;                                      PAS
               ELSET := RTYP;  PWBITS := CV + 1;                               PAS
            END;                                                               PAS
            WITH RTYP↑ DO                                                      PAS
            BEGIN FCONST := NEXT;  SIZE := 1;                                  PAS
               BITSIZE := LOG2(CV - 1);  PWSET := P;                           PAS
            END;                                                               PAS
         IF ERR THEN P1 := NIL ELSE                                            PAS
         BEGIN ERR := LERR ; P1 := RTYP END ;                                  PAS
         TL := 1 ;                                                             PAS
         IF NO ≠ 10 THEN TYPERR(9) ELSE INSYMBOL ;                             PAS
      END  ↱ SYMBOLIC ↳  ELSE                                                  PAS
      IF NO IN [2,7] THEN ↱ SUBRANGE ↳                                         PAS
      BEGIN LERR := ERR; ERR := FALSE;                                         PAS
         SCALDECL(NEXT);                                                       PAS
```

LIST OF CONTROL, ACTIVE, AND/OR INACTIVE CARDS IN PASCSYM 000062

```
      IF ERR THEN ERROR(6) ELSE ERR := LERR                                  PAS
    END ↱SUBRANGE↳ ELSE                                                      PAS
    IF NO = 38 THEN  ↱ STRUCTURED TYPES ↳                                    PAS
    CASE CL OF                                                               PAS
↱ARRAY↳ 1:                                                                   PAS
    BEGIN INSYMBOL;                                                          PAS
      IF NO ≠ 11 THEN                                                        PAS
      BEGIN TYPERR(98);                                                      PAS
        IF TERRCL[NO] = BEGSY THEN TYPEDECL(I,CTPTR); GOTO 19;               PAS
      END;                                                                   PAS
        NXTA := NIL;                                                         PAS
        REPEAT ALLOC(P,TYPES,ARRAYS);                                        PAS
          WITH P↑ DO                                                         PAS
          BEGIN NAME := BLANK;  NXTEL := NIL;  KLASS := TYPES;               PAS
            FORM := ARRAYS;  AELTYPE := NXTA;                                PAS
            ↱ AELTYPE TEMPORARILY LINKS SUBARRAYS ↳                          PAS
          END;  NXTA := P;                                                   PAS
          INSYMBOL;                                                          PAS
          LERR := ERR;  ERR := FALSE;                                        PAS
          SUBTYPE(I,J,P);                                                    PAS
          IF ERR THEN                                                        PAS
          BEGIN ERROR(6);  SKIPT(15); I := 0; J := 0; P := NIL               PAS
          END ELSE                                                           PAS
          ERR := LERR;                                                       PAS
          WITH NXTA↑ DO                                                      PAS
          BEGIN LO := I;  HI := J;  INXTYPE := P END;                        PAS
        UNTIL NO ≠ 15;                                                       PAS
        IF NO ≠ 12 THEN                                                      PAS
        BEGIN ERROR(37); SKIPT(27);                                          PAS
          IF TERRCL[NO] = BEGSY THEN GOTO 11;                                PAS
          IF NO = 27 THEN                                                    PAS
          BEGIN INSYMBOL; GOTO 11 END ;                                      PAS
          IF NO ≠ 12 THEN                                                    PAS
          BEGIN P1 := NIL ; TL := 0; GOTO 19 END;                            PAS
        END;                                                                 PAS
        INSYMBOL;                                                            PAS
          IF NO ≠ 27 THEN ERROR(14) ELSE INSYMBOL;                           PAS
11:       TYPEDECL(TL,CTPTR);                                                PAS
          IF CTPTR ≠ NIL THEN                                                PAS
          IF CTPTR↑.FORM > RECORDS THEN                                      PAS
          BEGIN ERROR(30); CTPTR := NIL END ELSE                             PAS
          BEGIN                                                              PAS
            REPEAT                                                           PAS
              WITH NXTA↑ DO                                                  PAS
              BEGIN MULOPT(TL,E1,E2,OPT);                                    PAS
                OPTTYP := OPT;  EXP1 := E1;  EXP2 := E2;                     PAS
                TL := TL*(HI - LO + 1);                                      PAS
                SIZE := TL;                                                  PAS
                P := AELTYPE;  AELTYPE := CTPTR;                             PAS
              END;                                                           PAS
              CTPTR := NXTA;  NXTA := P;                                     PAS
            UNTIL NXTA = NIL;                                                PAS
```

LIST OF CONTROL, ACTIVE, AND/OR INACTIVE CARDS IN PASCSYM 000063

```
                ↱ NOW TL = SIZE OF ARRAY, CTPTR POINTS TO IT ↓                  PAS
            END;  P1 := CTPTR;                                                  PAS
 19:  END ↱ARRAY↓;                                                              PAS
 ↱RECORD↓ 2:                                                                    PAS
     BEGIN ALLOC(P,TYPES,RECORDS);                                              PAS
        WITH P↑ DO                                                              PAS
        BEGIN NAME := BLANK;  NXTEL := NIL;  KLASS := TYPES;                    PAS
           FORM := RECORDS;                                                     PAS
        END;  RTYP := P;                                                        PAS
        INSYMBOL;  NXTF := NIL;                                                 PAS
        DISPL := 0;  BDISPL := 0;  LERR := ERR;  ERR := FALSE;                  PAS
        FIELDLIST(TL,P,NXTF);                                                   PAS
        IF NO ≠ 22 THEN ERROR(17);                                              PAS
        IF ERR THEN TYPERR(18) ELSE                                             PAS
        WITH RTYP↑ DO                                                           PAS
        BEGIN SIZE := TL;  FSTFLD := NXTF;  RECVAR := P;                        PAS
           P1 := RTYP;  ERR := LERR;                                            PAS
        END;                                                                    PAS
        IF NO = 22 THEN INSYMBOL;                                               PAS
     END ↱RECORD↓;                                                              PAS
 ↱FILE↓ 3:                                                                      PAS
     BEGIN INSYMBOL;                                                            PAS
        IF NO = 27 THEN ↱OF↓                                                    PAS
        BEGIN I := 8 ; INSYMBOL END ELSE                                        PAS
        BEGIN INCONST(I,PT,NEXT) ;                                              PAS
           IF (I≤0)∨(PT≠INTPTR) THEN                                            PAS
           BEGIN ERROR(95) ; I := 0 END ;                                       PAS
           IF NO ≠ 27 THEN ERROR(14) ELSE INSYMBOL ;                            PAS
        END ;                                                                   PAS
        TYPEDECL(TL,CTPTR);                                                     PAS
        IF CTPTR ≠ NIL THEN                                                     PAS
        IF CTPTR↑.FORM > RECORDS THEN                                           PAS
        BEGIN ERROR(30); P1 := NIL END ELSE                                     PAS
        BEGIN L := CTPTR↑.SIZE;                                                 PAS
           LL := L DIV 64 + 1 ;                                                 PAS
           IF LL < I THEN LL := I ;                                             PAS
           LL := ((LL*64) DIV L + 1) * L ;                                      PAS
           ALLOC(P,TYPES,FILES);                                                PAS
           WITH P↑ DO                                                           PAS
           BEGIN NAME := BLANK;  NXTEL := NIL;  KLASS := TYPES;                 PAS
              FORM := FILES;  SIZE := LL;  FELTYPE := CTPTR;                    PAS
           END;                                                                 PAS
           TL := LL;  P1 := P;                                                  PAS
        END ELSE P1 := NIL;                                                     PAS
     END ↱FILE↓;                                                                PAS
 ↱CLASS↓ 4:                                                                     PAS
     BEGIN ALLOC(P,TYPES,CLASSS);                                               PAS
        WITH P↑ DO                                                              PAS
        BEGIN NAME := BLANK;  NXTEL := NIL;  KLASS := TYPES;                    PAS
           FORM := CLASSS;                                                      PAS
        END;                                                                    PAS
        INSYMBOL ;                                                              PAS
```

LIST OF CONTROL, ACTIVE, AND/OR INACTIVE CARDS IN PASCSYM

000064

```
        IF NO = 27 THEN                                                    PAS
        BEGIN I := 100 ; INSYMBOL END ELSE                                 PAS
        BEGIN INCONST(I,PT,NEXT) ;                                         PAS
           IF (I≤0)∨(PT≠INTPTR) THEN                                       PAS
           BEGIN ERROR(6) ; I := 0 END ;                                   PAS
           IF NO ≠ 27 THEN ERROR(14) ELSE INSYMBOL ;                       PAS
        END ;                                                              PAS
        TYPEDECL(TL,CTPTR);                                                PAS
        IF CTPTR ≠ NIL THEN                                                PAS
        IF CTPTR↑.FORM > RECORDS THEN                                      PAS
        BEGIN ERROR(30); P1 := NIL END ELSE                                PAS
        BEGIN TL := TL*I + 1;                                              PAS
           P↑.SIZE := TL;  P↑.PELTYPE := CTPTR;                            PAS
           P1 := P;                                                        PAS
        END ELSE P1 := NIL ;                                               PAS
     END ↱CLASS↳;                                                          PAS
↱POWERSET↳ 5:                                                              PAS
     BEGIN INSYMBOL;  LERR := ERR;  ERR := FALSE;                          PAS
        SUBTYPE(I,J,P);                                                    PAS
        IF ERR THEN TYPERR(6) ELSE                                         PAS
        BEGIN ERR := LERR;                                                 PAS
           CASE P↑.FORM OF                                                 PAS
  NUMERIC:    IF (I < 0) ∨ (J > WORDLENGTH - 2) THEN                       PAS
              BEGIN ERROR(6);  P1 := NIL END ELSE                          PAS
                 P1 := PNUMPTR;                                            PAS
  SYMBOLIC:   BEGIN IF P = CHARPTR THEN J := WORDLENGTH - 2  ;             PAS
                 IF J > WORDLENGTH - 2 THEN                                PAS
                 BEGIN ERROR(6); P1 := NIL END ELSE P1 := P↑.PWSET;        PAS
              END;                                                         PAS
           END ↱CASE↳;                                                     PAS
           TL := 1;                                                        PAS
        END ↱ERR↳;                                                         PAS
     END ↱POWERSET↳;                                                       PAS
                                                                           PAS
     END ↱STRUCTURED TYPES↳ ELSE                                           PAS
     IF NO = 18 THEN ↱POINTER↳                                             PAS
     BEGIN INSYMBOL;                                                       PAS
        IF NO ≠ 1 THEN TYPERR(11) ELSE                                     PAS
        BEGIN TL := 1;  ALLOC(P,TYPES,POINTER);                            PAS
           WITH P↑ DO                                                      PAS
           BEGIN NAME := BLANK;  NXTEL := NIL;  KLASS := TYPES;            PAS
              FORM := POINTER;  SIZE := 1;                                 PAS
           END;                                                            PAS
           SRCHREC(NEXT);                                                  PAS
           IF CTPTR = NIL THEN SEARCH;                                     PAS
           IF CTPTR ≠ NIL THEN                                             PAS
           IF CTPTR↑.VTYPE = NIL THEN CTPTR := NIL;                        PAS
           IF CTPTR = NIL THEN ↱UNDECLARED CLASS↳                          PAS
           IF PTX > PTLIMIT THEN                                           PAS
           BEGIN ERROR(92); P1 := NIL; INSYMBOL END ELSE                   PAS
           WITH PTLIST[PTX] , P↑ DO                                        PAS
           BEGIN HNAME := AVAL;  PPTR := P;                                PAS
```

LIST OF CONTROL, ACTIVE, AND/OR INACTIVE CARDS IN PASCSYM

000065

```
            DOMAIN := P;  ELTYPE := P;                                          PAS
            PTX := PTX + 1;  P1 := P;  INSYMBOL;                                PAS
         END ELSE                                                               PAS
         WITH CTPTR↑ DO                                                         PAS
         IF (KLASS = VARS) ∧ (VTYPE↑.FORM = CLASSS) THEN                        PAS
         BEGIN P↑.DOMAIN := CTPTR;                                              PAS
            P↑.ELTYPE := VTYPE↑.PELTYPE;                                        PAS
            P1 := P;  INSYMBOL;                                                 PAS
         END ELSE TYPERR(15);                                                   PAS
      END;                                                                      PAS
   END ↱POINTER↳ ELSE TYPERR(18);                                               PAS
END ↱TYPEDECL↳;                                                                 PAS
                                                                                PAS
↱ ---------------------------------------------------------------- ↳            PAS
                                                                                PAS
PROCEDURE BODY(SURRPTR,FIRSTENTRY:CTP);                                         PAS
                                                                                PAS
LABEL 1;                                                                        PAS
VAR I,J : INTEGER;                                                              PAS
   OLDLEV : RG3;                                                                PAS
   LC1,TL,LL,VLC,FSTIX,LCA,LCP : SHRTINT ;                                      PAS
   N,P,PROCPTR,LFIRSTENTRY : CTP;                                               PAS
   TYPID : ALFA ; LCH : CWORD ;                                                 PAS
                                                                                PAS
   PROCEDURE FINDSEMICOLON;                                                     PAS
   BEGIN IF NO ≠ 16 THEN                                                        PAS
      BEGIN ERROR(58);  SKIP(16);                                               PAS
         IF NO ≠ 16 THEN GOTO EXIT 1;                                           PAS
      END;  INSYMBOL;                                                           PAS
   END ↱FINDSEMICOLON↳;                                                         PAS
                                                                                PAS
PROCEDURE VARDECL;                                                              PAS
   LABEL 10;                                                                    PAS
   VAR AT : ADDRESS; I,J : SHRTINT;                                             PAS
                                                                                PAS
   PROCEDURE FILERR;                                                            PAS
   BEGIN ERROR(95);  SKIP(12);  GOTO EXIT 10 END;                               PAS
                                                                                PAS
BEGIN INSYMBOL;                                                                 PAS
   WHILE NO = 1 DO                                                              PAS
   BEGIN I := 0;                                                                PAS
      REPEAT                                                                    PAS
         SRCHREC(NEXT);                                                         PAS
         IF CTPTR ≠ NIL THEN ERROR(8) ELSE                                      PAS
         BEGIN ALLOC(P,VARS);                                                   PAS
            WITH P↑ DO                                                          PAS
            BEGIN NAME := AVAL;  NXTEL := NEXT;  KLASS := VARS;                 PAS
               VKIND := ACTUAL;  VTYPE := NIL;                                  PAS
               VLEVEL := 0 ;                                                    PAS
            END;                                                                PAS
            NEXT := P;  I := I + 1;                                             PAS
         END;                                                                   PAS
```

LIST OF CONTROL, ACTIVE, AND/OR INACTIVE CARDS IN PASCSYM 000366

```
        INSYMBOL;                                                          PAS
        IF NO = 11 THEN ↱ [ IN,OUT,PRINT,PUNCH  ] ↳                        PAS
        BEGIN INSYMBOL;                                                    PAS
           IF (NO = 8) ∧ (CL = 7) THEN ↱IN↳                                PAS
              P↑.VLEVEL := 4  ELSE                                         PAS
           IF NO = 1 THEN                                                  PAS
              IF AVAL = ΞOUTΞ THEN                                         PAS
                P↑.VLEVEL := 1  ELSE                                       PAS
              IF AVAL = ΞPRINTΞ THEN                                       PAS
                P↑.VLEVEL := 2 ELSE                                        PAS
              IF AVAL = ΞPUNCHΞ THEN                                       PAS
                P↑.VLEVEL := 3 ELSE FILERR ELSE FILERR ;                   PAS
           INSYMBOL;  IF NO ≠ 12 THEN FILERR;                              PAS
           INSYMBOL;                                                       PAS
        END;                                                               PAS
        ERR := FALSE;                                                      PAS
        IF NO = 15 THEN                                                    PAS
        BEGIN INSYMBOL;                                                    PAS
           IF NO ≠ 1 THEN                                                  PAS
           BEGIN ERROR(11);  GOTO 10 END;                                  PAS
        END ELSE IF NO ≠ 19 THEN ERROR(10);                                PAS
     UNTIL NO ≠ 1 ;                                                        PAS
     IF NO = 19 THEN INSYMBOL                                              PAS
        ELSE IF ¬ERR THEN ERROR(10) ;                                      PAS
     N := NEXT;  ERR := FALSE;                                             PAS
     TYPEDECL(TL,CTPTR);                                                   PAS
     IF ERR THEN GOTO 10;                                                  PAS
     LC := LC + I*TL;  LL := LC;                                           PAS
     FOR I := I DOWNTO 1 DO                                                PAS
     WITH N↑ DO                                                            PAS
     BEGIN LL := LL - TL;                                                  PAS
        VTYPE := CTPTR;                                                    PAS
        IF CTPTR ≠ NIL THEN                                                PAS
        IF CTPTR↑.FORM = CLASSS THEN                                       PAS
        BEGIN  ↱ CHECK FOR PREDECLARED CLASS ↳                             PAS
           TYPID := NAME;  P := CTPTR↑.PELTYPE;                            PAS
           IF PFTOP = FILLIMIT THEN ERROR(92) ELSE                         PAS
           BEGIN PFTOP := PFTOP + 1;  PFL[PFTOP] := LL END;                PAS
           FOR J := PTX - 1 DOWNTO 0 DO                                    PAS
           WITH PTLIST[J] DO                                               PAS
              IF HNAME = TYPID THEN                                        PAS
              BEGIN PPTR↑.DOMAIN := N;                                     PAS
                 PPTR↑.ELTYPE := P;                                        PAS
                 PTX := PTX - 1;                                           PAS
                 HNAME := PTLIST[PTX].HNAME;                               PAS
                 PPTR := PTLIST[PTX].PPTR;                                 PAS
              END  ↱ WITH, FOR ↳ ;                                         PAS
           VLEVEL := LEVEL;  VADDR := LL;                                  PAS
        END  ↱ CHECK CLASS ↳  ELSE                                         PAS
        IF CTPTR↑.FORM = FILES THEN                                        PAS
           IF FILTOP = FILLIMIT THEN ERROR(92) ELSE                        PAS
           BEGIN FILTOP := FILTOP + 1;  FILPTS[FILTOP] := N;               PAS
```

LIST OF CONTROL, ACTIVE, AND/OR INACTIVE CARDS IN PASCSYM

000067

```
                    VADDR := LL;                                                PAS
                    IF LEVEL > 0 THEN                                           PAS
                    ↱ TAKE EFET INTO ACCOUNT FOR ADDRESS ↓                      PAS
                    BEGIN IF CTPTR↑.FELTYPE = CHARPTR THEN                      PAS
                       J := ALFALENG + 7 ELSE J := 7;                           PAS
                      LC := LC + J;  VADDR := VADDR + I*J - 7;                  PAS
                      VLEVEL := LEVEL ;                                         PAS
                    END;                                                        PAS
                 END ↱FORM = FILES↓ ELSE                                        PAS
              BEGIN VLEVEL := LEVEL;  VADDR := LL END;                          PAS
              N := NXTEL;                                                       PAS
           END  ↱ FOR I ↓ ;                                                     PAS
10:        FINDSEMICOLON;                                                      PAS
        END ↱WHILE NO = 1↓;                                                     PAS
        IF NO = 47 THEN  ↱ VALUE ↓                                              PAS
           BEGIN IF LEVEL ≠ 0 THEN ERROR(22);                                   PAS
              INSYMBOL;                                                         PAS
              WHILE NO = 1 DO                                                   PAS
              BEGIN SRCHREC(NEXT);                                              PAS
                 IF CTPTR = NIL THEN SEARCH;                                    PAS
                 IF CTPTR = NIL THEN                                            PAS
                 BEGIN ERROR(12);  FINDSEMICOLON;  GOTO 20 END;                 PAS
                 IF CTPTR↑.KLASS ≠ VARS THEN                                    PAS
                 BEGIN ERROR(26);  FINDSEMICOLON;  GOTO 20 END;                 PAS
                 WITH CTPTR↑ DO                                                 PAS
                 BEGIN IT := VADDR;                                             PAS
                    IF VLC = 0 THEN                                             PAS
                    BEGIN VLC := IT;  LL := IT;  AT := IT;  END;                PAS
                    IF IT < LL THEN ERROR(27) ELSE                              PAS
                    WHILE IT > AT DO                                            PAS
                    BEGIN PUT(PASCLGO);  AT := AT + 1  END;                     PAS
                    INSYMBOL;                                                   PAS
                    IF (NO ≠ 8) ∨ (CL ≠ 6) THEN ERROR(4) ELSE INSYMBOL;         PAS
                    IF NO = 9 THEN  ↱ LIST ↓                                    PAS
                    BEGIN                                                       PAS
                       REPEAT INSYMBOL;  INCONST(I,PT,NEXT);  IT1 := 1;         PAS
                          IF (NO = 6) ∧ (CL = 1) THEN   ↱  *  ↓                 PAS
                          BEGIN IT1 := I;  INSYMBOL;                            PAS
                             INCONST(I,PT,NEXT);                                PAS
                          END;                                                  PAS
                          IF ¬ ERR THEN                                         PAS
                         BEGIN                                                  PAS
                          FOR J := 1 TO IT1 DO                                  PAS
                          BEGIN PASCLGO↑ := I;  PUT(PASCLGO)  END;              PAS
                          AT := AT + IT1;                                       PAS
                         END ↱¬ERR↓;                                            PAS
                          IF ¬(NO IN [15,10]) THEN  ↱ NOT , NOR ) ↓             PAS
                          BEGIN ERROR(20);  FINDSEMICOLON;  GOTO 20 END;        PAS
                       UNTIL NO = 10 ;  INSYMBOL;                               PAS
                    END ELSE                                                    PAS
                    BEGIN INCONST(I,PT,NEXT);                                   PAS
                       IF ¬ERR THEN                                             PAS
```

LIST OF CONTROL, ACTIVE, AND/OR INACTIVE CARDS IN PASCSYM

000068

```
                    BEGIN PASCLGO↑ := I;  PUT(PASCLGO);                    PAS
                       AT := AT + 1;                                       PAS
                    END;                                                   PAS
                 END;                                                      PAS
                 LL := IT + VTYPE↑.SIZE;                                   PAS
                 IF LL < AT THEN ERROR(20) ;                               PAS
                 FINDSEMICOLON;                                            PAS
              END ↱WITH↳;                                                  PAS
20:        END ↱WHILE NO = 1↳;                                            PAS
           FOR AT := AT + 1 TO LC DO PUT(PASCLGO) ;                        PAS
        END ↱VALUE↳ ELSE VLC := LC;                                        PAS
  END  ↱ VARDECL ↳ ;                                                       PAS
                                                                           PAS
  PROCEDURE FORMPARM;                                                      PAS
     ↱ ( HAS BEEN VERIFIED AND NEXT SYMBOL READ ↳                          PAS
     LABEL 3;                                                              PAS
     VAR SPEC : IDCLASS; REP : BOOLEAN;                                    PAS
                                                                           PAS
     PROCEDURE FORMERR;                                                    PAS
     BEGIN ERROR(11);  SKIP(10);  GOTO EXIT 3 END;                         PAS
                                                                           PAS
  BEGIN SPEC := KONST ;                                                    PAS
1:   REPEAT                                                                PAS
        IF NO = 45 THEN ↱PROCEDURE↳                                        PAS
        REPEAT INSYMBOL;                                                   PAS
           IF NO ≠ 1 THEN ERROR(11) ELSE                                   PAS
           BEGIN SRCHREC(NEXT);                                            PAS
              IF CTPTR ≠ NIL THEN ERROR(8) ELSE                            PAS
              BEGIN ALLOC(P,PROC);                                         PAS
                 WITH P↑ DO                                                PAS
                 BEGIN NAME := AVAL;  NXTEL := NEXT;  KLASS := PROC;       PAS
                   PROCTYPE := P;  PROCKIND := FORMAL;                     PAS
                   PROCADDR := LC;  PROCLEVEL := LEVEL;                    PAS
                   FORMALS := NIL;                                         PAS
                 END;  LC := LC + 1;  NEXT := P;                           PAS
              END ↱CTPTR = NIL↳;                                           PAS
              INSYMBOL;                                                    PAS
           END;                                                            PAS
        UNTIL NO ≠ 15                                                      PAS
        ELSE ↱NOT PROCEDURE↳                                               PAS
        BEGIN                                                              PAS
           IF NO ≠ 1 THEN                                                  PAS
           BEGIN IF NO = 43 THEN SPEC := VARS ELSE                         PAS
                 IF NO = 41 THEN SPEC := KONST ELSE                        PAS
                 IF NO = 44 THEN SPEC := PROC ELSE                         PAS
                   FORMERR;                                                PAS
              INSYMBOL;                                                    PAS
           END ELSE IF SPEC ≠ VARS THEN SPEC := KONST ;                    PAS
           IF NO ≠ 1 THEN FORMERR;                                         PAS
           REPEAT                                                          PAS
              SRCHREC(NEXT);                                               PAS
              IF CTPTR ≠ NIL THEN ERROR(8) ELSE                            PAS
```

```
D OLDPL                MASTER AUDIT, IDENT CARD TOTAL          UPDATE V1.2        10/

 LIST OF CONTROL, ACTIVE, AND/OR INACTIVE CARDS IN PASCSYM

                                                                      000069
                BEGIN                                                          PAS
                   CASE SPEC OF                                                PAS
   KONST:          BEGIN ALLOC(P,KONST,FORMAL);                                PAS
                     WITH P↑ DO                                                PAS
                     BEGIN CONTYPE := NIL;  CONKIND := FORMAL;                 PAS
                       CADDR := LC;  CLEVEL := LEVEL;                          PAS
                     END;                                                      PAS
                   END;                                                        PAS
   PROC:           BEGIN ALLOC(P,PROC);                                        PAS
                     WITH P↑ DO                                                PAS
                     BEGIN PROCTYPE := NIL;  PROCKIND := FORMAL;               PAS
                       PROCADDR := LC;  PROCLEVEL := LEVEL;                    PAS
                       FORMALS := NIL;                                         PAS
                     END;                                                      PAS
                   END;                                                        PAS
   VARS:           BEGIN ALLOC(P,VARS);                                        PAS
                     WITH P↑ DO                                                PAS
                     BEGIN VTYPE := NIL;  VKIND := FORMAL;                     PAS
                       VADDR := LC;  VLEVEL := LEVEL;                          PAS
                     END;                                                      PAS
                   END;                                                        PAS
                   END ↱CASE↳;                                                 PAS
                   P↑.NAME := AVAL;  P↑.NXTEL := NEXT;                         PAS
                   P↑.KLASS := SPEC;                                           PAS
                   LC := LC + 1;  NEXT := P;                                   PAS
                END ↱CTPTR = NIL↳;                                             PAS
                INSYMBOL;                                                      PAS
                IF NO = 15 THEN                                                PAS
                BEGIN INSYMBOL;                                                PAS
                   IF NO = 19 THEN ERROR(11);                                  PAS
                END ELSE IF NO ≠ 19 THEN GOTO 4;                               PAS
             UNTIL NO ≠ 1;                                                     PAS
 4:          IF NO ≠ 19 THEN ERROR(10) ELSE INSYMBOL;                          PAS
             IF NO ≠ 1 THEN FORMERR;                                           PAS
             SEARCH;                                                           PAS
             IF CTPTR = NIL THEN                                               PAS
             BEGIN ERROR(12);  GOTO 2 END;                                     PAS
             IF CTPTR↑.KLASS ≠ TYPES THEN                                      PAS
             BEGIN ERROR(18) ; GOTO 2 END ;                                    PAS
             N := NEXT;                                                        PAS
             REPEAT                                                            PAS
                CASE SPEC OF                                                   PAS
   KONST:       IF N↑.CONTYPE = NIL THEN                                       PAS
                BEGIN IF ¬(CTPTR↑.FORM IN                                      PAS
                      [NUMERIC,SYMBOLIC,POWER]) THEN                           PAS
                   BEGIN N↑.KLASS := VARS;                                     PAS
                     IF CTPTR↑.FORM = POINTER THEN                             PAS
                       N↑.VKIND := ACTUAL ;                                    PAS
                   END;                                                        PAS
                   N↑.CONTYPE := CTPTR;  N := N↑.NXTEL;                        PAS
                END ELSE N := NIL;                                             PAS
   PROC:        IF N↑.PROCTYPE = NIL THEN                                      PAS
```

LIST OF CONTROL, ACTIVE, AND/OR INACTIVE CARDS IN PASCSYM

000070

```
                    IF CTPTR↑.FORM > POWER THEN                                  PAS
                    BEGIN ERROR(93);  N := NIL END ELSE                          PAS
                    BEGIN N↑.PROCTYPE := CTPTR;  N := N↑.NXTEL END               PAS
                ELSE N := NIL;                                                   PAS
  VARS:         IF N↑.VTYPE = NIL THEN                                           PAS
                BEGIN N↑.VTYPE := CTPTR;  N := N↑.NXTEL;                         PAS
                END ELSE N := NIL;                                               PAS
                END ↱CASE↳;                                                      PAS
            UNTIL N = NIL;                                                       PAS
2:          INSYMBOL;                                                          PAS
        END ↱NOT PROCEDURE↳;                                                     PAS
3:      REP := NO IN [1,41,43,44,45];                                          PAS
        IF NO = 16 THEN                                                          PAS
        BEGIN INSYMBOL;  REP := NO ≠ 10  END;                                    PAS
      UNTIL ¬REP;                                                                PAS
      IF NO ≠ 10 THEN                                                            PAS
      BEGIN ERROR(9);  SKIP(10);                                                 PAS
        IF NO IN [41,43,44,45] THEN GOTO 1;                                      PAS
      END;                                                                       PAS
      ↱REVERSE POINTERS↳                                                         PAS
      N := NEXT;  NEXT := NIL;                                                   PAS
      WHILE N ≠ NIL DO                                                           PAS
      BEGIN P := N;  N := P↑.NXTEL;                                              PAS
        P↑.NXTEL := NEXT;  NEXT := P;                                            PAS
      END;                                                                       PAS
  END ↱FORMPARM↳;                                                                PAS
                                                                                 PAS
  PROCEDURE OUTPMD(SURRPTR : CTP);                                               PAS
    CONST DIRECT = 0, INDIRECT = 1;                                              PAS
    VAR N,P : CTP;  ADR,ACC : SHRTINT;  OUT : BOOLEAN;                           PAS
  BEGIN                                                                          PAS
    WITH PMDFILE↑ DO                                                             PAS
    BEGIN IF SURRPTR = NIL THEN                                                  PAS
      BEGIN PNAME := Ξ(MAIN)Ξ;  SA := IC END ELSE                                PAS
      BEGIN PNAME := SURRPTR↑.NAME; SA := SURRPTR↑.PROCADDR END;                 PAS
      PKIND := PROCDR;  LINK := LASTLINK;  COUNT := 0;                           PAS
    END;                                                                         PAS
    PUT(PMDFILE);  PMCTR := PMCTR + 2;  LASTLINK := PMCTR;                       PAS
    N := NEXT;                                                                   PAS
    WHILE N ≠ NIL DO                                                             PAS
    WITH N↑ DO                                                                   PAS
    BEGIN OUT := FALSE;  ACC := DIRECT;                                          PAS
      CASE KLASS OF                                                              PAS
KONST:    IF CONKIND = FORMAL THEN                                               PAS
          BEGIN OUT := TRUE;  P := CONTYPE;  ADR := CADDR END;                   PAS
VARS:     BEGIN OUT := TRUE;  P := VTYPE;  ADR := VADDR;                         PAS
            IF VKIND = FORMAL THEN ACC := INDIRECT;                              PAS
          END;                                                                   PAS
TYPES,PROC:                                                                      PAS
      END;                                                                       PAS
      IF OUT THEN IF P↑.FORM < POWER THEN                                        PAS
      BEGIN WITH PMDFILE↑ DO                                                     PAS
```

```
            BEGIN PNAME := NAME;  PKIND := OBJ;                                  PAS
               IF P↑.FORM = NUMERIC THEN TYP := 1 ELSE                           PAS
               IF P = REALPTR THEN TYP := 2 ELSE                                 PAS
               IF P = ALFAPTR THEN TYP := 3 ELSE                                 PAS
               IF P = CHARPTR THEN TYP := 4 ELSE                                 PAS
               IF P = BOOLPTR THEN TYP := 5 ELSE                                 PAS
               IF P↑.FORM = SYMBOLIC THEN TYP := 1 ELSE TYP := 6 ;               PAS
               RLA := ADR;  ACCESS := ACC;                                       PAS
            END;                                                                 PAS
            PUT(PMDFILE);  PMCTR := PMCTR + 2;                                   PAS
         END;                                                                    PAS
         N := NXTEL;                                                             PAS
      END ↱WITH, WHILE↳;                                                         PAS
   END ↱OUTPMD↳;                                                                 PAS
                                                                                 PAS
   PROCEDURE ENTERBODY ;                                                         PAS
    ↱GENERATES PROCEDURE PROLOG CODE AND PROCEDURE ENTRY CODE, CODE              PAS
     TO INITIALIZE EFETS AND TO OPEN LOCAL FILES, AND CODE TO INITIALIZE         PAS
     LOCAL CLASSPOINTERS↳                                                        PAS
      CONST OPN = 104B ;                                                         PAS
      VAR IT2 : SHRTINT;                                                         PAS
   BEGIN CA := 0 ; CP := 4 ; LCX := 0 ;                                          PAS
      IF LEVEL ≠ 0 THEN                                                          PAS
      BEGIN ↱PROCEDURE ENTRY CODE↳                                               PAS
         GEN15(76B,0,5,0) ; GEN15(20B,0,0,18) ;                                  PAS
         GEN15(12B,7,7,0) ; GEN15(66B,5,6,0) ;                                   PAS
         GEN30(51B,7,5,1) ; GEN30(61B,6,6,0) ;                                   PAS
         LCA := CA ; LCP := CP ;                                                 PAS
         IF LEVEL ≠ 1 THEN GEN15(56B,6,5,0) ;                                    PAS
         ↱CODE TO INITIALIZE EFETS↳                                              PAS
         FOR IT := FILEV[LEVEL] TO FILTOP DO                                     PAS
         WITH FILPTS[IT]↑ DO                                                     PAS
         BEGIN                                                                   PAS
            UNPACK(NAME,LCH,1) ; IT2 := 7 ;                                      PAS
            WHILE LCH[IT2] = ≡ DO IT2 := IT2 - 1 ;                               PAS
            RP := 0;  LDCST(INT(NAME));                                          PAS
            GEN15(43B,0,0,6*IT2);  GEN15(11B,6,0,1);                            PAS
            GEN30(51B,6,5,VADDR+2);                                              PAS
            GEN30(71B,6,0,VTYPE↑.FELTYPE↑.SIZE);                                 PAS
            IF VTYPE↑.FELTYPE ≠ CHARPTR THEN                                     PAS
            BEGIN GEN15(43B,0,0,1);                                              PAS
               GEN15(20B,0,0,56);  GEN15(12B,6,6,0);                             PAS
            END;                                                                 PAS
            GEN30(51B,6,5,VADDR+1);                                              PAS
            GEN30(71B,6,5,VADDR+7);                                              PAS
            GEN30(51B,6,5,VADDR+3);                                              PAS
            GEN30(72B,6,6,VTYPE↑.SIZE);                                          PAS
            GEN30(51B,6,5,VADDR+6);                                              PAS
         END;                                                                    PAS
      END ELSE                                                                   PAS
      BEGIN                                                                      PAS
         ↱MAIN PROGRAM, SAVE X7,B5 SET B6↳                                       PAS
```

LIST OF CONTROL, ACTIVE, AND/OR INACTIVE CARDS IN PASCSYM 033072

```
      GEN15(56B,7,5,0) ; GEN15(76B,7,5,0) ;                                    PAS
      GEN30(51B,7,0,PARMLIST.IC0) ; GEN30(61B,6,5,0) ;                         PAS
      LCA := CA ; LCP := CP ; LC := 1 ;                                        PAS
   END ;                                                                       PAS
   IF STOFLCHECK THEN GEN30(06B,6,4,STOFLERR) ;                                PAS
   ↑CODE TO OPEN FILES↓                                                        PAS
   IF LEVEL = 0 THEN IT2 := 0 ELSE IT2 := 5 ;                                  PAS
   FOR IT := FILEV[LEVEL] TO FILTOP DO                                         PAS
   BEGIN GEN30(61B,7,IT2,FILPTS[IT]↑.VADDR+1);                                 PAS
      JUMPTO(OPN) ;                                                            PAS
   END;                                                                        PAS
   ↑CODE TO INITIALIZE CLASSPOINTERS↓                                          PAS
   FOR IT := PILEV[LEVEL] TO PFTOP DO                                          PAS
   BEGIN IT1 := PFL[IT] ;                                                      PAS
      GEN30(71B,7,IT2,IT1+1) ; GEN30(51B,7,IT2,IT1) ;                          PAS
   END ;                                                                       PAS
   PFTOP := PILEV[LEVEL] - 1 ; TCT := 0 ; TMAX := 0 ; CLABIX := 0 ;            PAS
END ↑ENTERBODY↓ ;                                                              PAS
                                                                               PAS
PROCEDURE LEAVEBODY ;                                                          PAS
 ↑CALLED AT PROCEDURE END. GOES THROUGH LABELTABLE LABTAB AND                  PAS
  EXITTABLE EXTAB, UPDATING CODE, AND GENERATES PROCEDURE EXIT CODE↓            PAS
   CONST CLS = 105B ;                                                          PAS
   VAR IT2,IT3 : INTEGER;                                                      PAS
   PROCEDURE ERRMESSAGE(ALF:ALFA; VAL:SHRTINT) ;                               PAS
   BEGIN ERROR(43) ; WRITE(EOL) ; PRTERR ;                                     PAS
      WRITE(Ξ Ξ,ALF:6,VAL:6,EOL) ;                                             PAS
      IF ¬EOLFLAG THEN WRITE(Ξ Ξ:CHCNT+8)                                      PAS
   END ↑ERRMESSAGE↓ ;                                                          PAS
BEGIN IF LEVEL = 0 THEN IT2 := 0 ELSE IT2 := 5 ;                               PAS
   FOR IT := FILEV[LEVEL] TO FILTOP DO                                         PAS
   BEGIN GEN30(61B,7,IT2,FILPTS[IT]↑.VADDR + 1) ; JUMPTO(CLS) ;                PAS
   END ;                                                                       PAS
   FILTOP := FILEV[LEVEL] - 1 ;                                                PAS
   IF LEVEL ≠ 0 THEN ↑PROCEDURE EXIT CODE↓                                     PAS
   BEGIN GEN30(51B,1,5,1) ; GEN15(66B,6,5,0) ;                                 PAS
      GEN15(63B,7,1,0) ; GEN15(20B,1,0,42) ;                                   PAS
      GEN15(63B,5,1,0) ;                                                       PAS
   END ELSE ↑MAIN, RETURN TO MONITOR↓                                          PAS
   BEGIN GEN15(56B,1,5,0) ; GEN15(63B,7,1,0) END ;                             PAS
   GEN30(02B,7,0,0) ; LC := LC + TMAX ; INS(LC,LCP,LCA) ;                      PAS
   FOR IT := 1 TO CLABIX DO                                                    PAS
   WITH LABTAB[IT] DO                                                          PAS
   BEGIN IF FLD2 > 0 THEN ↑UNDECLARED  LABEL↓                                  PAS
      BEGIN                                                                    PAS
         ERRMESSAGE(ΞLABEL:Ξ,LABVAL) ;                                         PAS
         UNDLAB[FLD2].SUCC := CHNIX ; CHNIX := FLD3 ;                          PAS
      END ↑IF FLD2 > 0↓ ;                                                      PAS
   END ↑WITH, FOR↓ ;                                                           PAS
   FOR IT := FSTIX TO CEXTABIX DO                                              PAS
   WITH EXTAB[IT] DO                                                           PAS
   BEGIN FOR IT1 := 1 TO CLABIX DO                                             PAS
```

LIST OF CONTROL, ACTIVE, AND/OR INACTIVE CARDS IN PASCSYM

000373

```
        WITH LABTAB[IT1] DO                                                    PAS
        IF LABVAL = EXVAL THEN                                                 PAS
        BEGIN IF FLD2 > 0 THEN ERRMESSAGE(=EXIT:=,EXVAL) ELSE                  PAS
           BEGIN IT2 := JMPCST ; INSERT(LC,30,IT2) ;                           PAS
              INSERT(FLD3,0,IT2) ; JMPTAB[JMPTABIX] := IT2                     PAS
           END ;                                                               PAS
           GOTO 2                                                              PAS
        END ↱IF, WITH, FOR↳ ;                                                  PAS
        ERRMESSAGE(=EXIT:=,EXVAL) ;                                            PAS
2:    END ↱WITH, FOR↳ ;                                                        PAS
      CEXTABIX := FSTIX - 1 ;                                                  PAS
      WRITOUT ;                                                                PAS
      IF LEVEL = 0 THEN  ↱ WRITE OUT JUMPTABLE ↳                               PAS
      FOR IT := 0 TO JMPMAX DO                                                 PAS
      BEGIN PASCLGO↑ := JMPTAB[IT] ; PUT(PASCLGO) END                          PAS
   END ↱LEAVEBODY↳ ;                                                           PAS
                                                                               PAS
   BEGIN  ↱BODY↳                                                               PAS
      PILEV[LEVEL]:= PFTOP+1;  VLC := 0;                                       PAS
      FILEV[LEVEL] := FILTOP + 1;                                              PAS
      FSTIX := CEXTABIX + 1 ;                                                  PAS
1:    IF NO = 40 THEN ↱LABEL↳                                                  PAS
      BEGIN REPEAT INSYMBOL ;                                                  PAS
            IF (NO = 2)∧(CL = 1) THEN                                          PAS
            BEGIN IF IVAL ≥ TWOTO17 THEN                                       PAS
               BEGIN ERROR(100) ; GOTO 2 END ;                                 PAS
               FOR IT := FSTIX TO CEXTABIX DO                                  PAS
               IF EXTAB[IT].EXVAL = IVAL THEN                                  PAS
               BEGIN ERROR(77) ; GOTO 2 END ;                                  PAS
               IF CEXTABIX = MAXEXLABS THEN ERROR(78) ELSE                     PAS
               BEGIN CEXTABIX := CEXTABIX + 1 ;                                PAS
                  IF JMPIX > JMPMAX THEN ERROR(91) ELSE                        PAS
                  BEGIN WITH EXTAB[CEXTABIX] DO                                PAS
                     BEGIN EXVAL := IVAL ; JMPTABIX := JMPIX END ;             PAS
                     JMPTAB[JMPIX] := LEVEL ; JMPIX := JMPIX + 1 ;             PAS
                  END                                                          PAS
               END ;                                                           PAS
2:             INSYMBOL                                                        PAS
            END ↱IF (NO=2)∧(CL=1)↳ ELSE                                        PAS
            BEGIN ERROR(61) ; GOTO 3 END                                       PAS
         UNTIL NO ≠ 15 ↱,↳ ;                                                   PAS
3:       FINDSEMICOLON;                                                        PAS
      END ;                                                                    PAS
      IF NO = 41 THEN ↱ CONST ↳                                                PAS
      BEGIN INSYMBOL ;                                                         PAS
            WHILE NO = 1 DO                                                    PAS
            BEGIN REPEAT SRCHREC(NEXT) ; IF CIPTR ≠ NIL THEN ERROR(8);         PAS
               ALLOC(P,KONST,ACTUAL);                                          PAS
               WITH P↑ DO .                                                    PAS
               BEGIN NAME := AVAL;  NXTEL := NEXT;  KLASS := KONST;            PAS
                 CONKIND := ACTUAL;                                            PAS
               END;  NEXT := P;                                                PAS
```

LIST OF CONTROL, ACTIVE, AND/OR INACTIVE CARDS IN PASCSYM 000074

```
            INSYMBOL;                                                          PAS
            IF (NO ≠ 8) ∨ (CL ≠ 6) THEN ERROR(4) ELSE INSYMBOL;                PAS
            IF NO = 36 THEN ↱ NIL ↳                                            PAS
            BEGIN P↑.CONTYPE := NILPTR;                                        PAS
               P↑.VALUES := NILVAL;  INSYMBOL;                                 PAS
            END ELSE                                                           PAS
            BEGIN INCONST(I,N,NEXT↑.NXTEL);                                    PAS
               P↑.CONTYPE := N;  P↑.VALUES := I;                               PAS
            END;                                                               PAS
               WHILE NO = 15 DO                                                PAS
               BEGIN INSYMBOL;                                                 PAS
                  IF NO ≠ 1 THEN                                               PAS
                  BEGIN ERROR(11); SKIP(15) END;                               PAS
               END;                                                            PAS
            UNTIL NO ≠ 1;                                                      PAS
            FINDSEMICOLON;                                                     PAS
         END ↱ WHILE NO = 1 ↳ ;                                                PAS
   END ↱CONST↳;                                                                PAS
   IF NO =37 THEN  ↱TYPE↳                                                      PAS
   BEGIN INSYMBOL;                                                             PAS
   WHILE NO = 1 DO                                                             PAS
   BEGIN SRCHREC(NEXT); IF CTPTR ≠ NIL THEN ERROR(8);                          PAS
      TYPID := AVAL;  INSYMBOL;                                                PAS
      IF (NO ≠ 8) ∨ (CL ≠ 6) THEN ERROR(4) ELSE INSYMBOL;                      PAS
      ERR := FALSE;  TYPEDECL(TL,P);                                           PAS
      IF ¬ERR THEN                                                             PAS
      IF P↑.NAME ≠ BLANK THEN ERROR(96) ELSE                                   PAS
      BEGIN P↑.NAME := TYPID;  P↑.NXTEL := NEXT;                               PAS
         NEXT := P;                                                            PAS
      END;                                                                     PAS
      FINDSEMICOLON;                                                           PAS
   END;                                                                        PAS
   END  ↱TYPE↳ ;                                                               PAS
   IF NO=43 THEN  ↱VAR↳                                                        PAS
      VARDECL ELSE VLC := LC;                                                  PAS
   IF PTX > 0 THEN                                                             PAS
   BEGIN ERROR(12); WRITE(EOL); PRTERR;                                        PAS
      FOR PTX := PTX-1 DOWNTO 0 DO                                             PAS
         WRITE(Ξ Ξ,ΞCLASS-IDΞ,PTLIST[PTX].HNAME,EOL) ;                         PAS
      IF ¬EOLFLAG THEN WRITE(Ξ Ξ:CHCNT+8);                                     PAS
   END;                                                                        PAS
   IF LEVEL = 0 THEN                                                           PAS
   BEGIN                                                                       PAS
      FOR I := 0 TO FILTOP DO                                                  PAS
      WITH FILPTS[I]↑ DO                                                       PAS
      BEGIN                                                                    PAS
         IF VTYPE↑.FELTYPE = CHARPTR THEN                                      PAS
         BEGIN IT := 0;  LC := LC + ALFALENG;                                  PAS
            FOR J := 1 TO ALFALENG DO PUT(PASCLGO);  ↱CHAR BUFFER↳             PAS
         END ELSE IT := 1;                                                     PAS
         PUT(PASCLGO);   ↱ P-PTR ↳                                             PAS
         J := 2*VLEVEL + IT;  ↱SPECIFIER↳  VLEVEL := 0;                        PAS
```

LIST OF CONTROL, ACTIVE, AND/OR INACTIVE CARDS IN PASCSYM

000075

```
            UNPACK(NAME,LCH,1) ; IT1 := 7 ;                                    PAS
            WHILE LCH[IT1] = ≡ ≡ DO IT1 := IT1 - 1 ;                           PAS
            IT1 := (ALFALENG - IT1)*6 ;                                        PAS
            APPEND(J,55,VTYPE↑.FELTYPE↑.SIZE);                                 PAS
            PASCLGO↑ := J;  PUT(PASCLGO);  ↱ SPECIFIER + LRL ↳                 PAS
            J := INT(NAME);  APPEND(J,-IT1,0);  APPEND(J,IT1,0);               PAS
            PASCLGO↑ := J;  PUT(PASCLGO);  ↱ LFN ↳                             PAS
            PASCLGO↑ := VADDR;                                                 PAS
            FOR J := 1 TO 3 DO PUT(PASCLGO);  ↱FIRST,IN,OUT↳                   PAS
            PASCLGO↑ := VADDR + VTYPE↑.SIZE;  PUT(PASCLGO);   ↱ LIMIT ↳        PAS
            VADDR := LC;  LC := LC + 7;                                        PAS
         END;                                                                  PAS
         PARMLIST.LPJMPTAB := LC ;                                             PAS
         IF VLC ≠ LC THEN FOR J := 0 TO JMPMAX DO PUT(PASCLGO)                 PAS
                     ELSE VLC := LC + (JMPMAX + 1) ;                           PAS
         LC := LC + (JMPMAX + 1);                                              PAS
         IC := LC;  PARMLIST.LOADPT := VLC;  ↱LOADPT↳                          PAS
      END;                                                                     PAS
      DP := FALSE;                                                             PAS
      IF NO IN [44,45] THEN ↱FUNCTION OR PROCEDURE↳                            PAS
      BEGIN IF SURRPTR ≠ NIL THEN                                              PAS
         IF SURRPTR↑.PROCADDR = 0 THEN                                         PAS
         BEGIN IF JMPIX > JMPMAX THEN ERROR(91) ELSE                           PAS
            BEGIN SURRPTR↑.PROCADDR := PARMLIST.LPJMPTAB + JMPIX ;             PAS
               JMPIX := JMPIX + 1 ;                                            PAS
            END                                                                PAS
         END;                                                                  PAS
         REPEAT                                                                PAS
            LL := NO; ERR := FALSE;                                            PAS
            OLDLEV := LEVEL;                                                   PAS
            IF LEVEL < MAXLEVEL THEN LEVEL := LEVEL + 1 ELSE ERROR(76);        PAS
            INSYMBOL;                                                          PAS
            IF NO ≠ 1 THEN                                                     PAS
            BEGIN ERROR(11);  LEVEL := OLDLEV;  GOTO 1 END;                    PAS
            LC1 := LC;  SRCHREC(NEXT);                                         PAS
            IF CTPTR ≠ NIL THEN                                                PAS
            IF CTPTR↑.KLASS ≠ PROC THEN                                        PAS
            BEGIN ERROR(8); CTPTR := NIL END;                                  PAS
            IF CTPTR = NIL THEN  ↱ UNDECLARED PROCEDURE ↳                      PAS
            BEGIN ALLOC(PROCPTR,PROC);                                         PAS
               WITH PROCPTR↑ DO                                                PAS
               BEGIN NAME := AVAL;  NXTEL := NEXT;  KLASS := PROC;             PAS
                  PROCTYPE := PROCPTR;  PROCKIND := ACTUAL;                    PAS
                  PROCLEVEL := LEVEL - 1;                                      PAS
               END;                                                            PAS
               DISPLAY[TOP].FNAME := PROCPTR;                                  PAS
               NEXT := NIL ;                                                   PAS
               INSYMBOL;                                                       PAS
               IF LL = 44 THEN  ↱ FUNCTION ↳                                   PAS
               BEGIN IF NO ≠ 9 THEN ERROR(29) ELSE                             PAS
                  BEGIN LC := 3 ; INSYMBOL ; FORMPARM END ;                    PAS
                  IF ERR THEN PROCPTR↑.PROCTYPE := NIL;                        PAS
```

LIST OF CONTROL, ACTIVE, AND/OR INACTIVE CARDS IN PASCSYM 000076

```
            IF NO = 10 THEN                                                 PAS
            BEGIN INSYMBOL;                                                 PAS
              IF NO ≠ 19 THEN ERROR(10) ELSE INSYMBOL;                      PAS
              IF NO ≠ 1 THEN                                                PAS
              BEGIN ERROR(11);  SKIP(49);                                   PAS
                PROCPTR↑.PROCTYPE := NIL;                                   PAS
              END ELSE                                                      PAS
              BEGIN SEARCH;                                                 PAS
            IF CTPTR ≠ NIL THEN                                             PAS
            BEGIN IF CTPTR↑.KLASS ≠ TYPES THEN                              PAS
               BEGIN ERROR(93); CTPTR := NIL END ELSE                       PAS
               IF CTPTR↑.FORM > POWER THEN                                  PAS
               BEGIN ERROR(93) ; CTPTR := NIL END ;                         PAS
            END ELSE ERROR(12) ;                                            PAS
            PROCPTR↑.PROCTYPE := CTPTR ;                                    PAS
            INSYMBOL;                                                       PAS
              END;                                                          PAS
            END ↱NO = 10↳;                                                  PAS
          END  ↱ FUNCTION ↳  ELSE  ↱ PROCEDURE ↳                            PAS
          BEGIN LC := 3 ;                                                   PAS
            IF NO = 9 THEN  ↱ PARMLIST ↳                                    PAS
            BEGIN INSYMBOL; FORMPARM;                                       PAS
               IF ERR THEN PROCPTR↑.PROCTYPE := NIL;                        PAS
               IF NO = 10 THEN INSYMBOL;                                    PAS
            END ↱NO = 9↳;                                                   PAS
          END ↱PROCEDURE↳;                                                  PAS
          IF NO ≠ 16 THEN                                                   PAS
          BEGIN PROCPTR↑.PROCTYPE := NIL;                                   PAS
            ERROR(58);  SKIP(16);                                           PAS
          END ELSE INSYMBOL;                                                PAS
          PROCPTR↑.FORMALS := NEXT;                                         PAS
          IF (NO=1) ∧ (AVAL=ΞFORWARDΞ) THEN                                 PAS
          BEGIN NEXT := PROCPTR;                                            PAS
            IF JMPIX > JMPMAX THEN ERROR(91) ELSE                           PAS
               BEGIN PROCPTR↑.PROCADDR :=                                   PAS
                  PARMLIST.LPJMPTAB + JMPIX ;                               PAS
                  JMPIX := JMPIX + 1 ;                                      PAS
               END ;                                                        PAS
            PROCPTR↑.SEGSIZE := -LC;                                        PAS
             ↱ SEGSIZE < 0 SIGNIFIES FORWARD-DECLARATION ↳                  PAS
            INSYMBOL;                                                       PAS
          END ELSE                                                          PAS
          BEGIN TOP := LEVEL + 1;                                           PAS
            WITH DISPLAY[TOP] DO                                            PAS
            BEGIN FNAME := NEXT;  OCCUR := BLOCK  END;                      PAS
            PROCPTR↑.PROCADDR := 0;                                         PAS
            ALLOC(LFIRSTENTRY,DUMMYCLASS) ;                                 PAS
            BODY(PROCPTR,LFIRSTENTRY);                                      PAS
          END;                                                              PAS
        END  ↱ NEW PROC ↳  ELSE  ↱ PROC ID ALREADY DECLARED ↳               PAS
        BEGIN                                                               PAS
          IF CTPTR↑.SEGSIZE ≥ 0 THEN  ↱ PREV. DECL NOT FORWARD ↳            PAS
```

000077

```
                ERROR(16);                                                    PAS
              INSYMBOL;                                                       PAS
              IF NO = 9 THEN  ↱ IGNORE PARM-LIST ↳                            PAS
              BEGIN ERROR(23);                                                PAS
                 REPEAT SKIP(10);                                             PAS
                    IF NO IN [16,41,43,44,45] THEN INSYMBOL;                  PAS
                 UNTIL ¬(NO IN [16,41,43,44,45]);                             PAS
                 IF NO ≠ 10 THEN ERROR(9) ELSE INSYMBOL;                      PAS
              END;                                                            PAS
              IF NO = 15 THEN SKIP(16);                                       PAS
              IF NO ≠ 16 THEN ERROR(58) ELSE INSYMBOL;                        PAS
              IF (NO=1) ∧ (AVAL=≡FORWARD≡) THEN ↱AGAIN FORWARD↳               PAS
                 INSYMBOL ELSE                                                PAS
             BEGIN                                                            PAS
              PROCPTR := CTPTR;                                               PAS
              WITH PROCPTR↑ DO                                                PAS
              BEGIN LC := -SEGSIZE;  NEXT := FORMALS  END;                    PAS
              TOP := LEVEL + 1;                                               PAS
              WITH DISPLAY[TOP] DO                                            PAS
              BEGIN FNAME := NEXT;  OCCUR := BLOCK  END;                      PAS
              ALLOC(LFIRSTENTRY,DUMMYCLASS);                                  PAS
              BODY(PROCPTR,LFIRSTENTRY);                                      PAS
             END ↱NOT FORWARD↳;                                               PAS
          END  ↱ OLD PROCEDURE ↳ ;                                            PAS
          LC := LC1;                                                          PAS
          LEVEL := OLDLEV;                                                    PAS
          FINDSEMICOLON;                                                      PAS
       UNTIL ¬(NO IN [44,45]) ;                                               PAS
    END  ↱ FUNCTION OR PROCEDURE ↳ ;                                          PAS
    DISPLAY[TOP].FNAME := NEXT;                                               PAS
    IF NO≠21 THEN                                                             PAS
    BEGIN ERROR(24); SKIP(49);                                                PAS
       WHILE NO IN [16,22] DO                                                 PAS
       BEGIN INSYMBOL;                                                        PAS
          SKIP(49);                                                           PAS
       END ;                                                                  PAS
       IF NO IN [37,40,41,43,44,45] THEN GOTO 1;                              PAS
    END ELSE                                                                  PAS
    IF SURRPTR = NIL THEN PARMLIST.ENTRYPT := IC ELSE                         PAS
    WITH SURRPTR↑ DO                                                          PAS
    BEGIN IF PROCADDR ≠ 0 THEN                                                PAS
       BEGIN I := IC ; APPEND(I,30,EQJMP) ;                                   PAS
          JMPTAB[PROCADDR - PARMLIST.LPJMPTAB] := I                           PAS
       END ;                                                                  PAS
       PROCADDR := IC                                                         PAS
    END ;                                                                     PAS
    N := NEXT ; ERR := FALSE ;                                                PAS
    WHILE N ≠ NIL DO                                                          PAS
       IF N↑.KLASS ≠ PROC THEN N := NIL ELSE                                  PAS
       BEGIN IF N↑.SEGSIZE < 0 THEN ↱FORWARD, NO REAL DECLARATION↳            PAS
          BEGIN IF ¬ERR THEN                                                  PAS
             BEGIN ERROR(12) ; WRITE(EOL) ; PRTERR END ;                      PAS
```

LIST OF CONTROL, ACTIVE, AND/OR INACTIVE CARDS IN PASCSYM 000073

```
            WRITE(Ξ PROCEDUREΞ,Ξ Ξ:2,N↑.NAME,EOL) ;                    PA
          END ;                                                        PA
          N := N↑.NXTEL ;                                              PA
       END ↱IF, WHILE↳ ;                                               PA
    IF ERR THEN IF ¬EOLFLAG THEN WRITE(Ξ Ξ:CHCNT+8) ;                  PA
    IF PARMLIST.ERRFLAG = FALSE THEN OUTPMD(SURRPTR) ;                 PA
    ENTERBODY;                                                         PA
    COMPSTAT ;                                                         PA
    LEAVEBODY ;                                                        PA
    IF SURRPTR ≠ NIL THEN SURRPTR↑.SEGSIZE := LC;                      PA
    ALLOC(P,DUMMYCLASS);                                               PA
    IF P > MAXCTP THEN MAXCTP := P;                                    PA
    RESET(FIRSTENTRY);                                                 PA
    TOP := LEVEL;  NEXT := DISPLAY[TOP].FNAME;                         PA
  END ↱ BODY ↳ ;                                                       PA
                                                                       PA
↱ ---------------------------------------------------------------- ↳   PA
↱ M A I N   P R O G R A M ↳                                            PA
                                                                       PA
BEGIN                                                                  PA
                                                                       PA
  ↱ INITIALIZE CONTEXT-TABLE ↳                                         PA
  ALLOC(NILPTR,TYPES,POINTER);   ↱TYPE OF NIL↳                         PA
  WITH NILPTR↑ DO                                                      PA
  BEGIN NAME := BLANK;  NXTEL := NIL;  KLASS := TYPES;                 PA
    SIZE := 0;  FORM := POINTER;  DOMAIN := NIL;  ELTYPE := NIL;       PA
  END;                                                                 PA
  NEXT := NIL ;                                                        PA
  FOR IT := 0 TO 11 DO     ↱TEXT, WEOR, GET, PUT, RESET, ALLOC↳        PA
  BEGIN ALLOC(PT,PROC);    ↱PACK, UNPACK, INSERT, APPEND, READ, WRITE↳ PA
    WITH PT↑ DO                                                        PA
    BEGIN NAME := INITNAM[IT-1];  NXTEL := NEXT;  KLASS := PROC;       PA
      PROCTYPE := PT;  FORMALS := NIL;                                 PA
      PROCADDR := 0 ; PROCLEVEL := 0 ; SEGSIZE := IT ;                 PA
      PROCKIND := ACTUAL;                                              PA
    END;  NEXT := PT;                                                  PA
  END;                                                                 PA
  FOR IT := 1 TO 9 DO                                                  PA
  BEGIN ALLOC(PT,PROC);    ↱ODD, INT, CHR, EOF, ABS, SQR, TRUNC,↳      PA
    WITH PT↑ DO            ↱PRED, SUCC↳                              PA
    BEGIN NAME := INITNAM[IT+10];  NXTEL := NEXT;  KLASS := PROC;      PA
      PROCTYPE := NILPTR;  FORMALS := NIL;                             PA
      PROCADDR := 0 ; PROCLEVEL := 0 ; SEGSIZE := IT ;                 PA
      PROCKIND := ACTUAL;                                              PA
    END;  NEXT := PT;                                                  PA
  END;                                                                 PA
  ALLOC(CTPTR,TYPES,FILES);   ↱TYPE OF INPUT AND OUTPUT↳               PA
  WITH CTPTR↑ DO                                                       PA
  BEGIN NAME := BLANK;  NXTEL := NIL;  KLASS := TYPES;                 PA
    SIZE := 513;  FORM := FILES;                                       PA
  END;  FILPTS[1] := CTPTR;                                            PA
  FOR IT := 22 TO 23 DO                                                PA
```

LIST OF CONTROL, ACTIVE, AND/OR INACTIVE CARDS IN PASCSYM

```
   BEGIN ALLOC(PT,VARS);   {INPUT, OUTPUT}                          000079   PA
      WITH PT^ DO                                                           PA
      BEGIN NAME := INITNAM[IT];  NXTEL := NEXT;  KLASS := VARS;            PA
         VTYPE := CTPTR;  VKIND := ACTUAL;                                  PA
         IF IT = 22 THEN                                                    PA
            VADDR := INPT ELSE VADDR := OUTPT ;                             PA
         VLEVEL := 0;                                                       PA
      END;  NEXT := PT;                                                     PA
   END;                                                                     PA
   ALLOC(ALFAPTR,TYPES,SYMBOLIC);   {ALFA}                                  PA
   WITH ALFAPTR^ DO                                                         PA
   BEGIN NAME := INITNAM[24];  NXTEL := NEXT;  KLASS := TYPES;              PA
      SIZE := 1;  FORM := SYMBOLIC;  FCONST := NIL;  PWSET := NIL;          PA
      BITSIZE := WORDLENGTH;                                                PA
   END;                                                                     PA
   ALLOC(REALPTR,TYPES,SYMBOLIC);   {REAL}                                  PA
   WITH REALPTR^ DO                                                         PA
   BEGIN NAME := INITNAM[25];  NXTEL := ALFAPTR;  KLASS := TYPES;           PA
      SIZE := 1;  FORM := SYMBOLIC;  FCONST := NIL;  PWSET := NIL;          PA
      BITSIZE := WORDLENGTH;                                                PA
   END;                                                                     PA
   ALLOC(CHARPTR,TYPES,SYMBOLIC);   {CHAR}                                  PA
   WITH CHARPTR^ DO                                                         PA
   BEGIN NAME := INITNAM[26];  NXTEL := REALPTR;  KLASS := TYPES;           PA
      SIZE := 1;  FORM := SYMBOLIC;  BITSIZE := 7;                          PA
   END;                                                                     PA
   ALLOC(PT,KONST,ACTUAL);   {=;=}                                          PA
   WITH PT^ DO                                                              PA
   BEGIN NAME := BLANK;  NXTEL := NIL;  KLASS := KONST;                     PA
      CONTYPE := CHARPTR;  CONKIND := ACTUAL;                               PA
      VALUES := 63;  SUCC := NIL;                                           PA
   END;  CHARPTR^.FCONST := PT;                                             PA
   ALLOC(BOOLPTR,TYPES,SYMBOLIC);   {BOOLEAN}                               PA
   WITH BOOLPTR^ DO                                                         PA
   BEGIN NAME := INITNAM[27];  NXTEL := CHARPTR;  KLASS := TYPES;           PA
      SIZE := 1;  FORM := SYMBOLIC;  BITSIZE := 2;                          PA
   END;                                                                     PA
   ALLOC(PT,KONST,ACTUAL);   {EOL}                                          PA
   WITH PT^ DO                                                              PA
   BEGIN NAME := INITNAM[20]; NXTEL := BOOLPTR;                             PA
      KLASS := KONST; CONTYPE := CHARPTR; CONKIND := ACTUAL;                PA
      VALUES := 0; SUCC := NIL                                              PA
   END ;                                                                    PA
   NEXT := PT; CTPTR := NIL;                                                PA
   FOR IT := 0 TO 1 DO                                                      PA
   BEGIN ALLOC(PT,KONST,ACTUAL);   {FALSE, TRUE}                            PA
      WITH PT^ DO                                                           PA
      BEGIN NAME := INITNAM[IT+28];  NXTEL := NEXT;                         PA
         KLASS := KONST;  CONTYPE := BOOLPTR;  CONKIND := ACTUAL;           PA
         VALUES := IT;  SUCC := CTPTR;                                      PA
      END;  CTPTR := PT;  NEXT := PT;                                       PA
   END;                                                                     PA
```

LIST OF CONTROL, ACTIVE, AND/OR INACTIVE CARDS IN PASCSYM 000080

```
  BOOLPTR↑.FCONST := NEXT;                                                    PA:
  ALLOC(INTPTR,TYPES,NUMERIC);   ↱INTEGER↳                                    PA:
  WITH INTPTR↑ DO                                                             PA:
  BEGIN NAME := INITNAM[30];  NXTEL := NEXT;  KLASS := TYPES;                 PA:
     SIZE := 1 ; FORM := NUMERIC ;                                            PA:
     MIN := 777740000000000000B;  MAX := 377777777777777B;                    PA:
     BITS := WORDLENGTH;                                                      PA:
  END;  PREDEFP := INTPTR;                                                    PA:
  FOR IT := 1 TO 2 DO                                                         PA:
  BEGIN ALLOC(PT,TYPES,POWER);   ↱POWERSET CHAR, POWERSET BOOLEAN↳            PA:
     WITH PT↑ DO                                                              PA:
     BEGIN NAME := BLANK;  NXTEL := NIL;  KLASS := TYPES;                     PA:
        SIZE := 1;  FORM := POWER;                                            PA:
        IF IT = 1 THEN ↱POWERSET BOOLEAN↳                                     PA:
        BEGIN ELSET := BOOLPTR;  BOOLPTR↑.PWSET := PT;  PWBITS := 3;          PA:
        END ELSE ↱POWERSET CHAR↳                                              PA:
        BEGIN ELSET := CHARPTR;  CHARPTR↑.PWSET := PT;                        PA:
           PWBITS := WORDLENGTH;                                              PA:
        END;                                                                  PA:
     END;                                                                     PA:
  END;                                                                        PA:
  ALLOC(UNDECPTR,VARS);   ↱TYPE OF ALL UNDECLARED IDES↳                       PA:
  WITH UNDECPTR↑ DO                                                           PA:
  BEGIN NAME := BLANK ; NXTEL := NIL ; KLASS := VARS ; VTYPE := NIL ;         PA:
     VKIND := ACTUAL ; VADDR := 0 ; VLEVEL := 0 ;                             PA:
  END ;                                                                       PA:
  ALLOC(PNUMPTR,TYPES,POWER);   ↱TYPE OF POWERSET 0..58↳                      PA:
  WITH PNUMPTR↑ DO                                                            PA:
  BEGIN NAME := BLANK;  NXTEL := NIL;  KLASS := TYPES;                        PA:
     SIZE := 1;  FORM := POWER;  ELSET := INTPTR;  PWBITS := WORDLENGTH;      PA:
  END;                                                                        PA:
  ALLOC(LAMPTR,TYPES,POWER);   ↱TYPE OF [ ]↳                                  PA:
  WITH LAMPTR↑ DO                                                             PA:
  BEGIN NAME := BLANK;  NXTEL := NIL;  KLASS := TYPES;                        PA:
     SIZE := 1;  FORM := POWER;  ELSET := NIL;  PWBITS := 0;                  PA:
  END;                                                                        PA:
  ALLOC(CTPTR,KONST,FORMAL);   ↱TYPE OF PARAM OF EXTERNAL PROCEDURES↳         PA:
  WITH CTPTR↑ DO                                                              PA:
  BEGIN NAME := BLANK ; NXTEL := NIL ; KLASS := KONST ;                       PA:
     CONTYPE := REALPTR ; CONKIND := FORMAL ;                                 PA:
     CADDR := 3 ; CLEVEL := 0 ;                                               PA:
  END ;                                                                       PA:
  NEXT := INTPTR ;                                                            PA:
  ALLOC(PT,KONST,ACTUAL);   ↱ALFALENG↳                                        PA:
  WITH PT↑ DO                                                                 PA:
  BEGIN NAME := INITNAM[21] ; NXTEL := NEXT ; KLASS := KONST ;                PA:
     CONTYPE := INTPTR ; CONKIND := ACTUAL ; VALUES := 10 ;                   PA:
     SUCC := NIL                                                              PA:
  END ;                                                                       PA:
  NEXT := PT ;                                                                PA:
  FOR IT := 32 TO 37 DO                                                       PA:
  BEGIN ALLOC(PT,PROC);   ↱SIN, COS, EXP, LN, SQRT, ARCTAN↳                   PA:
```

LIST OF CONTROL, ACTIVE, AND/OR INACTIVE CARDS IN PASCSYM

000081

```
      WITH PT↑ DO                                                                    PA
      BEGIN NAME := INITNAM[IT] ; NXTEL := NEXT ;                                    PA
         KLASS := PROC ; PROCTYPE := REALPTR ;                                       PA
         PROCKIND := ACTUAL ; PROCLEVEL := 0 ;                                       PA
         FORMALS := CTPTR; PROCADDR := IT + 58;                                      PA
         IF IT ≤ 33 THEN SEGSIZE := 0 ELSE SEGSIZE := IT - 33 ;                      PA
      END ;                                                                          PA
      NEXT := PT                                                                     PA
   END ;                                                                             PA
   EXTPTR := NEXT ;                                                                  PA
   FILPTS[1]↑.FELTYPE := CHARPTR;                                                    PA
                                                                                     PA
   DIGITS := [Ξ0Ξ,Ξ1Ξ,Ξ2Ξ,Ξ3Ξ,Ξ4Ξ,Ξ5Ξ,Ξ6Ξ,Ξ7Ξ,Ξ8Ξ,Ξ9Ξ];                                    PA
   FOR IT := 0 TO 3 DO PARMLIST.ERRNRS[IT] := [ ] ;                                  PA
   ERRINX := 0;  EOLFLAG := TRUE;                                                    PA
   PARMLIST.ERRFLAG := FALSE ;                                                       PA
   DP := TRUE;  PRCODE := FALSE;                                                     PA
   PMCTR := 0;  LASTLINK := 0;  ↱FOR PMDFILE↳                                        PA
   ASSCHECK := TRUE ; INXCHECK := TRUE ; DIVCHECK := TRUE ;                          PA
   STOFLCHECK := TRUE ; ROUNDING := TRUE ;                                           PA
   KK := ALFALENG; FILTOP := -1;  JMPIX := 0;  CEXTABIX := 0;                        PA
   PTX := 0;  CHNIX := 1;  B6DPL := 3;                                               PA
   FOR IT := 1 TO UNDMAX-1 DO UNDLAB[IT].SUCC := IT + 1 ;                            PA
   UNDLAB[UNDMAX].SUCC := 0 ;                                                        PA
   IC := PARMLIST.IC0 + 1 ;                                                          PA
   MAXCTP := LAMPTR;                                                                 PA
   PARMLIST.EXTFLAGS := [ ] ;                                                        PA
   WITH DISPLAY[0] DO                                                                PA
   BEGIN FNAME := EXTPTR ; OCCUR := BLOCK END ;                                      PA
   NO := 0; LC := IC;                                                                PA
   NEXTCH;  INSYMBOL;                                                                PA
   REPEAT                                                                            PA
      DISPLAY[1].FNAME := NIL;  DISPLAY[1].OCCUR := BLOCK;                           PA
      TOP := 1;  LEVEL := 0;  NEXT := NIL;                                           PA
      ALLOC(PT,DUMMYCLASS); LC := IC ; PFTOP := -1;                                  PA
                                                                                     PA
    ↱********************                                                           PA
     *                    *                                                          PA
     *↳ BODY(NIL,PT); ↱*  COMPILE USER PROGRAM                                       PA
     *                    *                                                          PA
     *********************↳                                                          PA
                                                                                     PA
      IF (LEVEL = 0)^(NO ≠ 17) THEN ERROR(54) ;                                      PA
   UNTIL NO = 17 ↱ . ↳ ;                                                             PA
   WHILE ¬EOLFLAG DO NEXTCH; IF ERRINX > 0 THEN PRTERR;                              PA
   IF PRCODE THEN PRJMPTAB;                                                          PA
   PMDFILE↑.PKIND := PROCDR; PMDFILE↑.COUNT := LASTLINK;                             PA
   PUT(PMDFILE);  ↱LINK TO MAIN PROGRAM ENTRY↳                                       PA
   WRITE(EOL) ;                                                                      PA
   PARMLIST.LIMCODE := IC ;                                                          PA
END .                                                                                PA
```

MARY OF UPDATE IDENTIFIERS WITHIN DECK - PASCSYM

000082

IDENTIFIER	TOTAL	ACTIVE
PASCSYM	4203	4203

MARY OF UPDATE IDENTIFIERS WITHIN DECK - TOTALS

IDENTIFIER	TOTAL	ACTIVE
YANK$$$	0	0
XREFSYM	0	0
PASCSYM	4203	4203
TEST	0	0
PASCAL	0	0

```
72    R. M. PARSONS      SUM 304H  10/16/72
.T99JW4Z
.T99JW,CM40000,T200.                                    000083
.SEQ,061.
. CHARGE 07200017 01 6140
.ATTACH OLDPL,PASCPL.
.CYCLE **, PASCPL
.PFN FOUND IN SD  009
.CYCLE 01, PASCPL
.FILE HAS BEEN ATTACHED
.UPDATE,Q,L=A1234567.
.READING INPUT
.UPDATE COMPLETE
.FL   40000
.CP       8.892 SEC.
.PP       7.712 SEC.
.IO       3.515 SEC.
```

T99J061 //// END OF LIST ////