```
0001:  (*$A+  USE ASCII CHARACTER SET. *)
0002:                                                                          PASCP    2
0003:  (*$C+,T-,D-,L-*)                                                        BOOT     2
0004:   (*****************************************                            P        2
0005:    *                                       *                            P        3
0006:    *                                       *                            P        4
0007:    *         PORTABLE PASCAL COMPILER      *                            P        5
0008:    *         ************************      *                            P        6
0009:    *                                       *                            P        7
0010:    *               PASCAL P4               *                            P        8
0011:    *                                       *                            P        9
0012:    *                                       *                            P       10
0013:    *     AUTHORS:                          *                            P       11
0014:    *              URS AMMANN               *                            P       12
0015:    *              KESAV NORI               *                            P       13
0016:    *              CHRISTIAN JACOBI         *                            P       14
0017:    *                                       *                            P       15
0018:    *     ADDRESS:                          *                            P       16
0019:    *                                       *                            P       17
0020:    *          INSTITUT FUER INFORMATIK     *                            P       18
0021:    *          EIDG. TECHNISCHE HOCHSCHULE  *                            P       19
0022:    *          CH-8096 ZUERICH              *                            P       20
0023:    *                                       *                            P       21
0024:    *                                       *                            P       22
0025:    *     LAST CHANGES COMPLETED IN MAY 76  *                            P       23
0026:    *                                       *                            P       24
0027:    *                                       *                            P       25
0028:    *****************************************)                           P       26
0029:                                                                          PASCP   44
0030:                                                                          PASCP   45
0031:  PROGRAM PASCALCOMPILER(INPUT,OUTPUT,PRR);                               PASCP   46
0032:                                                                          PASCP   47
0033:                                                                          PASCP   48
0034:                                                                          PASCP   49
0035:  CONST DISPLIMIT = 20; MAXLEVEL = 10;                                    X        1
0036:     INTSIZE     = 2;   (* GROSS MINNEAPOLIS *)                          GEN      2
0037:     INTAL       = 2;                                                    GEN      3
0038:     REALSIZE    = 4;                                                    GEN      4
0039:     REALAL      = 4;                                                    GEN      5
0040:     CHARSIZE    = 1;                                                    GEN      6
0041:     CHARAL      = 1;                                                    GEN      7
0042:     CHARMAX     = 1;                                                    GEN      8
0043:     BOOLSIZE    = 1;                                                    GEN      9
0044:     BOOLAL      = 1;                                                    GEN     10
0045:     PTRSIZE     = 2;                                                    GEN     11
0046:     ADRAL       = 2;                                                    GEN     12
0047:     SETSIZE      = 8;                                                   GEN     13
0048:     SETAL       = 8;                                                    GEN     14
0049:     STACKELSIZE = 8;                                                    GEN     15
0050:     STACKAL = 8;                                                        GEN     16
0051:     STRGLGTH = 16;                                                      GEN     17
0052:     SETHIGH = 95;                                                       GEN     18
0053:     SETLOW = 0;                                                         GEN     19
0054:     ORDMAXCHAR = 95;                                                    GEN     20
0055:     ORDMINCHAR  = 0;                                                    GEN     21
0056:     LCAFTERMARKSTACK = 36;                                              GEN     22
0057:     MAXINT      =   8388607;                                            GEN     23
0058:        FILEAL = CHARAL;                                                 P       42
0059:     (* STACKELSIZE = MINIMUM SIZE FOR 1 STACKELEMENT                    P       44
0060:                     = K*STACKAL                                         P       45
0061:        STACKAL     = SCM(ALL OTHER AL-CONSTANTS)                       P       46
0062:        CHARMAX     = SCM(CHARSIZE,CHARAL)                              P       47
0063:                        SCM = SMALLEST COMMON MULTIPLE                   P       48
0064:        LCAFTERMARKSTACK >= 4*PTRSIZE+MAX(X-SIZE)                       P       49
0065:                           = K1*STACKELSIZE          *)                  P       50
```

```
0066:        MAXSTACK = 1;                                                          P       51
0067:        PARMAL = STACKAL;                                                      P       54
0068:        PARMSIZE = STACKELSIZE;                                                P       55
0069:        RECAL = STACKAL;                                                       P       56
0070:        FILEBUFFER = 4;                                                        PASCP   56
0071:        MAXADDR = MAXINT;                                                      X        2
0072:                                                                               PASCP   57
0073:                                                                               PASCP   58
0074:                                                                               PASCP   59
0075:  TYPE                                                  (*DESCRIBING:*)       PASCP   60
0076:                                                        (*************)       PASCP   61
0077:                                                                               PASCP   62
0078:                                                                               PASCP   63
0079:                                                        (*BASIC SYMBOLS*)     PASCP   64
0080:                                                        (***************)     PASCP   65
0081:                                                                               PASCP   66
0082:       SYMBOL = (IDENT,INTCONST,REALCONST,STRINGCONST,NOTSY,MULOP,ADDOP,RELOP,  PASCP   67
0083:                 LPARENT,RPARENT,LBRACK,RBRACK,COMMA,SEMICOLON,PERIOD,ARROW,   PASCP   68
0084:                 COLON,BECOMES,LABELSY,CONSTSY,TYPESY,VARSY,FUNCSY,PROGSY,     PASCP   69
0085:                 PROCSY,SETSY,PACKEDSY,ARRAYSY,RECORDSY,FILESY,FORWARDSY,      PASCP   70
0086:                 BEGINSY,IFSY,CASESY,REPEATSY,WHILESY,FORSY,WITHSY,            PASCP   71
0087:                 GOTOSY,ENDSY,ELSESY,UNTILSY,OFSY,DOSY,TOSY,DOWNTOSY,          PASCP   72
0088:                 THENSY,OTHERSY);                                              PASCP   73
0089:       OPERATOR = (MUL,RDIV,ANDOP,IDIV,IMOD,PLUS,MINUS,OROP,LTOP,LEOP,GEOP,GTOP, PASCP   74
0090:                   NEOP,EQOP,INOP,NOOP);                                       PASCP   75
0091:       SETOFSYS = SET OF SYMBOL;                                               PASCP   76
0092:       CHTP = (LETTER,NUMBER,SPECIAL,ILLEGAL);                                 P       57
0093:                                                                               PASCP   77
0094:                                                        (*CONSTANTS*)         PASCP   78
0095:                                                        (***********)         PASCP   79
0096:                                                                               PASCP   80
0097:       CSTCLASS = (REEL,PSET,STRG);                                            PASCP   81
0098:       CSP = ^ CONSTANT;                                                       PASCP   82
0099:       CONSTANT = RECORD CASE CCLASS: CSTCLASS OF                              PASCP   83
0100:                           REEL: (RVAL: PACKED ARRAY [1..STRGLGTH] OF CHAR);   PASCP   84
0101:                           PSET: (PVAL: SET OF 0..58);                         PASCP   85
0102:                           STRG: (SLGTH: 0..STRGLGTH;                          PASCP   86
0103:                                  SVAL: PACKED ARRAY [1..STRGLGTH] OF CHAR)    PASCP   87
0104:                         END;                                                  PASCP   88
0105:                                                                               PASCP   89
0106:       VALU = RECORD CASE INTVAL: BOOLEAN OF  (*INTVAL NEVER SET NORE TESTED*) PASCP   90
0107:                       TRUE:  (IVAL: INTEGER);                                 PASCP   91
0108:                       FALSE: (VALP: CSP)                                      PASCP   92
0109:                     END;                                                      PASCP   93
0110:                                                                               PASCP   94
0111:                                                          (*DATA STRUCTURES*)  PASCP   95
0112:                                                          (*****************)  PASCP   96
0113:       LEVRANGE = 0..MAXLEVEL; ADDRRANGE = 0..MAXADDR;                         PASCP   97
0114:       STRUCTFORM = (SCALAR,SUBRANGE,POINTER,POWER,ARRAYS,RECORDS,FILES,       PASCP   98
0115:                     TAGFLD,VARIANT);                                          PASCP   99
0116:       DECLKIND = (STANDARD,DECLARED);                                         PASCP  100
0117:       STP = ^ STRUCTURE; CTP = ^ IDENTIFIER;                                  PASCP  101
0118:                                                                               PASCP  102
0119:       STRUCTURE = PACKED RECORD                                               PASCP  103
0120:                     MARKED: BOOLEAN;   (*FOR TEST PHASE ONLY*)                PASCP  104
0121:                     SIZE: ADDRRANGE;                                          PASCP  105
0122:                     CASE FORM: STRUCTFORM OF                                  PASCP  106
0123:                       SCALAR:   (CASE SCALKIND: DECLKIND OF                   PASCP  107
0124:                                    DECLARED: (FCONST: CTP));                  PASCP  108
0125:                       SUBRANGE: (RANGETYPE: STP; MIN,MAX: VALU);              PASCP  109
0126:                       POINTER:  (ELTYPE: STP);                                PASCP  110
0127:                       POWER:    (ELSET: STP);                                 PASCP  111
0128:                       ARRAYS:   (AELTYPE,INXTYPE: STP);                       PASCP  112
0129:                       RECORDS:  (FSTFLD: CTP; RECVAR: STP);                   PASCP  113
0130:                       FILES:    (FILTYPE: STP);                               PASCP  114
```

```
0131:                       TAGFLD:   (TAGFIELDP: CTP; FSTVAR: STP);                    PASCP  115
0132:                       VARIANT:  (NXTVAR,SUBVAR: STP; VARVAL: VALU)                PASCP  116
0133:                     END;                                                          PASCP  117
0134:                                                                                   PASCP  118
0135:                                                           (*NAMES*)               PASCP  119
0136:                                                           (*******)               PASCP  120
0137:                                                                                   PASCP  121
0138:       IDCLASS = (TYPES,KONST,VARS,FIELD,PROC,FUNC);                               PASCP  122
0139:       SETOFIDS = SET OF IDCLASS;                                                  PASCP  123
0140:       IDKIND = (ACTUAL,FORMAL);                                                   PASCP  124
0141:       ALPHA = PACKED ARRAY [1..8] OF CHAR;                                        PASCP  125
0142:                                                                                   PASCP  126
0143:       IDENTIFIER = PACKED RECORD                                                  PASCP  127
0144:                     NAME: ALPHA; LLINK, RLINK: CTP;                               PASCP  128
0145:                     IDTYPE: STP; NEXT: CTP;                                       PASCP  129
0146:                     CASE KLASS: IDCLASS OF                                        PASCP  130
0147:                       KONST: (VALUES: VALU);                                      PASCP  131
0148:                       VARS:  (VKIND: IDKIND; VLEV: LEVRANGE; VADDR: ADDRRANGE);   PASCP  132
0149:                       FIELD: (FLDADDR: ADDRRANGE);                                PASCP  133
0150:                       PROC,                                                       PASCP  134
0151:                       FUNC:  (CASE PFDECKIND: DECLKIND OF                         PASCP  135
0152:                                STANDARD: (KEY: 1..15);                            PASCP  136
0153:                                DECLARED: (PFLEV: LEVRANGE; PFNAME: INTEGER;       PASCP  137
0154:                                           CASE PFKIND: IDKIND OF                  PASCP  138
0155:                                            ACTUAL: (FORWDECL, EXTERN:             PASCP  139
0156:                                                     BOOLEAN)))                    PASCP  140
0157:                     END;                                                          PASCP  141
0158:                                                                                   PASCP  142
0159:                                                                                   PASCP  143
0160:       DISPRANGE = 0..DISPLIMIT;                                                   PASCP  144
0161:       WHERE = (BLCK,CREC,VREC,REC);                                               PASCP  145
0162:                                                                                   PASCP  146
0163:                                                           (*EXPRESSIONS*)         PASCP  147
0164:                                                           (************)         PASCP  148
0165:       ATTRKIND = (CST,VARBL,EXPR);                                                PASCP  149
0166:       VACCESS = (DRCT,INDRCT,INXD);                                               PASCP  150
0167:                                                                                   PASCP  151
0168:       ATTR = RECORD TYPTR: STP;                                                   PASCP  152
0169:                CASE KIND: ATTRKIND OF                                             PASCP  153
0170:                  CST:   (CVAL: VALU);                                             PASCP  154
0171:                  VARBL: (CASE ACCESS: VACCESS OF                                  PASCP  155
0172:                            DRCT: (VLEVEL: LEVRANGE; DPLMT: ADDRRANGE);            PASCP  156
0173:                            INDRCT: (IDPLMT: ADDRRANGE))                           PASCP  157
0174:                END;                                                               PASCP  158
0175:                                                                                   PASCP  159
0176:       TESTP = ^ TESTPOINTER;                                                      PASCP  160
0177:       TESTPOINTER = PACKED RECORD                                                 PASCP  161
0178:                       ELT1,ELT2 : STP;                                            PASCP  162
0179:                       LASTTESTP : TESTP                                           PASCP  163
0180:                     END;                                                          PASCP  164
0181:                                                                                   PASCP  165
0182:                                                               (*LABELS*)          PASCP  166
0183:                                                               (********)          PASCP  167
0184:       LBP = ^ LABL;                                                               PASCP  168
0185:       LABL = RECORD NEXTLAB: LBP; DEFINED: BOOLEAN;                               PASCP  169
0186:                     LABVAL, LABNAME: INTEGER                                      PASCP  170
0187:              END;                                                                 PASCP  171
0188:                                                                                   PASCP  172
0189:       EXTFILEP = ^FILEREC;                                                        PASCP  173
0190:       FILEREC = RECORD FILENAME:ALPHA; NEXTFILE:EXTFILEP END;                     PASCP  174
0191:                                                                                   PASCP  175
0192:  (*-------------------------------------------------------------------------*)   PASCP  176
0193:                                                                                   PASCP  177
0194:                                                                                   PASCP  178
0195:  VAR                                                                             PASCP  179
```

```
0196:                                  (*RETURNED BY SOURCE PROGRAM SCANNER       PASCP  181
0197:                                   INSYMBOL:                                 PASCP  182
0198:                                   **********)                               PASCP  183
0199:                                                                             PASCP  184
0200:      SY: SYMBOL;                 (*LAST SYMBOL*)                           PASCP  185
0201:      OP: OPERATOR;               (*CLASSIFICATION OF LAST SYMBOL*)         PASCP  186
0202:      VAL: VALU;                  (*VALUE OF LAST CONSTANT*)                PASCP  187
0203:      LGTH: INTEGER;              (*LENGTH OF LAST STRING CONSTANT*)        PASCP  188
0204:      ID: ALPHA;                  (*LAST IDENTIFIER (POSSIBLY TRUNCATED)*)  PASCP  189
0205:      KK: 1..8;                   (*NR OF CHARS IN LAST IDENTIFIER*)        PASCP  190
0206:      CH: CHAR;                   (*LAST CHARACTER*)                        PASCP  191
0207:      EOL: BOOLEAN;               (*END OF LINE FLAG*)                      PASCP  192
0208:                                                                            PASCP  193
0209:                                                                            PASCP  194
0210:                                  (*COUNTERS:*)                             PASCP  195
0211:                                  (**********)                              PASCP  196
0212:                                                                            PASCP  197
0213:      CHCNT: INTEGER;             (*CHARACTER COUNTER*)                     P       58
0214:      LC,IC: ADDRRANGE;           (*DATA LOCATION AND INSTRUCTION COUNTER*) PASCP  199
0215:      LINECOUNT: INTEGER;                                                   PASCP  200
0216:                                                                            PASCP  201
0217:                                                                            PASCP  202
0218:                                  (*SWITCHES:*)                             PASCP  203
0219:                                  (**********)                              PASCP  204
0220:                                                                            PASCP  205
0221:      DP,                         (*DECLARATION PART*)                      PASCP  206
0222:      PRTERR,                   (*TO ALLOW FORWARD REFERENCES IN POINTER TYPE PASCP  207
0223:                                  DECLARATION BY SUPPRESSING ERROR MESSAGE*) PASCP  208
0224:      LIST,PRCODE,PRTABLES: BOOLEAN;  (*OUTPUT OPTIONS FOR                  PASCP  209
0225:                                         -- SOURCE PROGRAM LISTING          PASCP  210
0226:                                         -- PRINTING SYMBOLIC CODE          PASCP  211
0227:                                         -- DISPLAYING IDENT AND STRUCT TABLES PASCP 212
0228:                                         --> PROCEDURE OPTION*)             PASCP  213
0229:      DEBUG: BOOLEAN;                                                       P       59
0230:                                                                            PASCP  214
0231:                                                                            PASCP  215
0232:                                  (*POINTERS:*)                             PASCP  216
0233:                                  (**********)                              PASCP  217
0234:      PARMPTR,                                                              P       60
0235:      INTPTR,REALPTR,CHARPTR,                                               PASCP  218
0236:      BOOLPTR,NILPTR,TEXTPTR: STP;    (*POINTERS TO ENTRIES OF STANDARD IDS*) PASCP 219
0237:      UTYPPTR,UCSTPTR,UVARPTR,                                              PASCP  220
0238:      UFLDPTR,UPRCPTR,UFCTPTR,        (*POINTERS TO ENTRIES FOR UNDECLARED IDS*) PASCP 221
0239:      FWPTR: CTP;                     (*HEAD OF CHAIN OF FORW DECL TYPE IDS*) PASCP 222
0240:      FEXTFILEP: EXTFILEP;            (*HEAD OF CHAIN OF EXTERNAL FILES*)   PASCP  223
0241:      GLOBTESTP: TESTP;                (*LAST TESTPOINTER*)                 PASCP  224
0242:                                                                            PASCP  225
0243:                                                                            PASCP  226
0244:                                  (*BOOKKEEPING OF DECLARATION LEVELS:*)    PASCP  227
0245:                                  (***********************************)     PASCP  228
0246:                                                                            PASCP  229
0247:      LEVEL: LEVRANGE;            (*CURRENT STATIC LEVEL*)                  PASCP  230
0248:      DISX,                       (*LEVEL OF LAST ID SEARCHED BY SEARCHID*) PASCP  231
0249:      TOP: DISPRANGE;             (*TOP OF DISPLAY*)                        PASCP  232
0250:                                                                            PASCP  233
0251:      DISPLAY:                    (*WHERE:   MEANS:*)                       PASCP  234
0252:        ARRAY [DISPRANGE] OF                                                PASCP  235
0253:          PACKED RECORD           (*=BLCK:   ID IS VARIABLE ID*)            PASCP  236
0254:            FNAME: CTP; FLABEL: LBP;  (*=CREC:   ID IS FIELD ID IN RECORD WITH*) PASCP 237
0255:            CASE OCCUR: WHERE OF    (*         CONSTANT ADDRESS*)         PASCP  238
0256:              CREC: (CLEV: LEVRANGE;  (*=VREC:   ID IS FIELD ID IN RECORD WITH*) PASCP 239
0257:                     CDSPL: ADDRRANGE);(*        VARIABLE ADDRESS*)        PASCP  240
0258:              VREC: (VDSPL: ADDRRANGE)                                      PASCP  241
0259:            END;                    (* --> PROCEDURE WITHSTATEMENT*)        PASCP  242
0260:                                                                            PASCP  243
```

```
0261:                                                                                PASCP  244
0262:                                  (*ERROR MESSAGES:*)                           PASCP  245
0263:                                  (*****************)                           PASCP  246
0264:                                                                                PASCP  247
0265:      ERRINX: 0..10;              (*NR OF ERRORS IN CURRENT SOURCE LINE*)       PASCP  248
0266:      ERRLIST:                                                                  PASCP  249
0267:        ARRAY [1..10] OF                                                        PASCP  250
0268:          PACKED RECORD POS: INTEGER;                                           P       61
0269:                        NMR: 1..400                                             PASCP  252
0270:                 END;                                                         PASCP  253
0271:                                                                                PASCP  254
0272:                                                                                PASCP  255
0273:                                                                                PASCP  256
0274:                                                                                PASCP  257
0275:                                  (*EXPRESSION COMPILATION:*)                   PASCP  258
0276:                                  (*************************)                   PASCP  259
0277:                                                                                PASCP  260
0278:      GATTR: ATTR;                (*DESCRIBES THE EXPR CURRENTLY COMPILED*)     PASCP  261
0279:                                                                                PASCP  262
0280:                                                                                PASCP  263
0281:                                  (*STRUCTURED CONSTANTS:*)                     PASCP  264
0282:                                  (************************)                    PASCP  265
0283:                                                                                PASCP  266
0284:      CONSTBEGSYS,SIMPTYPEBEGSYS,TYPEBEGSYS,BLOCKBEGSYS,SELECTSYS,FACBEGSYS,    PASCP  267
0285:      STATBEGSYS,TYPEDELS: SETOFSYS;                                            PASCP  268
0286:      CHARTP : ARRAY[CHAR] OF CHTP;                                             P       62
0287:      RW:  ARRAY [1..35(*NR. OF RES. WORDS*)] OF ALPHA;                         PASCP  269
0288:      FRW: ARRAY [1..9] OF 1..36(*NR. OF RES. WORDS + 1*);                      PASCP  270
0289:      RSY: ARRAY [1..35(*NR. OF RES. WORDS*)] OF SYMBOL;                        PASCP  271
0290:     SSY: ARRAY [CHAR] OF SYMBOL;                                               J        1
0291:      ROP: ARRAY [1..35(*NR. OF RES. WORDS*)] OF OPERATOR;                      PASCP  273
0292:     SOP: ARRAY [CHAR] OF OPERATOR;                                             J        2
0293:      NA:  ARRAY [1..35] OF ALPHA;                                              PASCP  275
0294:      MN:  ARRAY[0..60] OF PACKED ARRAY[1..4] OF CHAR;                          P       63
0295:      SNA: ARRAY [1..23] OF PACKED ARRAY [1..4] OF CHAR;                        PASCP  277
0296:      CDX: ARRAY[0..60] OF -4..+4;                                              P       64
0297:      PDX: ARRAY[1..23] OF -7..+7;                                              P       65
0298:      ORDINT: ARRAY[CHAR] OF INTEGER;                                           CH       1
0299:                                                                                CH       2
0300:      INTLABEL,MXINT10,DIGMAX: INTEGER;                                         PASCP  279
0301:                                                                                PASCP  280
0302:  (*-------------------------------------------------------------------------*)  PASCP  281
0303:                                                                                PASCP  282
0304:                                                                                PASCP  283
0305:  PROCEDURE ENDOFLINE;                                                         PASCP  284
0306:      VAR LASTPOS,FREEPOS,CURRPOS,CURRNMR,F,K: INTEGER;                         PASCP  285
0307:    BEGIN                                                                       PASCP  286
0308:      IF ERRINX > 0 THEN   (*OUTPUT ERROR MESSAGES*)                            PASCP  287
0309:        BEGIN WRITE(OUTPUT,' ****  ':15);                                      PASCP  288
0310:          LASTPOS := 0; FREEPOS := 1;                                         PASCP  289
0311:          FOR K := 1 TO ERRINX DO                                             PASCP  290
0312:            BEGIN                                                             PASCP  291
0313:              WITH ERRLIST[K] DO                                              PASCP  292
0314:                BEGIN CURRPOS := POS; CURRNMR := NMR END;                     PASCP  293
0315:              IF CURRPOS = LASTPOS THEN WRITE(OUTPUT,',')                     PASCP  294
0316:              ELSE                                                            PASCP  295
0317:                BEGIN                                                         PASCP  296
0318:                  WHILE FREEPOS < CURRPOS DO                                  PASCP  297
0319:                    BEGIN WRITE(OUTPUT,' '); FREEPOS := FREEPOS + 1 END;      PASCP  298
0320:                  WRITE(OUTPUT,'^');                                          PASCP  299
0321:                  LASTPOS := CURRPOS                                          PASCP  300
0322:                END;                                                          PASCP  301
0323:              IF CURRNMR < 10 THEN F := 1                                     PASCP  302
0324:              ELSE IF CURRNMR < 100 THEN F := 2                               PASCP  303
0325:                ELSE F := 3;                                                  PASCP  304
```

```
0326:              WRITE(OUTPUT,CURRNMR:F);                                          PASCP  305
0327:              FREEPOS := FREEPOS + F + 1                                        PASCP  306
0328:            END;                                                                PASCP  307
0329:          WRITELN(OUTPUT); ERRINX := 0                                          PASCP  308
0330:        END;                                                                    PASCP  309
0331:      IF LIST AND (NOT EOF(INPUT)) THEN                                         P       66
0332:        BEGIN LINECOUNT := LINECOUNT + 1; WRITE(OUTPUT,LINECOUNT:6,'  ':2);     PASCP  311
0333:          IF DP THEN WRITE(OUTPUT,LC:7) ELSE WRITE(OUTPUT,IC:7);                PASCP  312
0334:          WRITE(OUTPUT,' ')                                                     PASCP  313
0335:        END;                                                                    PASCP  314
0336:      CHCNT := 0                                                                PASCP  315
0337:    END  (*ENDOFLINE*) ;                                                        PASCP  316
0338:                                                                                PASCP  317
0339:    PROCEDURE ERROR(FERRNR: INTEGER);                                           PASCP  318
0340:    BEGIN                                                                       PASCP  319
0341:      IF ERRINX >= 9 THEN                                                       PASCP  320
0342:        BEGIN ERRLIST[10].NMR := 255; ERRINX := 10 END                          PASCP  321
0343:      ELSE                                                                      PASCP  322
0344:        BEGIN ERRINX := ERRINX + 1;                                             PASCP  323
0345:          ERRLIST[ERRINX].NMR := FERRNR                                         PASCP  324
0346:        END;                                                                    PASCP  325
0347:      ERRLIST[ERRINX].POS := CHCNT                                              PASCP  326
0348:    END (*ERROR*) ;                                                             PASCP  327
0349:                                                                                PASCP  328
0350:    PROCEDURE INSYMBOL;                                                         PASCP  329
0351:      (*READ NEXT BASIC SYMBOL OF SOURCE PROGRAM AND RETURN ITS                 PASCP  330
0352:      DESCRIPTION IN THE GLOBAL VARIABLES SY, OP, ID, VAL AND LGTH*)            PASCP  331
0353:      LABEL 1,2,3;                                                              PASCP  332
0354:      VAR I,K: INTEGER;                                                         PASCP  333
0355:          DIGIT: PACKED ARRAY [1..STRGLGTH] OF CHAR;                            PASCP  334
0356:          STRING: PACKED ARRAY [1..STRGLGTH] OF CHAR;                           PASCP  335
0357:          LVP: CSP;TEST: BOOLEAN;                                               PASCP  336
0358:                                                                                PASCP  337
0359:      PROCEDURE NEXTCH;                                                         PASCP  338
0360:      BEGIN IF EOL THEN                                                         PASCP  339
0361:        BEGIN IF LIST THEN WRITELN(OUTPUT); ENDOFLINE                           PASCP  340
0362:        END;                                                                    PASCP  341
0363:        IF NOT EOF(INPUT) THEN                                                  PASCP  342
0364:         BEGIN EOL := EOLN(INPUT); READ(INPUT,CH);                              PASCP  343
0365:          IF LIST THEN WRITE(OUTPUT,CH);                                        PASCP  344
0366:          CHCNT := CHCNT + 1                                                    PASCP  345
0367:         END                                                                    PASCP  346
0368:        ELSE                                                                    P       67
0369:          BEGIN WRITELN(OUTPUT,'   *** EOF ','ENCOUNTERED');                    P       68
0370:            TEST := FALSE                                                       P       69
0371:          END                                                                   P       70
0372:      END;                                                                      PASCP  348
0373:                                                                                PASCP  349
0374:      PROCEDURE OPTIONS;                                                        PASCP  350
0375:      BEGIN                                                                     PASCP  351
0376:        REPEAT NEXTCH;                                                          PASCP  352
0377:          IF CH <> '*' THEN                                                     PASCP  353
0378:            BEGIN                                                               PASCP  354
0379:              IF CH = 'T' THEN                                                  PASCP  355
0380:                BEGIN NEXTCH; PRTABLES := CH = '+' END                          PASCP  356
0381:              ELSE                                                              PASCP  357
0382:                IF CH = 'L' THEN                                                PASCP  358
0383:                  BEGIN NEXTCH; LIST := CH = '+';                               PASCP  359
0384:                    IF NOT LIST THEN WRITELN(OUTPUT)                            PASCP  360
0385:                  END                                                           PASCP  361
0386:                ELSE                                                            PASCP  362
0387:               IF CH = 'D' THEN                                                 P       71
0388:                 BEGIN NEXTCH; DEBUG := CH = '+' END                            P       72
0389:               ELSE                                                             P       73
0390:                  IF CH = 'C' THEN                                              PASCP  363
```

```
0391:                   BEGIN NEXTCH; PRCODE := CH = '+' END;                    PASCP  364
0392:             NEXTCH                                                         PASCP  365
0393:           END                                                              PASCP  366
0394:       UNTIL CH <> ','                                                      PASCP  367
0395:     END (*OPTIONS*) ;                                                      PASCP  368
0396:                                                                            PASCP  369
0397:   BEGIN (*INSYMBOL*)                                                       PASCP  370
0398:   1:                                                                       PASCP  371
0399:     REPEAT WHILE (CH = ' ') AND NOT EOL DO NEXTCH;                         PASCP  372
0400:       TEST := EOL;                                                         PASCP  373
0401:       IF TEST THEN NEXTCH                                                  PASCP  374
0402:     UNTIL NOT TEST;                                                        PASCP  375
0403:     IF CHARTP[CH] = ILLEGAL THEN                                           P       74
0404:       BEGIN SY := OTHERSY; OP := NOOP;                                     P       75
0405:         ERROR(399); NEXTCH                                                 P       76
0406:       END                                                                  P       77
0407:     ELSE                                                                   P       78
0408:     CASE CH OF                                                             PASCP  376
0409:       'A','B','C','D','E','F','G','H','I',                                 PASCP  377
0410:       'J','K','L','M','N','O','P','Q','R',                                 PASCP  378
0411:       'S','T','U','V','W','X','Y','Z':                                     PASCP  379
0412:         BEGIN K := 0;                                                      PASCP  380
0413:           REPEAT                                                           PASCP  381
0414:             IF K < 8 THEN                                                  PASCP  382
0415:              BEGIN K := K + 1; ID[K] := CH END ;                           PASCP  383
0416:             NEXTCH                                                         PASCP  384
0417:            UNTIL CHARTP[CH] IN [SPECIAL,ILLEGAL];                          P       79
0418:           IF K >= KK THEN KK := K                                          PASCP  386
0419:           ELSE                                                             PASCP  387
0420:             REPEAT ID[KK] := ' '; KK := KK - 1                             PASCP  388
0421:             UNTIL KK = K;                                                  PASCP  389
0422:           FOR I := FRW[K] TO FRW[K+1] - 1 DO                               PASCP  390
0423:             IF RW[I] = ID THEN                                             PASCP  391
0424:               BEGIN SY := RSY[I]; OP := ROP[I]; GOTO 2 END;                PASCP  392
0425:             SY := IDENT; OP := NOOP;                                       PASCP  393
0426:   2:    END;                                                              PASCP  394
0427:       '0','1','2','3','4','5','6','7','8','9':                            PASCP  395
0428:         BEGIN OP := NOOP; I := 0;                                          PASCP  396
0429:           REPEAT I := I+1; IF I<= DIGMAX THEN DIGIT[I] := CH; NEXTCH       PASCP  397
0430:           UNTIL CHARTP[CH] <> NUMBER;                                      P       80
0431:           IF (CH = '.') OR (CH = 'E') THEN                                 PASCP  399
0432:             BEGIN                                                          PASCP  400
0433:                   K := I;                                                  PASCP  401
0434:                   IF CH = '.' THEN                                         PASCP  402
0435:                     BEGIN K := K+1; IF K <= DIGMAX THEN DIGIT[K] := CH;    PASCP  403
0436:                       NEXTCH; IF CH = '.' THEN BEGIN CH := ':'; GOTO 3 END;PASCP  404
0437:                       IF CHARTP[CH] <> NUMBER THEN ERROR(201)              P       81
0438:                       ELSE                                                 PASCP  407
0439:                         REPEAT K := K + 1;                                 PASCP  408
0440:                           IF K <= DIGMAX THEN DIGIT[K] := CH; NEXTCH       PASCP  409
0441:                         UNTIL CHARTP[CH] <>  NUMBER                        P       82
0442:                     END;                                                   PASCP  411
0443:                   IF CH = 'E' THEN                                         PASCP  412
0444:                     BEGIN K := K+1; IF K <= DIGMAX THEN DIGIT[K] := CH;    PASCP  413
0445:                       NEXTCH;                                              PASCP  414
0446:                       IF (CH = '+') OR (CH ='-') THEN                      PASCP  415
0447:                         BEGIN K := K+1; IF K <= DIGMAX THEN DIGIT[K] := CH;PASCP  416
0448:                           NEXTCH                                           PASCP  417
0449:                         END;                                               PASCP  418
0450:                       IF CHARTP[CH] <> NUMBER THEN ERROR(201)              P       83
0451:                       ELSE                                                 PASCP  421
0452:                         REPEAT K := K+1;                                   PASCP  422
0453:                           IF K <= DIGMAX THEN DIGIT[K] := CH; NEXTCH       PASCP  423
0454:                         UNTIL CHARTP[CH] <> NUMBER                         P       84
0455:                      END;                                                  PASCP  425
```

```
0456:                   NEW(LVP,REEL); SY:= REALCONST; LVP^.CCLASS := REEL;          PASCP  426
0457:                   WITH LVP^ DO                                                 PASCP  427
0458:                     BEGIN FOR I := 1 TO STRGLGTH DO RVAL[I] := ' ';            PASCP  428
0459:                       IF K <= DIGMAX THEN                                      PASCP  429
0460:                         FOR I := 2 TO K + 1 DO RVAL[I] := DIGIT[I-1]           PASCP  430
0461:                       ELSE BEGIN ERROR(203); RVAL[2] := '0';                   PASCP  431
0462:                              RVAL[3] := '.'; RVAL[4] := '0'                    PASCP  432
0463:                            END                                                 PASCP  433
0464:                     END;                                                       PASCP  434
0465:                   VAL.VALP := LVP                                              PASCP  435
0466:               END                                                              PASCP  436
0467:             ELSE                                                               PASCP  437
0468:    3:         BEGIN                                                           PASCP  438
0469:                 IF I > DIGMAX THEN BEGIN ERROR(203); VAL.IVAL := 0 END       PASCP  439
0470:                 ELSE                                                         PASCP  440
0471:                   WITH VAL DO                                                PASCP  441
0472:                     BEGIN IVAL := 0;                                         PASCP  442
0473:                       FOR K := 1 TO I DO                                     PASCP  443
0474:                         BEGIN                                                PASCP  444
0475:                           IF IVAL <= MXINT10 THEN                            PASCP  445
0476:                             IVAL := IVAL*10+ORDINT[DIGIT[K]]                 CH       3
0477:                           ELSE BEGIN ERROR(203); IVAL := 0 END               PASCP  447
0478:                         END;                                                 PASCP  448
0479:                       SY := INTCONST                                         PASCP  449
0480:                     END                                                      PASCP  450
0481:               END                                                            PASCP  451
0482:           END;                                                               PASCP  452
0483:         '''':                                                                PASCP  453
0484:           BEGIN LGTH := 0; SY := STRINGCONST;  OP := NOOP;                   PASCP  454
0485:             REPEAT                                                           PASCP  455
0486:               REPEAT NEXTCH; LGTH := LGTH + 1;                               PASCP  456
0487:                      IF LGTH <= STRGLGTH THEN STRING[LGTH] := CH             PASCP  457
0488:               UNTIL (EOL) OR (CH = '''');                                    PASCP  458
0489:               IF EOL THEN ERROR(202) ELSE NEXTCH                             PASCP  459
0490:             UNTIL CH <> '''';                                                PASCP  460
0491:             LGTH := LGTH - 1;   (*NOW LGTH = NR OF CHARS IN STRING*)         PASCP  461
0492:             IF LGTH = 1 THEN VAL.IVAL := ORD(STRING[1])                      PASCP  462
0493:             ELSE                                                             PASCP  463
0494:               BEGIN NEW(LVP,STRG); LVP^.CCLASS:=STRG;                        PASCP  464
0495:                 IF LGTH > STRGLGTH THEN                                      PASCP  465
0496:                   BEGIN ERROR(399); LGTH := STRGLGTH END;                    PASCP  466
0497:                 WITH LVP^ DO                                                 PASCP  467
0498:                   BEGIN SLGTH := LGTH;                                       PASCP  468
0499:                     FOR I := 1 TO LGTH DO SVAL[I] := STRING[I]               PASCP  469
0500:                   END;                                                       PASCP  470
0501:                 VAL.VALP := LVP                                              PASCP  471
0502:               END                                                            PASCP  472
0503:           END;                                                               PASCP  473
0504:         ':':                                                                 PASCP  474
0505:           BEGIN OP := NOOP; NEXTCH;                                          PASCP  475
0506:             IF CH = '=' THEN                                                 PASCP  476
0507:               BEGIN SY := BECOMES; NEXTCH END                                PASCP  477
0508:             ELSE SY := COLON                                                 PASCP  478
0509:           END;                                                               PASCP  479
0510:         '.':                                                                 PASCP  480
0511:           BEGIN OP := NOOP; NEXTCH;                                          PASCP  481
0512:             IF CH = '.' THEN                                                 PASCP  482
0513:               BEGIN SY := COLON; NEXTCH END                                  PASCP  483
0514:             ELSE SY := PERIOD                                                PASCP  484
0515:           END;                                                               PASCP  485
0516:         '<':                                                                 PASCP  486
0517:           BEGIN NEXTCH; SY := RELOP;                                         PASCP  487
0518:             IF CH = '=' THEN                                                 PASCP  488
0519:               BEGIN OP := LEOP; NEXTCH END                                   PASCP  489
0520:             ELSE                                                             PASCP  490
```

```
0521:              IF CH = '>' THEN                                                    PASCP  491
0522:                BEGIN OP := NEOP; NEXTCH END                                      PASCP  492
0523:              ELSE OP := LTOP                                                     PASCP  493
0524:            END;                                                                  PASCP  494
0525:          '>':                                                                    PASCP  495
0526:            BEGIN NEXTCH; SY := RELOP;                                            PASCP  496
0527:              IF CH = '=' THEN                                                    PASCP  497
0528:                BEGIN OP := GEOP; NEXTCH END                                      PASCP  498
0529:              ELSE OP := GTOP                                                     PASCP  499
0530:            END;                                                                  PASCP  500
0531:          '(':                                                                    PASCP  501
0532:           BEGIN NEXTCH;                                                        PASCP  502
0533:             IF CH = '*' THEN                                                    PASCP  503
0534:               BEGIN NEXTCH;                                                     PASCP  504
0535:                 IF CH = '$' THEN OPTIONS;                                       PASCP  505
0536:                 REPEAT                                                          PASCP  506
0537:                   WHILE CH <> '*'  DO NEXTCH;                                   PASCP  507
0538:                   NEXTCH                                                        PASCP  508
0539:                 UNTIL CH = ')';                                                 PASCP  509
0540:                 NEXTCH; GOTO 1                                                  PASCP  510
0541:               END;                                                              PASCP  511
0542:             SY := LPARENT; OP := NOOP                                           PASCP  512
0543:           END;                                                                  PASCP  513
0544:          '*','+','-',                                                          PASCP  514
0545:          '=','/',')',                                                          PASCP  515
0546:          '[',']',',',';','^','$':                                              PASCP  516
0547:            BEGIN SY := SSY[CH]; OP := SOP[CH];                                 PASCP  517
0548:              NEXTCH                                                            PASCP  518
0549:            END;                                                                PASCP  519
0550:          ' ': SY := OTHERSY                                                    P       85
0551:        END (*CASE*)                                                            PASCP  523
0552:      END (*INSYMBOL*) ;                                                        PASCP  524
0553:                                                                                PASCP  525
0554:      PROCEDURE ENTERID(FCP: CTP);                                              PASCP  526
0555:        (*ENTER ID POINTED AT BY FCP INTO THE NAME-TABLE,                       PASCP  527
0556:         WHICH ON EACH DECLARATION LEVEL IS ORGANISED AS                        PASCP  528
0557:         AN UNBALANCED BINARY TREE*)                                           PASCP  529
0558:        VAR NAM: ALPHA; LCP, LCP1: CTP; LLEFT: BOOLEAN;                         PASCP  530
0559:      BEGIN NAM := FCP^.NAME;                                                   PASCP  531
0560:        LCP := DISPLAY[TOP].FNAME;                                              PASCP  532
0561:        IF LCP = NIL THEN                                                       PASCP  533
0562:          DISPLAY[TOP].FNAME := FCP                                             PASCP  534
0563:        ELSE                                                                    PASCP  535
0564:          BEGIN                                                                 PASCP  536
0565:            REPEAT LCP1 := LCP;                                                 PASCP  537
0566:              IF LCP^.NAME = NAM THEN   (*NAME CONFLICT, FOLLOW RIGHT LINK*)    PASCP  538
0567:                BEGIN ERROR(101); LCP := LCP^.RLINK; LLEFT := FALSE END         PASCP  539
0568:              ELSE                                                              PASCP  540
0569:                IF LCP^.NAME < NAM THEN                                         PASCP  541
0570:                  BEGIN LCP := LCP^.RLINK; LLEFT := FALSE END                   PASCP  542
0571:                ELSE BEGIN LCP := LCP^.LLINK; LLEFT := TRUE END                 PASCP  543
0572:            UNTIL LCP = NIL;                                                    PASCP  544
0573:            IF LLEFT THEN LCP1^.LLINK := FCP ELSE LCP1^.RLINK := FCP            PASCP  545
0574:          END;                                                                  PASCP  546
0575:        FCP^.LLINK := NIL; FCP^.RLINK := NIL                                    PASCP  547
0576:      END (*ENTERID*) ;                                                         PASCP  548
0577:                                                                                PASCP  549
0578:      PROCEDURE SEARCHSECTION(FCP: CTP; VAR FCP1: CTP);                         PASCP  550
0579:        (*TO FIND RECORD FIELDS AND FORWARD DECLARED PROCEDURE ID'S              PASCP  551
0580:         --> PROCEDURE PROCEDUREDECLARATION                                     PASCP  552
0581:         --> PROCEDURE SELECTOR*)                                               PASCP  553
0582:         LABEL 1;                                                               PASCP  554
0583:      BEGIN                                                                     PASCP  555
0584:        WHILE FCP <> NIL DO                                                     PASCP  556
0585:          IF FCP^.NAME = ID THEN GOTO 1                                         PASCP  557
```

```
0586:        ELSE IF FCP^.NAME < ID THEN FCP := FCP^.RLINK                    PASCP  558
0587:          ELSE FCP := FCP^.LLINK;                                        PASCP  559
0588:  1:  FCP1 := FCP                                                        PASCP  560
0589:    END (*SEARCHSECTION*) ;                                              PASCP  561
0590:                                                                        PASCP  562
0591:    PROCEDURE SEARCHID(FIDCLS: SETOFIDS; VAR FCP: CTP);                 PASCP  563
0592:      LABEL 1;                                                          PASCP  564
0593:      VAR LCP: CTP;                                                     PASCP  565
0594:    BEGIN                                                               PASCP  566
0595:      FOR DISX := TOP DOWNTO 0 DO                                       PASCP  567
0596:        BEGIN LCP := DISPLAY[DISX].FNAME;                               PASCP  568
0597:          WHILE LCP <> NIL DO                                           PASCP  569
0598:            IF LCP^.NAME = ID THEN                                      PASCP  570
0599:              IF LCP^.KLASS IN FIDCLS THEN GOTO 1                       PASCP  571
0600:              ELSE                                                      PASCP  572
0601:                BEGIN IF PRTERR THEN ERROR(103);                        PASCP  573
0602:                  LCP := LCP^.RLINK                                     PASCP  574
0603:                END                                                     PASCP  575
0604:            ELSE                                                        PASCP  576
0605:              IF LCP^.NAME < ID THEN                                    PASCP  577
0606:                LCP := LCP^.RLINK                                       PASCP  578
0607:              ELSE LCP := LCP^.LLINK                                    PASCP  579
0608:        END;                                                            PASCP  580
0609:      (*SEARCH NOT SUCCSESSFUL; SUPPRESS ERROR MESSAGE IN CASE          PASCP  581
0610:       OF FORWARD REFERENCED TYPE ID IN POINTER TYPE DEFINITION         PASCP  582
0611:       --> PROCEDURE SIMPLETYPE*)                                       PASCP  583
0612:      IF PRTERR THEN                                                    PASCP  584
0613:        BEGIN ERROR(104);                                               PASCP  585
0614:          (*TO AVOID RETURNING NIL, REFERENCE AN ENTRY                  PASCP  586
0615:           FOR AN UNDECLARED ID OF APPROPRIATE CLASS                    PASCP  587
0616:           --> PROCEDURE ENTERUNDECL*)                                  PASCP  588
0617:          IF TYPES IN FIDCLS THEN LCP := UTYPPTR                        PASCP  589
0618:          ELSE                                                          PASCP  590
0619:            IF VARS IN FIDCLS THEN LCP := UVARPTR                       PASCP  591
0620:            ELSE                                                        PASCP  592
0621:              IF FIELD IN FIDCLS THEN LCP := UFLDPTR                    PASCP  593
0622:              ELSE                                                      PASCP  594
0623:                IF KONST IN FIDCLS THEN LCP := UCSTPTR                  PASCP  595
0624:                ELSE                                                    PASCP  596
0625:                  IF PROC IN FIDCLS THEN LCP := UPRCPTR                 PASCP  597
0626:                  ELSE LCP := UFCTPTR;                                  PASCP  598
0627:        END;                                                            PASCP  599
0628:  1:  FCP := LCP                                                       PASCP  600
0629:    END (*SEARCHID*) ;                                                  PASCP  601
0630:                                                                        PASCP  602
0631:    PROCEDURE GETBOUNDS(FSP: STP; VAR FMIN,FMAX: INTEGER);              PASCP  603
0632:      (*GET INTERNAL BOUNDS OF SUBRANGE OR SCALAR TYPE*)                PASCP  604
0633:     (*ASSUME FSP<>INTPTR AND FSP<>REALPTR*)                            P       86
0634:    BEGIN                                                               PASCP  607
0635:      FMIN := 0; FMAX := 0;                                             P       87
0636:      IF FSP <> NIL THEN                                                P       88
0637:      WITH FSP^ DO                                                      PASCP  608
0638:        IF FORM = SUBRANGE THEN                                         PASCP  609
0639:          BEGIN FMIN := MIN.IVAL; FMAX := MAX.IVAL END                  PASCP  610
0640:        ELSE                                                            PASCP  611
0641:            IF FSP = CHARPTR THEN                                       P       89
0642:              BEGIN FMIN := ORDMINCHAR; FMAX := ORDMAXCHAR              P       90
0643:              END                                                       P       91
0644:            ELSE                                                        PASCP  614
0645:              IF FCONST <> NIL THEN                                     P       92
0646:                FMAX := FCONST^.VALUES.IVAL                             P       93
0647:    END (*GETBOUNDS*) ;                                                 PASCP  619
0648:                                                                        P       94
0649:    FUNCTION ALIGNQUOT(FSP: STP): INTEGER;                              P       95
0650:    BEGIN                                                               P       96
```

```
0651:      ALIGNQUOT := 1;                                                    P       97
0652:      IF FSP <> NIL THEN                                                 P       98
0653:        WITH FSP^ DO                                                     P       99
0654:          CASE FORM OF                                                   P      100
0655:            SCALAR:   IF FSP=INTPTR THEN ALIGNQUOT := INTAL              P      101
0656:                      ELSE IF FSP=BOOLPTR THEN ALIGNQUOT := BOOLAL       P      102
0657:                      ELSE IF SCALKIND=DECLARED THEN ALIGNQUOT := INTAL  P      103
0658:                      ELSE IF FSP=CHARPTR THEN ALIGNQUOT := CHARAL       P      104
0659:                      ELSE IF FSP=REALPTR THEN ALIGNQUOT := REALAL       P      105
0660:                      ELSE (*PARMPTR*) ALIGNQUOT := PARMAL;              P      106
0661:            SUBRANGE: ALIGNQUOT := ALIGNQUOT(RANGETYPE);                 P      107
0662:            POINTER:  ALIGNQUOT := ADRAL;                                P      108
0663:            POWER:    ALIGNQUOT := SETAL;                                P      109
0664:            FILES:    ALIGNQUOT := FILEAL;                               P      110
0665:            ARRAYS:   ALIGNQUOT := ALIGNQUOT(AELTYPE);                   P      111
0666:            RECORDS:  ALIGNQUOT := RECAL;                                P      112
0667:            VARIANT,TAGFLD: ERROR(501)                                   P      113
0668:          END                                                            P      114
0669:    END (*ALIGNQUOT*);                                                   P      115
0670:                                                                         P      116
0671:    PROCEDURE ALIGN(FSP: STP; VAR FLC: INTEGER);                         P      117
0672:      VAR K,L: INTEGER;                                                  P      118
0673:    BEGIN                                                                P      119
0674:      K := ALIGNQUOT(FSP);                                               P      120
0675:      L := FLC-1;                                                        P      121
0676:      FLC := L + K  -  L MOD K                                           P      122
0677:    END (*ALIGN*);                                                       P      123
0678:                                                                         PASCP  620
0679:    PROCEDURE PRINTTABLES(FB: BOOLEAN);                                  PASCP  621
0680:      (*PRINT DATA STRUCTURE AND NAME TABLE*)                            PASCP  622
0681:      VAR I, LIM: DISPRANGE;                                             PASCP  623
0682:                                                                         PASCP  624
0683:      PROCEDURE MARKER;                                                  PASCP  625
0684:        (*MARK DATA STRUCTURE ENTRIES TO AVOID MULTIPLE PRINTOUT*)       PASCP  626
0685:        VAR I: INTEGER;                                                  PASCP  627
0686:                                                                         PASCP  628
0687:        PROCEDURE MARKCTP(FP: CTP); FORWARD;                             PASCP  629
0688:                                                                         PASCP  630
0689:        PROCEDURE MARKSTP(FP: STP);                                      PASCP  631
0690:          (*MARK DATA STRUCTURES, PREVENT CYCLES*)                       PASCP  632
0691:        BEGIN                                                            PASCP  633
0692:          IF FP <> NIL THEN                                              PASCP  634
0693:            WITH FP^ DO                                                  PASCP  635
0694:              BEGIN MARKED := TRUE;                                      PASCP  636
0695:                CASE FORM OF                                             PASCP  637
0696:                SCALAR:   ;                                              PASCP  638
0697:                SUBRANGE: MARKSTP(RANGETYPE);                            PASCP  639
0698:                POINTER:  (*DON'T MARK ELTYPE: CYCLE POSSIBLE; WILL BE MARKED  PASCP  640
0699:                          ANYWAY, IF FP = TRUE*) ;                       PASCP  641
0700:                POWER:    MARKSTP(ELSET) ;                               PASCP  642
0701:                ARRAYS:   BEGIN MARKSTP(AELTYPE); MARKSTP(INXTYPE) END;  PASCP  643
0702:                RECORDS:  BEGIN MARKCTP(FSTFLD); MARKSTP(RECVAR) END;    PASCP  644
0703:                FILES:    MARKSTP(FILTYPE);                              PASCP  645
0704:                TAGFLD:   MARKSTP(FSTVAR);                               PASCP  646
0705:                VARIANT:  BEGIN MARKSTP(NXTVAR); MARKSTP(SUBVAR) END     PASCP  647
0706:                END (*CASE*)                                             PASCP  648
0707:              END (*WITH*)                                               PASCP  649
0708:        END (*MARKSTP*);                                                 PASCP  650
0709:                                                                         PASCP  651
0710:        PROCEDURE MARKCTP;                                               PASCP  652
0711:        BEGIN                                                            PASCP  653
0712:          IF FP <> NIL THEN                                              PASCP  654
0713:            WITH FP^ DO                                                  PASCP  655
0714:              BEGIN MARKCTP(LLINK); MARKCTP(RLINK);                      PASCP  656
0715:                MARKSTP(IDTYPE)                                          PASCP  657
```

```
0716:              END                                                                  PASCP  658
0717:        END (*MARKCTP*);                                                           PASCP  659
0718:                                                                                   PASCP  660
0719:      BEGIN (*MARK*)                                                               PASCP  661
0720:        FOR I := TOP DOWNTO LIM DO                                                 PASCP  662
0721:          MARKCTP(DISPLAY[I].FNAME)                                                PASCP  663
0722:      END (*MARK*);                                                                PASCP  664
0723:                                                                                   PASCP  665
0724:      PROCEDURE FOLLOWCTP(FP: CTP); FORWARD;                                       PASCP  666
0725:                                                                                   PASCP  667
0726:      PROCEDURE FOLLOWSTP(FP: STP);                                                PASCP  668
0727:      BEGIN                                                                        PASCP  669
0728:        IF FP <> NIL THEN                                                          PASCP  670
0729:          WITH FP^ DO                                                              PASCP  671
0730:            IF MARKED THEN                                                         PASCP  672
0731:              BEGIN MARKED := FALSE; WRITE(OUTPUT,' ':4,ORD(FP):6,SIZE:10);        PASCP  673
0732:                CASE FORM OF                                                       PASCP  674
0733:                SCALAR:   BEGIN WRITE(OUTPUT,'SCALAR':10);                         PASCP  675
0734:                            IF SCALKIND = STANDARD THEN                            PASCP  676
0735:                             WRITE(OUTPUT,'STANDARD':10)                           PASCP  677
0736:                            ELSE WRITE(OUTPUT,'DECLARED':10,' ':4,ORD(FCONST):6); PASCP  678
0737:                            WRITELN(OUTPUT)                                        PASCP  679
0738:                          END;                                                     PASCP  680
0739:                SUBRANGE:BEGIN                                                     PASCP  681
0740:                          WRITE(OUTPUT,'SUBRANGE':10,' ':4,ORD(RANGETYPE):6);      PASCP  682
0741:                            IF RANGETYPE <> REALPTR THEN                           PASCP  683
0742:                              WRITE(OUTPUT,MIN.IVAL,MAX.IVAL)                      PASCP  684
0743:                            ELSE                                                   PASCP  685
0744:                              IF (MIN.VALP <> NIL) AND (MAX.VALP <> NIL) THEN      PASCP  686
0745:                                WRITE(OUTPUT,' ',MIN.VALP^.RVAL:9,                 PASCP  687
0746:                                      ' ',MAX.VALP^.RVAL:9);                       PASCP  688
0747:                            WRITELN(OUTPUT); FOLLOWSTP(RANGETYPE);                 PASCP  689
0748:                          END;                                                     PASCP  690
0749:                POINTER:  WRITELN(OUTPUT,'POINTER':10,' ':4,ORD(ELTYPE):6);       PASCP  691
0750:                POWER:    BEGIN WRITELN(OUTPUT,'SET':10,' ':4,ORD(ELSET):6);       PASCP  692
0751:                            FOLLOWSTP(ELSET)                                       PASCP  693
0752:                          END;                                                     PASCP  694
0753:                ARRAYS:   BEGIN                                                    PASCP  695
0754:                           WRITELN(OUTPUT,'ARRAY':10,' ':4,ORD(AELTYPE):6,' ':4,  PASCP  696
0755:                            ORD(INXTYPE):6);                                       PASCP  697
0756:                            FOLLOWSTP(AELTYPE); FOLLOWSTP(INXTYPE)                 PASCP  698
0757:                          END;                                                     PASCP  699
0758:                RECORDS:  BEGIN                                                    PASCP  700
0759:                          WRITELN(OUTPUT,'RECORD':10,' ':4,ORD(FSTFLD):6,' ':4,    PASCP  701
0760:                            ORD(RECVAR):6); FOLLOWCTP(FSTFLD);                     PASCP  702
0761:                            FOLLOWSTP(RECVAR)                                      PASCP  703
0762:                          END;                                                     PASCP  704
0763:                FILES:    BEGIN WRITE(OUTPUT,'FILE':10,' ':4,ORD(FILTYPE):6);      PASCP  705
0764:                            FOLLOWSTP(FILTYPE)                                     PASCP  706
0765:                          END;                                                     PASCP  707
0766:                TAGFLD:   BEGIN WRITELN(OUTPUT,'TAGFLD':10,' ':4,ORD(TAGFIELDP):6,PASCP  708
0767:                            ' ':4,ORD(FSTVAR):6);                                  PASCP  709
0768:                            FOLLOWSTP(FSTVAR)                                      PASCP  710
0769:                          END;                                                     PASCP  711
0770:                VARIANT:  BEGIN WRITELN(OUTPUT,'VARIANT':10,' ':4,ORD(NXTVAR):6,  PASCP  712
0771:                            ' ':4,ORD(SUBVAR):6,VARVAL.IVAL);                      PASCP  713
0772:                            FOLLOWSTP(NXTVAR); FOLLOWSTP(SUBVAR)                   PASCP  714
0773:                          END                                                      PASCP  715
0774:                END (*CASE*)                                                       PASCP  716
0775:              END (*IF MARKED*)                                                    PASCP  717
0776:      END (*FOLLOWSTP*);                                                           PASCP  718
0777:                                                                                   PASCP  719
0778:      PROCEDURE FOLLOWCTP;                                                         PASCP  720
0779:        VAR I: INTEGER;                                                            PASCP  721
0780:      BEGIN                                                                        PASCP  722
```

```
0781:        IF FP <> NIL THEN                                                    PASCP  723
0782:          WITH FP^ DO                                                        PASCP  724
0783:            BEGIN WRITE(OUTPUT,' ':4,ORD(FP):6,' ',NAME:9,' ':4,ORD(LLINK):6,  PASCP  725
0784:              ' ':4,ORD(RLINK):6,' ':4,ORD(IDTYPE):6);                       PASCP  726
0785:              CASE KLASS OF                                                  PASCP  727
0786:                TYPES: WRITE(OUTPUT,'TYPE':10);                              PASCP  728
0787:                KONST: BEGIN WRITE(OUTPUT,'CONSTANT':10,' ':4,ORD(NEXT):6);  PASCP  729
0788:                       IF IDTYPE <> NIL THEN                                 PASCP  730
0789:                           IF IDTYPE = REALPTR THEN                          PASCP  731
0790:                             BEGIN                                           PASCP  732
0791:                               IF VALUES.VALP <> NIL THEN                    PASCP  733
0792:                                 WRITE(OUTPUT,' ',VALUES.VALP^.RVAL:9)       PASCP  734
0793:                             END                                             PASCP  735
0794:                           ELSE                                              PASCP  736
0795:                             IF IDTYPE^.FORM = ARRAYS THEN  (*STRINGCONST*)  PASCP  737
0796:                               BEGIN                                         PASCP  738
0797:                                 IF VALUES.VALP <> NIL THEN                  PASCP  739
0798:                                   BEGIN WRITE(OUTPUT,' ');                  PASCP  740
0799:                                     WITH VALUES.VALP^ DO                    PASCP  741
0800:                                       FOR I := 1 TO SLGTH DO                PASCP  742
0801:                                        WRITE(OUTPUT,SVAL[I])                PASCP  743
0802:                                   END                                       PASCP  744
0803:                               END                                           PASCP  745
0804:                             ELSE WRITE(OUTPUT,VALUES.IVAL)                  PASCP  746
0805:                        END;                                                 PASCP  747
0806:                VARS:  BEGIN WRITE(OUTPUT,'VARIABLE':10);                    PASCP  748
0807:                          IF VKIND = ACTUAL THEN WRITE(OUTPUT,'ACTUAL':10)  PASCP  749
0808:                          ELSE WRITE(OUTPUT,'FORMAL':10);                   PASCP  750
0809:                          WRITE(OUTPUT,' ':4,ORD(NEXT):6,VLEV,' ':4,VADDR:6 );  PASCP  751
0810:                        END;                                                PASCP  752
0811:                FIELD: WRITE(OUTPUT,'FIELD':10,' ':4,ORD(NEXT):6,' ':4,FLDADDR:6);PASCP  753
0812:                PROC,                                                       PASCP  754
0813:                FUNC:  BEGIN                                                PASCP  755
0814:                          IF KLASS = PROC THEN WRITE(OUTPUT,'PROCEDURE':10) PASCP  756
0815:                          ELSE WRITE(OUTPUT,'FUNCTION':10);                 PASCP  757
0816:                          IF PFDECKIND = STANDARD THEN                      PASCP  758
0817:                           WRITE(OUTPUT,'STANDARD':10,                      PASCP  759
0818:                            KEY:10)                                         PASCP  760
0819:                          ELSE                                              PASCP  761
0820:                            BEGIN WRITE(OUTPUT,'DECLARED':10,' ':4,ORD(NEXT):6);  PASCP  762
0821:                              WRITE(OUTPUT,PFLEV,' ':4,PFNAME:6);           PASCP  763
0822:                              IF PFKIND = ACTUAL THEN                       PASCP  764
0823:                                BEGIN WRITE(OUTPUT,'ACTUAL':10);            PASCP  765
0824:                                  IF FORWDECL THEN WRITE(OUTPUT,'FORWARD':10)  PASCP  766
0825:                                  ELSE WRITE(OUTPUT,'NOTFORWARD':10);       PASCP  767
0826:                                  IF EXTERN THEN WRITE(OUTPUT,'EXTERN':10)  PASCP  768
0827:                                  ELSE WRITE(OUTPUT,'NOT EXTERN':10);       PASCP  769
0828:                                END                                         PASCP  770
0829:                              ELSE WRITE(OUTPUT,'FORMAL':10)                PASCP  771
0830:                            END                                             PASCP  772
0831:                        END                                                 PASCP  773
0832:              END (*CASE*);                                                 PASCP  774
0833:              WRITELN(OUTPUT); FOLLOWCTP(LLINK); FOLLOWCTP(RLINK);          PASCP  775
0834:              FOLLOWSTP(IDTYPE)                                             PASCP  776
0835:            END (*WITH*)                                                    PASCP  777
0836:      END (*FOLLOWCTP*);                                                    PASCP  778
0837:                                                                            PASCP  779
0838:    BEGIN (*PRINTTABLES*)                                                   PASCP  780
0839:      WRITELN(OUTPUT); WRITELN(OUTPUT); WRITELN(OUTPUT);                    PASCP  781
0840:      IF FB THEN LIM := 0                                                   PASCP  782
0841:      ELSE BEGIN LIM := TOP; WRITE(OUTPUT,' LOCAL') END;                    PASCP  783
0842:      WRITELN(OUTPUT,' TABLES '); WRITELN(OUTPUT);                          PASCP  784
0843:      MARKER;                                                               PASCP  785
0844:      FOR I := TOP DOWNTO LIM DO                                            PASCP  786
0845:        FOLLOWCTP(DISPLAY[I].FNAME);                                        PASCP  787
```

```
0846:        WRITELN(OUTPUT);                                                    PASCP  788
0847:        IF NOT EOL THEN WRITE(OUTPUT,' ':CHCNT+16)                          PASCP  789
0848:    END (*PRINTTABLES*);                                                    PASCP  790
0849:                                                                            PASCP  791
0850:    PROCEDURE GENLABEL(VAR NXTLAB: INTEGER);                                PASCP  792
0851:    BEGIN INTLABEL := INTLABEL + 1;                                         PASCP  793
0852:      NXTLAB := INTLABEL                                                    PASCP  794
0853:    END (*GENLABEL*);                                                       PASCP  795
0854:                                                                            PASCP  796
0855:    PROCEDURE BLOCK(FSYS: SETOFSYS; FSY: SYMBOL; FPROCP: CTP);              PASCP  797
0856:      VAR LSY: SYMBOL; TEST: BOOLEAN;                                       PASCP  798
0857:                                                                            PASCP  799
0858:      PROCEDURE SKIP(FSYS: SETOFSYS);                                       PASCP  800
0859:        (*SKIP INPUT STRING UNTIL RELEVANT SYMBOL FOUND*)                   PASCP  801
0860:      BEGIN                                                                 P      124
0861:        IF NOT EOF(INPUT) THEN                                              P      125
0862:          BEGIN WHILE NOT(SY IN FSYS) AND (NOT EOF(INPUT)) DO INSYMBOL;     P      126
0863:            IF NOT (SY IN FSYS) THEN INSYMBOL                               P      127
0864:          END                                                               P      128
0865:      END (*SKIP*) ;                                                        PASCP  803
0866:                                                                            PASCP  804
0867:      PROCEDURE CONSTANT(FSYS: SETOFSYS; VAR FSP: STP; VAR FVALU: VALU);    PASCP  805
0868:        VAR LSP: STP; LCP: CTP; SIGN: (NONE,POS,NEG);                       PASCP  806
0869:            LVP: CSP; I: 2..STRGLGTH;                                       PASCP  807
0870:      BEGIN LSP := NIL; FVALU.IVAL := 0;                                    PASCP  808
0871:        IF NOT(SY IN CONSTBEGSYS) THEN                                      PASCP  809
0872:          BEGIN ERROR(50); SKIP(FSYS+CONSTBEGSYS) END;                      PASCP  810
0873:        IF SY IN CONSTBEGSYS THEN                                           PASCP  811
0874:          BEGIN                                                             PASCP  812
0875:            IF SY = STRINGCONSTSY THEN                                      PASCP  813
0876:              BEGIN                                                         PASCP  814
0877:                IF LGTH = 1 THEN LSP := CHARPTR                             PASCP  815
0878:                ELSE                                                        PASCP  816
0879:                  BEGIN                                                     PASCP  817
0880:                    NEW(LSP,ARRAYS);                                        PASCP  818
0881:                    WITH LSP^ DO                                            PASCP  819
0882:                      BEGIN AELTYPE := CHARPTR; INXTYPE := NIL;             PASCP  820
0883:                         SIZE := LGTH*CHARSIZE; FORM := ARRAYS              PASCP  821
0884:                      END                                                   PASCP  822
0885:                  END;                                                      PASCP  823
0886:                FVALU := VAL; INSYMBOL                                      PASCP  824
0887:              END                                                           PASCP  825
0888:            ELSE                                                            PASCP  826
0889:              BEGIN                                                         PASCP  827
0890:                SIGN := NONE;                                               PASCP  828
0891:                IF (SY = ADDOP) AND (OP IN [PLUS,MINUS]) THEN               PASCP  829
0892:                  BEGIN IF OP = PLUS THEN SIGN := POS ELSE SIGN := NEG;     PASCP  830
0893:                    INSYMBOL                                                PASCP  831
0894:                  END;                                                      PASCP  832
0895:                IF SY = IDENT THEN                                          PASCP  833
0896:                  BEGIN SEARCHID([KONST],LCP);                              PASCP  834
0897:                    WITH LCP^ DO                                            PASCP  835
0898:                      BEGIN LSP := IDTYPE; FVALU := VALUES END;             PASCP  836
0899:                    IF SIGN <> NONE THEN                                    PASCP  837
0900:                      IF LSP = INTPTR THEN                                  PASCP  838
0901:                        BEGIN IF SIGN = NEG THEN FVALU.IVAL := -FVALU.IVAL END PASCP 839
0902:                      ELSE                                                  PASCP  840
0903:                        IF LSP = REALPTR THEN                               PASCP  841
0904:                          BEGIN                                             PASCP  842
0905:                            IF SIGN = NEG THEN                              PASCP  843
0906:                              BEGIN NEW(LVP,REEL);                          PASCP  844
0907:                                IF FVALU.VALP^.RVAL[1] = '-' THEN           PASCP  845
0908:                                  LVP^.RVAL[1] := '+'                       PASCP  846
0909:                                ELSE LVP^.RVAL[1] := '-';                   PASCP  847
0910:                                FOR I := 2 TO STRGLGTH DO                   PASCP  848
```

```
0911:                                LVP^.RVAL[I] := FVALU.VALP^.RVAL[I];          PASCP  849
0912:                              FVALU.VALP := LVP;                              PASCP  850
0913:                            END                                               PASCP  851
0914:                          END                                                 PASCP  852
0915:                        ELSE ERROR(105);                                      PASCP  853
0916:                    INSYMBOL;                                                 PASCP  854
0917:                  END                                                         PASCP  855
0918:                ELSE                                                          PASCP  856
0919:                  IF SY = INTCONST THEN                                       PASCP  857
0920:                    BEGIN IF SIGN = NEG THEN VAL.IVAL := -VAL.IVAL;           PASCP  858
0921:                      LSP := INTPTR; FVALU := VAL; INSYMBOL                   PASCP  859
0922:                    END                                                       PASCP  860
0923:                  ELSE                                                        PASCP  861
0924:                    IF SY = REALCONST THEN                                    PASCP  862
0925:                      BEGIN IF SIGN = NEG THEN VAL.VALP^.RVAL[1] := '-';      PASCP  863
0926:                        LSP := REALPTR; FVALU := VAL; INSYMBOL                PASCP  864
0927:                      END                                                     PASCP  865
0928:                    ELSE                                                      PASCP  866
0929:                      BEGIN ERROR(106); SKIP(FSYS) END                        PASCP  867
0930:              END;                                                            PASCP  868
0931:            IF NOT (SY IN FSYS) THEN                                          PASCP  869
0932:              BEGIN ERROR(6); SKIP(FSYS) END                                  PASCP  870
0933:            END;                                                            PASCP  871
0934:        FSP := LSP                                                          PASCP  872
0935:      END (*CONSTANT*) ;                                                    PASCP  873
0936:                                                                            PASCP  874
0937:      FUNCTION EQUALBOUNDS(FSP1,FSP2: STP): BOOLEAN;                        P      129
0938:        VAR LMIN1,LMIN2,LMAX1,LMAX2: INTEGER;                               P      130
0939:      BEGIN                                                                 P      131
0940:        IF (FSP1=NIL) OR (FSP2=NIL) THEN EQUALBOUNDS := TRUE                P      132
0941:        ELSE                                                                P      133
0942:          BEGIN                                                             P      134
0943:            GETBOUNDS(FSP1,LMIN1,LMAX1);                                    P      135
0944:            GETBOUNDS(FSP1,LMIN2,LMAX2);                                    P      136
0945:            EQUALBOUNDS := (LMIN1=LMIN2) AND (LMAX1=LMAX2)                  P      137
0946:          END                                                               P      138
0947:      END (*EQUALBOUNDS*) ;                                                 P      139
0948:                                                                            P      140
0949:      FUNCTION COMPTYPES(FSP1,FSP2: STP) : BOOLEAN;                         PASCP  875
0950:        (*DECIDE WHETHER STRUCTURES POINTED AT BY FSP1 AND FSP2 ARE COMPATIBLE*) PASCP  876
0951:        VAR NXT1,NXT2: CTP; COMP: BOOLEAN;                                  PASCP  877
0952:          LTESTP1,LTESTP2 : TESTP;                                          PASCP  878
0953:      BEGIN                                                                 PASCP  879
0954:        IF FSP1 = FSP2 THEN COMPTYPES := TRUE                               PASCP  880
0955:        ELSE                                                                PASCP  881
0956:          IF (FSP1 <> NIL) AND (FSP2 <> NIL) THEN                           PASCP  882
0957:            IF FSP1^.FORM = FSP2^.FORM THEN                                 PASCP  883
0958:              CASE FSP1^.FORM OF                                            PASCP  884
0959:                SCALAR:                                                     PASCP  885
0960:                  COMPTYPES := FALSE;                                       PASCP  886
0961:                  (* IDENTICAL SCALARS DECLARED ON DIFFERENT LEVELS ARE     PASCP  887
0962:                   NOT RECOGNIZED TO BE COMPATIBLE*)                        PASCP  888
0963:                SUBRANGE:                                                   PASCP  889
0964:                  COMPTYPES := COMPTYPES(FSP1^.RANGETYPE,FSP2^.RANGETYPE);  PASCP  890
0965:                POINTER:                                                    PASCP  891
0966:                    BEGIN                                                   PASCP  892
0967:                      COMP := FALSE; LTESTP1 := GLOBTESTP;                  PASCP  893
0968:                      LTESTP2 := GLOBTESTP;                                 PASCP  894
0969:                      WHILE LTESTP1 <> NIL DO                               PASCP  895
0970:                        WITH LTESTP1^ DO                                    PASCP  896
0971:                          BEGIN                                             PASCP  897
0972:                            IF (ELT1 = FSP1^.ELTYPE) AND                    PASCP  898
0973:                              (ELT2 = FSP2^.ELTYPE) THEN COMP := TRUE;      PASCP  899
0974:                            LTESTP1 := LASTTESTP                            PASCP  900
0975:                          END;                                              PASCP  901
```

```
0976:                     IF NOT COMP THEN                                                PASCP  902
0977:                       BEGIN NEW(LTESTP1);                                           PASCP  903
0978:                         WITH LTESTP1^ DO                                            PASCP  904
0979:                           BEGIN ELT1 := FSP1^.ELTYPE;                               PASCP  905
0980:                             ELT2 := FSP2^.ELTYPE;                                   PASCP  906
0981:                             LASTTESTP := GLOBTESTP                                  PASCP  907
0982:                           END;                                                      PASCP  908
0983:                         GLOBTESTP := LTESTP1;                                       PASCP  909
0984:                         COMP := COMPTYPES(FSP1^.ELTYPE,FSP2^.ELTYPE)                PASCP  910
0985:                       END;                                                          PASCP  911
0986:                     COMPTYPES := COMP; GLOBTESTP := LTESTP2                         PASCP  912
0987:                   END;                                                            PASCP  913
0988:                 POWER:                                                            PASCP  914
0989:                   COMPTYPES := COMPTYPES(FSP1^.ELSET,FSP2^.ELSET);                PASCP  915
0990:                 ARRAYS:                                                           PASCP  916
0991:                   BEGIN                                                           P      141
0992:                     COMP := COMPTYPES(FSP1^.AELTYPE,FSP2^.AELTYPE)                P      142
0993:                         AND COMPTYPES(FSP1^.INXTYPE,FSP2^.INXTYPE);               P      143
0994:                     COMPTYPES := COMP AND                                         P      144
0995:                         EQUALBOUNDS(FSP1^.INXTYPE,FSP2^.INXTYPE)                  P      145
0996:                    END;                                                           P      146
0997:                 RECORDS:                                                          PASCP  923
0998:                   BEGIN NXT1 := FSP1^.FSTFLD; NXT2 := FSP2^.FSTFLD; COMP:=TRUE;   PASCP  924
0999:                     WHILE (NXT1 <> NIL) AND (NXT2 <> NIL) DO                      PASCP  925
1000:                       BEGIN COMP:=COMP AND COMPTYPES(NXT1^.IDTYPE,NXT2^.IDTYPE);  PASCP  926
1001:                         NXT1 := NXT1^.NEXT; NXT2 := NXT2^.NEXT                    PASCP  927
1002:                       END;                                                        PASCP  928
1003:                     COMPTYPES := COMP AND (NXT1 = NIL) AND (NXT2 = NIL)           PASCP  929
1004:                                  AND(FSP1^.RECVAR = NIL)AND(FSP2^.RECVAR = NIL)   PASCP  930
1005:                   END;                                                            PASCP  931
1006:                   (*IDENTICAL RECORDS ARE RECOGNIZED TO BE COMPATIBLE             PASCP  932
1007:                    IFF NO VARIANTS OCCUR*)                                        PASCP  933
1008:                 FILES:                                                            PASCP  934
1009:                   COMPTYPES := COMPTYPES(FSP1^.FILTYPE,FSP2^.FILTYPE)             PASCP  935
1010:               END (*CASE*)                                                        PASCP  936
1011:             ELSE (*FSP1^.FORM <> FSP2^.FORM*)                                     PASCP  937
1012:               IF FSP1^.FORM = SUBRANGE THEN                                       PASCP  938
1013:                 COMPTYPES := COMPTYPES(FSP1^.RANGETYPE,FSP2)                      PASCP  939
1014:               ELSE                                                                PASCP  940
1015:                 IF FSP2^.FORM = SUBRANGE THEN                                     PASCP  941
1016:                   COMPTYPES := COMPTYPES(FSP1,FSP2^.RANGETYPE)                    PASCP  942
1017:                 ELSE COMPTYPES := FALSE                                           PASCP  943
1018:           ELSE COMPTYPES := TRUE                                                  PASCP  944
1019:       END (*COMPTYPES*) ;                                                         PASCP  945
1020:                                                                                   PASCP  946
1021:       FUNCTION STRING(FSP: STP) : BOOLEAN;                                        PASCP  947
1022:       BEGIN STRING := FALSE;                                                      PASCP  948
1023:         IF FSP <> NIL THEN                                                        PASCP  949
1024:           IF FSP^.FORM = ARRAYS THEN                                              PASCP  950
1025:             IF COMPTYPES(FSP^.AELTYPE,CHARPTR) THEN STRING := TRUE                PASCP  951
1026:       END (*STRING*) ;                                                            PASCP  952
1027:                                                                                   PASCP  953
1028:       PROCEDURE TYP(FSYS: SETOFSYS; VAR FSP: STP; VAR FSIZE: ADDRRANGE);          PASCP  954
1029:         VAR LSP,LSP1,LSP2: STP; OLDTOP: DISPRANGE; LCP: CTP;                      PASCP  955
1030:             LSIZE,DISPL: ADDRRANGE; LMIN,LMAX: INTEGER;                           PASCP  956
1031:                                                                                   PASCP  957
1032:         PROCEDURE SIMPLETYPE(FSYS:SETOFSYS; VAR FSP:STP; VAR FSIZE:ADDRRANGE);    PASCP  958
1033:           VAR LSP,LSP1: STP; LCP,LCP1: CTP; TTOP: DISPRANGE;                      PASCP  959
1034:               LCNT: INTEGER; LVALU: VALU;                                         PASCP  960
1035:         BEGIN FSIZE := 1;                                                         PASCP  961
1036:           IF NOT (SY IN SIMPTYPEBEGSYS) THEN                                      PASCP  962
1037:             BEGIN ERROR(1); SKIP(FSYS + SIMPTYPEBEGSYS) END;                      PASCP  963
1038:           IF SY IN SIMPTYPEBEGSYS THEN                                            PASCP  964
1039:             BEGIN                                                                 PASCP  965
1040:               IF SY = LPARENT THEN                                                PASCP  966
```

```
1041:              BEGIN TTOP := TOP;   (*DECL. CONSTS LOCAL TO INNERMOST BLOCK*)    PASCP  967
1042:                WHILE DISPLAY[TOP].OCCUR <> BLCK DO TOP := TOP - 1;            PASCP  968
1043:                NEW(LSP,SCALAR,DECLARED);                                      PASCP  969
1044:                WITH LSP^ DO                                                   PASCP  970
1045:                  BEGIN SIZE := INTSIZE; FORM := SCALAR;                       PASCP  971
1046:                    SCALKIND := DECLARED                                       PASCP  972
1047:                  END;                                                         PASCP  973
1048:                LCP1 := NIL; LCNT := 0;                                        PASCP  974
1049:                REPEAT INSYMBOL;                                               PASCP  975
1050:                  IF SY = IDENT THEN                                           PASCP  976
1051:                    BEGIN NEW(LCP,KONST);                                      PASCP  977
1052:                      WITH LCP^ DO                                             PASCP  978
1053:                        BEGIN NAME := ID; IDTYPE := LSP; NEXT := LCP1;         PASCP  979
1054:                          VALUES.IVAL := LCNT; KLASS := KONST                  PASCP  980
1055:                        END;                                                   PASCP  981
1056:                      ENTERID(LCP);                                            PASCP  982
1057:                      LCNT := LCNT + 1;                                        PASCP  983
1058:                      LCP1 := LCP; INSYMBOL                                    PASCP  984
1059:                    END                                                        PASCP  985
1060:                  ELSE ERROR(2);                                               PASCP  986
1061:                  IF NOT (SY IN FSYS + [COMMA,RPARENT]) THEN                   PASCP  987
1062:                    BEGIN ERROR(6); SKIP(FSYS + [COMMA,RPARENT]) END           PASCP  988
1063:                UNTIL SY <> COMMA;                                             PASCP  989
1064:                LSP^.FCONST := LCP1; TOP := TTOP;                              PASCP  990
1065:                IF SY = RPARENT THEN INSYMBOL ELSE ERROR(4)                    PASCP  991
1066:              END                                                              PASCP  992
1067:            ELSE                                                               PASCP  993
1068:              BEGIN                                                            PASCP  994
1069:                IF SY = IDENT THEN                                             PASCP  995
1070:                  BEGIN SEARCHID([TYPES,KONST],LCP);                           PASCP  996
1071:                    INSYMBOL;                                                  PASCP  997
1072:                    IF LCP^.KLASS = KONST THEN                                 PASCP  998
1073:                      BEGIN NEW(LSP,SUBRANGE);                                 PASCP  999
1074:                        WITH LSP^, LCP^ DO                                     PASCP 1000
1075:                          BEGIN RANGETYPE := IDTYPE; FORM := SUBRANGE;         PASCP 1001
1076:                            IF STRING(RANGETYPE) THEN                          PASCP 1002
1077:                              BEGIN ERROR(148); RANGETYPE := NIL END;          PASCP 1003
1078:                            MIN := VALUES; SIZE := INTSIZE                     PASCP 1004
1079:                          END;                                                 PASCP 1005
1080:                        IF SY = COLON THEN INSYMBOL ELSE ERROR(5);             PASCP 1006
1081:                        CONSTANT(FSYS,LSP1,LVALU);                             PASCP 1007
1082:                        LSP^.MAX := LVALU;                                     PASCP 1008
1083:                        IF LSP^.RANGETYPE <> LSP1 THEN ERROR(107)              PASCP 1009
1084:                      END                                                      PASCP 1010
1085:                    ELSE                                                       PASCP 1011
1086:                      BEGIN LSP := LCP^.IDTYPE;                                PASCP 1012
1087:                        IF LSP <> NIL THEN FSIZE := LSP^.SIZE                  PASCP 1013
1088:                      END                                                      PASCP 1014
1089:                  END (*SY = IDENT*)                                           PASCP 1015
1090:                ELSE                                                         PASCP 1016
1091:                  BEGIN NEW(LSP,SUBRANGE); LSP^.FORM := SUBRANGE;            PASCP 1017
1092:                    CONSTANT(FSYS + [COLON],LSP1,LVALU);                       PASCP 1018
1093:                    IF STRING(LSP1) THEN                                       PASCP 1019
1094:                      BEGIN ERROR(148); LSP1 := NIL END;                       PASCP 1020
1095:                    WITH LSP^ DO                                               PASCP 1021
1096:                      BEGIN RANGETYPE:=LSP1; MIN:=LVALU; SIZE:=INTSIZE END;    PASCP 1022
1097:                    IF SY = COLON THEN INSYMBOL ELSE ERROR(5);                 PASCP 1023
1098:                    CONSTANT(FSYS,LSP1,LVALU);                                 PASCP 1024
1099:                    LSP^.MAX := LVALU;                                         PASCP 1025
1100:                    IF LSP^.RANGETYPE <> LSP1 THEN ERROR(107)                  PASCP 1026
1101:                  END;                                                         PASCP 1027
1102:                IF LSP <> NIL THEN                                           PASCP 1028
1103:                  WITH LSP^ DO                                               PASCP 1029
1104:                    IF FORM = SUBRANGE THEN                                  PASCP 1030
1105:                      IF RANGETYPE <> NIL THEN                               PASCP 1031
```

```
1106:                         IF RANGETYPE = REALPTR THEN ERROR(399)                  PASCP 1032
1107:                         ELSE                                                    PASCP 1033
1108:                           IF MIN.IVAL > MAX.IVAL THEN ERROR(102)                PASCP 1034
1109:               END;                                                              PASCP 1035
1110:             FSP := LSP;                                                         PASCP 1036
1111:             IF NOT (SY IN FSYS) THEN                                            PASCP 1037
1112:               BEGIN ERROR(6); SKIP(FSYS) END                                    PASCP 1038
1113:           END                                                                   PASCP 1039
1114:             ELSE FSP := NIL                                                     PASCP 1040
1115:         END (*SIMPLETYPE*) ;                                                    PASCP 1041
1116:                                                                                 PASCP 1042
1117:         PROCEDURE FIELDLIST(FSYS: SETOFSYS; VAR FRECVAR: STP);                  PASCP 1043
1118:           VAR LCP,LCP1,NXT,NXT1: CTP; LSP,LSP1,LSP2,LSP3,LSP4: STP;             PASCP 1044
1119:               MINSIZE,MAXSIZE,LSIZE: ADDRRANGE; LVALU: VALU;                    PASCP 1045
1120:         BEGIN NXT1 := NIL; LSP := NIL;                                          PASCP 1046
1121:           IF NOT (SY IN (FSYS+[IDENT,CASESY])) THEN                             X2       1
1122:             BEGIN ERROR(19); SKIP(FSYS + [IDENT,CASESY]) END;                   PASCP 1048
1123:           WHILE SY = IDENT DO                                                   PASCP 1049
1124:             BEGIN NXT := NXT1;                                                  PASCP 1050
1125:               REPEAT                                                            PASCP 1051
1126:                 IF SY = IDENT THEN                                              PASCP 1052
1127:                   BEGIN NEW(LCP,FIELD);                                         PASCP 1053
1128:                     WITH LCP^ DO                                                PASCP 1054
1129:                       BEGIN NAME := ID; IDTYPE := NIL; NEXT := NXT;             PASCP 1055
1130:                         KLASS := FIELD                                          PASCP 1056
1131:                       END;                                                      PASCP 1057
1132:                     NXT := LCP;                                                 PASCP 1058
1133:                     ENTERID(LCP);                                               PASCP 1059
1134:                     INSYMBOL                                                    PASCP 1060
1135:                   END                                                           PASCP 1061
1136:                 ELSE ERROR(2);                                                  PASCP 1062
1137:                 IF NOT (SY IN [COMMA,COLON]) THEN                               PASCP 1063
1138:                   BEGIN ERROR(6); SKIP(FSYS + [COMMA,COLON,SEMICOLON,CASESY])   PASCP 1064
1139:                   END;                                                          PASCP 1065
1140:               TEST := SY <> COMMA;                                              PASCP 1066
1141:                 IF NOT TEST  THEN INSYMBOL                                      PASCP 1067
1142:               UNTIL TEST;                                                       PASCP 1068
1143:               IF SY = COLON THEN INSYMBOL ELSE ERROR(5);                        PASCP 1069
1144:               TYP(FSYS + [CASESY,SEMICOLON],LSP,LSIZE);                         PASCP 1070
1145:               WHILE NXT <> NXT1 DO                                              PASCP 1071
1146:                 WITH NXT^ DO                                                    PASCP 1072
1147:                   BEGIN ALIGN(LSP,DISPL);                                       X3       1
1148:                     IDTYPE := LSP; FLDADDR := DISPL;                            X3       2
1149:                     NXT := NEXT; DISPL := DISPL + LSIZE                         PASCP 1074
1150:                   END;                                                          PASCP 1075
1151:               NXT1 := LCP;                                                      PASCP 1076
1152:               IF SY = SEMICOLON THEN                                            PASCP 1077
1153:                 BEGIN INSYMBOL;                                                 PASCP 1078
1154:                   IF NOT (SY IN [IDENT,CASESY]) THEN                            PASCP 1079
1155:                     BEGIN ERROR(19); SKIP(FSYS + [IDENT,CASESY]) END            PASCP 1080
1156:                 END                                                           PASCP 1081
1157:             END (*WHILE*);                                                      PASCP 1082
1158:           NXT := NIL;                                                           PASCP 1083
1159:           WHILE NXT1 <> NIL DO                                                  PASCP 1084
1160:             WITH NXT1^ DO                                                       PASCP 1085
1161:               BEGIN LCP := NEXT; NEXT := NXT; NXT := NXT1; NXT1 := LCP END;     PASCP 1086
1162:           IF SY = CASESY THEN                                                   PASCP 1087
1163:             BEGIN NEW(LSP,TAGFLD);                                              PASCP 1088
1164:               WITH LSP^ DO                                                      PASCP 1089
1165:                 BEGIN TAGFIELDP := NIL; FSTVAR := NIL; FORM:=TAGFLD END;        PASCP 1090
1166:               FRECVAR := LSP;                                                   PASCP 1091
1167:               INSYMBOL;                                                         PASCP 1092
1168:               IF SY = IDENT THEN                                                PASCP 1093
1169:                 BEGIN NEW(LCP,FIELD);                                           PASCP 1094
1170:                   WITH LCP^ DO                                                  PASCP 1095
```

```
1171:                   BEGIN NAME := ID; IDTYPE := NIL; KLASS:=FIELD;                PASCP 1096
1172:                     NEXT := NIL; FLDADDR := DISPL                               PASCP 1097
1173:                   END;                                                          PASCP 1098
1174:                 ENTERID(LCP);                                                   PASCP 1099
1175:                 INSYMBOL;                                                       PASCP 1100
1176:                 IF SY = COLON THEN INSYMBOL ELSE ERROR(5);                      PASCP 1101
1177:                 IF SY = IDENT THEN                                              PASCP 1102
1178:                   BEGIN SEARCHID([TYPES],LCP1);                                 PASCP 1103
1179:                     LSP1 := LCP1^.IDTYPE;                                       PASCP 1104
1180:                     IF LSP1 <> NIL THEN                                         PASCP 1105
1181:                       BEGIN ALIGN(LCP^.IDTYPE,DISPL);                           P      147
1182:                         LCP^.FLDADDR := DISPL;                                  P      148
1183:                         DISPL := DISPL+LSP1^.SIZE;                              P      149
1184:                         IF (LSP1^.FORM <= SUBRANGE) OR STRING(LSP1) THEN        PASCP 1107
1185:                           BEGIN IF COMPTYPES(REALPTR,LSP1) THEN ERROR(109)      PASCP 1108
1186:                             ELSE IF STRING(LSP1) THEN ERROR(399);               PASCP 1109
1187:                             LCP^.IDTYPE := LSP1; LSP^.TAGFIELDP := LCP;         PASCP 1110
1188:                           END                                                   PASCP 1111
1189:                         ELSE ERROR(110);                                        PASCP 1112
1190:                     END;                                                        PASCP 1113
1191:                     INSYMBOL;                                                   PASCP 1114
1192:                   END                                                           PASCP 1115
1193:                 ELSE BEGIN ERROR(2); SKIP(FSYS + [OFSY,LPARENT]) END            PASCP 1116
1194:               END                                                               PASCP 1117
1195:             ELSE BEGIN ERROR(2); SKIP(FSYS + [OFSY,LPARENT]) END;               PASCP 1118
1196:             LSP^.SIZE := DISPL;                                                 PASCP 1119
1197:             IF SY = OFSY THEN INSYMBOL ELSE ERROR(8);                           PASCP 1120
1198:             LSP1 := NIL; MINSIZE := DISPL; MAXSIZE := DISPL;                    PASCP 1121
1199:             REPEAT LSP2 := NIL;                                                 PASCP 1122
1200:               IF NOT (SY IN [SEMICOLON,ENDSY]) THEN                             P      150
1201:               BEGIN                                                             P      151
1202:               REPEAT CONSTANT(FSYS + [COMMA,COLON,LPARENT],LSP3,LVALU);         PASCP 1123
1203:                 IF LSP^.TAGFIELDP <> NIL THEN                                   PASCP 1124
1204:                  IF NOT COMPTYPES(LSP^.TAGFIELDP^.IDTYPE,LSP3)THEN ERROR(111);  PASCP 1125
1205:                 NEW(LSP3,VARIANT);                                              PASCP 1126
1206:                 WITH LSP3^ DO                                                   PASCP 1127
1207:                   BEGIN NXTVAR := LSP1; SUBVAR := LSP2; VARVAL := LVALU;        PASCP 1128
1208:                     FORM := VARIANT                                             PASCP 1129
1209:                   END;                                                          PASCP 1130
1210:                 LSP4 := LSP1;                                                   P      152
1211:                 WHILE LSP4 <> NIL DO                                            P      153
1212:                   WITH LSP4^ DO                                                 P      154
1213:                     BEGIN                                                       P      155
1214:                       IF VARVAL.IVAL = LVALU.IVAL THEN ERROR(178);              P      156
1215:                       LSP4 := NXTVAR                                            P      157
1216:                     END;                                                        P      158
1217:                 LSP1 := LSP3; LSP2 := LSP3;                                     PASCP 1131
1218:                 TEST := SY <> COMMA;                                            PASCP 1132
1219:                 IF NOT TEST THEN INSYMBOL                                       PASCP 1133
1220:               UNTIL TEST;                                                       PASCP 1134
1221:               IF SY = COLON THEN INSYMBOL ELSE ERROR(5);                        PASCP 1135
1222:               IF SY = LPARENT THEN INSYMBOL ELSE ERROR(9);                      PASCP 1136
1223:               FIELDLIST(FSYS + [RPARENT,SEMICOLON],LSP2);                       PASCP 1137
1224:               IF DISPL > MAXSIZE THEN MAXSIZE := DISPL;                         PASCP 1138
1225:               WHILE LSP3 <> NIL DO                                              PASCP 1139
1226:                 BEGIN LSP4 := LSP3^.SUBVAR; LSP3^.SUBVAR := LSP2;               PASCP 1140
1227:                   LSP3^.SIZE := DISPL;                                          PASCP 1141
1228:                   LSP3 := LSP4                                                  PASCP 1142
1229:                 END;                                                            PASCP 1143
1230:               IF SY = RPARENT THEN                                              PASCP 1144
1231:                 BEGIN INSYMBOL;                                                 PASCP 1145
1232:                   IF NOT (SY IN FSYS + [SEMICOLON]) THEN                        PASCP 1146
1233:                     BEGIN ERROR(6); SKIP(FSYS + [SEMICOLON]) END                PASCP 1147
1234:                 END                                                           PASCP 1148
1235:               ELSE ERROR(4);                                                    PASCP 1149
```

```
1236:               END;                                                                P      159
1237:               TEST := SY <> SEMICOLON;                                            PASCP 1150
1238:               IF NOT TEST THEN                                                    PASCP 1151
1239:                 BEGIN DISPL := MINSIZE;                                           PASCP 1152
1240:                       INSYMBOL                                                    PASCP 1153
1241:                 END                                                             PASCP 1154
1242:             UNTIL TEST;                                                         PASCP 1155
1243:             DISPL := MAXSIZE;                                                   PASCP 1156
1244:             LSP^.FSTVAR := LSP1;                                                PASCP 1157
1245:           END                                                                   PASCP 1158
1246:         ELSE FRECVAR := NIL                                                     PASCP 1159
1247:       END (*FIELDLIST*) ;                                                       PASCP 1160
1248:                                                                                 PASCP 1161
1249:     BEGIN (*TYP*)                                                               PASCP 1162
1250:       IF NOT (SY IN TYPEBEGSYS) THEN                                            PASCP 1163
1251:          BEGIN ERROR(10); SKIP(FSYS + TYPEBEGSYS) END;                          PASCP 1164
1252:       IF SY IN TYPEBEGSYS THEN                                                  PASCP 1165
1253:         BEGIN                                                                   PASCP 1166
1254:           IF SY IN SIMPTYPEBEGSYS THEN SIMPLETYPE(FSYS,FSP,FSIZE)               PASCP 1167
1255:           ELSE                                                                  PASCP 1168
1256:     (*^*)     IF SY = ARROW THEN                                              PASCP 1169
1257:               BEGIN NEW(LSP,POINTER); FSP := LSP;                             PASCP 1170
1258:                 WITH LSP^ DO                                                  PASCP 1171
1259:                   BEGIN ELTYPE := NIL; SIZE := PTRSIZE; FORM:=POINTER END;    PASCP 1172
1260:                 INSYMBOL;                                                     PASCP 1173
1261:                 IF SY = IDENT THEN                                            PASCP 1174
1262:                   BEGIN PRTERR := FALSE; (*NO ERROR IF SEARCH NOT SUCCESSFUL*)PASCP 1175
1263:                     SEARCHID([TYPES],LCP); PRTERR := TRUE;                    PASCP 1176
1264:                     IF LCP = NIL THEN   (*FORWARD REFERENCED TYPE ID*)        PASCP 1177
1265:                       BEGIN NEW(LCP,TYPES);                                   PASCP 1178
1266:                         WITH LCP^ DO                                          PASCP 1179
1267:                           BEGIN NAME := ID; IDTYPE := LSP;                    PASCP 1180
1268:                             NEXT := FWPTR; KLASS := TYPES                     PASCP 1181
1269:                           END;                                                PASCP 1182
1270:                         FWPTR := LCP                                          PASCP 1183
1271:                       END                                                     PASCP 1184
1272:                     ELSE                                                      PASCP 1185
1273:                       BEGIN                                                   PASCP 1186
1274:                         IF LCP^.IDTYPE <> NIL THEN                            PASCP 1187
1275:                           IF LCP^.IDTYPE^.FORM = FILES THEN ERROR(108)        PASCP 1188
1276:                           ELSE LSP^.ELTYPE := LCP^.IDTYPE                     PASCP 1189
1277:                       END;                                                    PASCP 1190
1278:                     INSYMBOL;                                                 PASCP 1191
1279:                   END                                                       PASCP 1192
1280:                 ELSE ERROR(2);                                              PASCP 1193
1281:               END                                                           PASCP 1194
1282:             ELSE                                                            PASCP 1195
1283:               BEGIN                                                         PASCP 1196
1284:                 IF SY = PACKEDSY THEN                                       PASCP 1197
1285:                   BEGIN INSYMBOL;                                           PASCP 1198
1286:                     IF NOT (SY IN TYPEDELS) THEN                            PASCP 1199
1287:                       BEGIN                                                 PASCP 1200
1288:                         ERROR(10); SKIP(FSYS + TYPEDELS)                    PASCP 1201
1289:                       END                                                   PASCP 1202
1290:                   END;                                                      PASCP 1203
1291:     (*ARRAY*)     IF SY = ARRAYSY THEN                                      PASCP 1204
1292:                   BEGIN INSYMBOL;                                           PASCP 1205
1293:                     IF SY = LBRACK THEN INSYMBOL ELSE ERROR(11);            PASCP 1206
1294:                     LSP1 := NIL;                                            PASCP 1207
1295:                     REPEAT NEW(LSP,ARRAYS);                                 PASCP 1208
1296:                       WITH LSP^ DO                                          PASCP 1209
1297:                         BEGIN AELTYPE := LSP1; INXTYPE := NIL; FORM:=ARRAYS END;PASCP 1210
1298:                       LSP1 := LSP;                                          PASCP 1211
1299:                       SIMPLETYPE(FSYS + [COMMA,RBRACK,OFSY],LSP2,LSIZE);    PASCP 1212
1300:                       LSP1^.SIZE := LSIZE;                                  PASCP 1213
```

```
1301:                     IF LSP2 <> NIL THEN                                          PASCP 1214
1302:                       IF LSP2^.FORM <= SUBRANGE THEN                             PASCP 1215
1303:                         BEGIN                                                    PASCP 1216
1304:                           IF LSP2 = REALPTR THEN                                 PASCP 1217
1305:                             BEGIN ERROR(109); LSP2 := NIL END                    PASCP 1218
1306:                           ELSE                                                   PASCP 1219
1307:                             IF LSP2 = INTPTR THEN                                PASCP 1220
1308:                               BEGIN ERROR(149); LSP2 := NIL END;                 PASCP 1221
1309:                           LSP^.INXTYPE := LSP2                                   PASCP 1222
1310:                         END                                                      PASCP 1223
1311:                       ELSE BEGIN ERROR(113); LSP2 := NIL END;                    PASCP 1224
1312:                     TEST := SY <> COMMA;                                         PASCP 1225
1313:                     IF NOT TEST THEN INSYMBOL                                    PASCP 1226
1314:                   UNTIL TEST;                                                    PASCP 1227
1315:                   IF SY = RBRACK THEN INSYMBOL ELSE ERROR(12);                   PASCP 1228
1316:                   IF SY = OFSY THEN INSYMBOL ELSE ERROR(8);                      PASCP 1229
1317:                   TYP(FSYS,LSP,LSIZE);                                           PASCP 1230
1318:                   REPEAT                                                         PASCP 1231
1319:                     WITH LSP1^ DO                                                PASCP 1232
1320:                       BEGIN LSP2 := AELTYPE; AELTYPE := LSP;                     PASCP 1233
1321:                         IF INXTYPE <> NIL THEN                                   PASCP 1234
1322:                           BEGIN GETBOUNDS(INXTYPE,LMIN,LMAX);                    PASCP 1235
1323:                             ALIGN(LSP,LSIZE);                                    P      160
1324:                             LSIZE := LSIZE*(LMAX - LMIN + 1);                    PASCP 1236
1325:                             SIZE := LSIZE                                        PASCP 1237
1326:                           END                                                    PASCP 1238
1327:                       END;                                                       PASCP 1239
1328:                     LSP := LSP1; LSP1 := LSP2                                    PASCP 1240
1329:                   UNTIL LSP1 = NIL                                               PASCP 1241
1330:                 END                                                              PASCP 1242
1331:               ELSE                                                               PASCP 1243
1332:     (*RECORD*)    IF SY = RECORDSY THEN                                          PASCP 1244
1333:                   BEGIN INSYMBOL;                                                PASCP 1245
1334:                     OLDTOP := TOP;                                               PASCP 1246
1335:                     IF TOP < DISPLIMIT THEN                                      PASCP 1247
1336:                       BEGIN TOP := TOP + 1;                                      PASCP 1248
1337:                         WITH DISPLAY[TOP] DO                                     PASCP 1249
1338:                           BEGIN FNAME := NIL;                                    PASCP 1250
1339:                             FLABEL := NIL;                                       PASCP 1251
1340:                                 OCCUR := REC                                     PASCP 1252
1341:                           END                                                    PASCP 1253
1342:                       END                                                        PASCP 1254
1343:                     ELSE ERROR(250);                                             PASCP 1255
1344:                     DISPL := 0;                                                  PASCP 1256
1345:                     FIELDLIST(FSYS-[SEMICOLON]+[ENDSY],LSP1);                    PASCP 1257
1346:                     NEW(LSP,RECORDS);                                            PASCP 1258
1347:                     WITH LSP^ DO                                                 PASCP 1259
1348:                       BEGIN FSTFLD := DISPLAY[TOP].FNAME;                        PASCP 1260
1349:                         RECVAR := LSP1; SIZE := DISPL; FORM := RECORDS           PASCP 1261
1350:                       END;                                                       PASCP 1262
1351:                     TOP := OLDTOP;                                               PASCP 1263
1352:                     IF SY = ENDSY THEN INSYMBOL ELSE ERROR(13)                   PASCP 1264
1353:                   END                                                            PASCP 1265
1354:                 ELSE                                                             PASCP 1266
1355:     (*SET*)       IF SY = SETSY THEN                                             PASCP 1267
1356:                   BEGIN INSYMBOL;                                                PASCP 1268
1357:                     IF SY = OFSY THEN INSYMBOL ELSE ERROR(8);                    PASCP 1269
1358:                     SIMPLETYPE(FSYS,LSP1,LSIZE);                                 PASCP 1270
1359:                     IF LSP1 <> NIL THEN                                          PASCP 1271
1360:                       IF LSP1^.FORM > SUBRANGE THEN                              PASCP 1272
1361:                         BEGIN ERROR(115); LSP1 := NIL END                        PASCP 1273
1362:                       ELSE                                                       PASCP 1274
1363:                         IF LSP1 = REALPTR THEN ERROR(114);                       PASCP 1275
1364:                     NEW(LSP,POWER);                                              PASCP 1276
1365:                     WITH LSP^ DO                                                 PASCP 1277
```

```
1366:                           BEGIN ELSET:=LSP1; SIZE:=SETSIZE; FORM:=POWER END;    PASCP 1278
1367:                       END                                                       PASCP 1279
1368:                     ELSE                                                        PASCP 1280
1369:       (*FILE*)          IF SY = FILESY THEN                                     PASCP 1281
1370:                           BEGIN INSYMBOL;                                       P      161
1371:                             ERROR(399); SKIP(FSYS); LSP := NIL                  P      162
1372:                           END;                                                  P      163
1373:                 FSP := LSP                                                      PASCP 1283
1374:               END;                                                              PASCP 1284
1375:             IF NOT (SY IN FSYS) THEN                                            PASCP 1285
1376:               BEGIN ERROR(6); SKIP(FSYS) END                                    PASCP 1286
1377:           END                                                                   PASCP 1287
1378:         ELSE FSP := NIL;                                                        PASCP 1288
1379:         IF FSP = NIL THEN FSIZE := 1 ELSE FSIZE := FSP^.SIZE                    PASCP 1289
1380:       END (*TYP*) ;                                                             PASCP 1290
1381:                                                                                 PASCP 1291
1382:       PROCEDURE LABELDECLARATION;                                               PASCP 1292
1383:         VAR LLP: LBP; REDEF: BOOLEAN; LBNAME: INTEGER;                          PASCP 1293
1384:       BEGIN                                                                     PASCP 1294
1385:         REPEAT                                                                  PASCP 1295
1386:           IF SY = INTCONST THEN                                                 PASCP 1296
1387:             WITH DISPLAY[TOP] DO                                                PASCP 1297
1388:               BEGIN LLP := FLABEL; REDEF := FALSE;                              PASCP 1298
1389:                 WHILE (LLP <> NIL) AND NOT REDEF DO                             PASCP 1299
1390:                   IF LLP^.LABVAL <> VAL.IVAL THEN                               PASCP 1300
1391:                     LLP := LLP^.NEXTLAB                                         PASCP 1301
1392:                   ELSE BEGIN REDEF := TRUE; ERROR(166) END;                     PASCP 1302
1393:                 IF NOT REDEF THEN                                               PASCP 1303
1394:                   BEGIN NEW(LLP);                                               PASCP 1304
1395:                     WITH LLP^ DO                                                PASCP 1305
1396:                       BEGIN LABVAL := VAL.IVAL; GENLABEL(LBNAME);               PASCP 1306
1397:                         DEFINED := FALSE; NEXTLAB := FLABEL; LABNAME := LBNAME  PASCP 1307
1398:                       END;                                                      PASCP 1308
1399:                     FLABEL := LLP                                               PASCP 1309
1400:                   END;                                                          PASCP 1310
1401:                 INSYMBOL                                                        PASCP 1311
1402:               END                                                               PASCP 1312
1403:           ELSE ERROR(15);                                                       PASCP 1313
1404:           IF NOT ( SY IN FSYS + [COMMA, SEMICOLON] ) THEN                       PASCP 1314
1405:             BEGIN ERROR(6); SKIP(FSYS+[COMMA,SEMICOLON]) END;                   PASCP 1315
1406:           TEST := SY <> COMMA;                                                  PASCP 1316
1407:           IF NOT TEST THEN INSYMBOL                                             PASCP 1317
1408:         UNTIL TEST;                                                             PASCP 1318
1409:         IF SY = SEMICOLON THEN INSYMBOL ELSE ERROR(14)                          PASCP 1319
1410:       END (* LABELDECLARATION *) ;                                              PASCP 1320
1411:                                                                                 PASCP 1321
1412:       PROCEDURE CONSTDECLARATION;                                               PASCP 1322
1413:         VAR LCP: CTP; LSP: STP; LVALU: VALU;                                    PASCP 1323
1414:       BEGIN                                                                     PASCP 1324
1415:         IF SY <> IDENT THEN                                                     PASCP 1325
1416:           BEGIN ERROR(2); SKIP(FSYS + [IDENT]) END;                             PASCP 1326
1417:         WHILE SY = IDENT DO                                                     PASCP 1327
1418:           BEGIN NEW(LCP,KONST);                                                 PASCP 1328
1419:             WITH LCP^ DO                                                        PASCP 1329
1420:               BEGIN NAME := ID; IDTYPE := NIL; NEXT := NIL; KLASS:=KONST END;    PASCP 1330
1421:             INSYMBOL;                                                           PASCP 1331
1422:             IF (SY = RELOP) AND (OP = EQOP) THEN INSYMBOL ELSE ERROR(16);       PASCP 1332
1423:             CONSTANT(FSYS + [SEMICOLON],LSP,LVALU);                             PASCP 1333
1424:             ENTERID(LCP);                                                       PASCP 1334
1425:             LCP^.IDTYPE := LSP; LCP^.VALUES := LVALU;                           PASCP 1335
1426:             IF SY = SEMICOLON THEN                                              PASCP 1336
1427:               BEGIN INSYMBOL;                                                   PASCP 1337
1428:                 IF NOT (SY IN FSYS + [IDENT]) THEN                              PASCP 1338
1429:                   BEGIN ERROR(6); SKIP(FSYS + [IDENT]) END                      PASCP 1339
1430:               END                                                               PASCP 1340
```

```
1431:           ELSE ERROR(14)                                                  PASCP 1341
1432:         END                                                               PASCP 1342
1433:       END (*CONSTDECLARATION*) ;                                          PASCP 1343
1434:                                                                           PASCP 1344
1435:       PROCEDURE TYPEDECLARATION;                                          PASCP 1345
1436:         VAR LCP,LCP1,LCP2: CTP; LSP: STP; LSIZE: ADDRRANGE;               PASCP 1346
1437:       BEGIN                                                               PASCP 1347
1438:         IF SY <> IDENT THEN                                               PASCP 1348
1439:           BEGIN ERROR(2); SKIP(FSYS + [IDENT]) END;                       PASCP 1349
1440:         WHILE SY = IDENT DO                                               PASCP 1350
1441:           BEGIN NEW(LCP,TYPES);                                           PASCP 1351
1442:             WITH LCP^ DO                                                  PASCP 1352
1443:               BEGIN NAME := ID; IDTYPE := NIL; KLASS := TYPES END;        PASCP 1353
1444:             INSYMBOL;                                                     PASCP 1354
1445:             IF (SY = RELOP) AND (OP = EQOP) THEN INSYMBOL ELSE ERROR(16); PASCP 1355
1446:             TYP(FSYS + [SEMICOLON],LSP,LSIZE);                            PASCP 1356
1447:             ENTERID(LCP);                                                 PASCP 1357
1448:             LCP^.IDTYPE := LSP;                                           PASCP 1358
1449:             (*HAS ANY FORWARD REFERENCE BEEN SATISFIED:*)                  PASCP 1359
1450:             LCP1 := FWPTR;                                                PASCP 1360
1451:             WHILE LCP1 <> NIL DO                                          PASCP 1361
1452:               BEGIN                                                       PASCP 1362
1453:                 IF LCP1^.NAME = LCP^.NAME THEN                            PASCP 1363
1454:                   BEGIN LCP1^.IDTYPE^.ELTYPE := LCP^.IDTYPE;              PASCP 1364
1455:                     IF LCP1 <> FWPTR THEN                                 PASCP 1365
1456:                       LCP2^.NEXT := LCP1^.NEXT                            PASCP 1366
1457:                     ELSE FWPTR := LCP1^.NEXT;                             PASCP 1367
1458:                   END;                                                    PASCP 1368
1459:                 LCP2 := LCP1; LCP1 := LCP1^.NEXT                          PASCP 1369
1460:               END;                                                        PASCP 1370
1461:             IF SY = SEMICOLON THEN                                        PASCP 1371
1462:               BEGIN INSYMBOL;                                             PASCP 1372
1463:                 IF NOT (SY IN FSYS + [IDENT]) THEN                        PASCP 1373
1464:                   BEGIN ERROR(6); SKIP(FSYS + [IDENT]) END                PASCP 1374
1465:               END                                                         PASCP 1375
1466:             ELSE ERROR(14)                                                PASCP 1376
1467:           END;                                                            PASCP 1377
1468:         IF FWPTR <> NIL THEN                                              PASCP 1378
1469:           BEGIN ERROR(117); WRITELN(OUTPUT);                              PASCP 1379
1470:             REPEAT WRITELN(OUTPUT,' TYPE-ID ',FWPTR^.NAME);               PASCP 1380
1471:               FWPTR := FWPTR^.NEXT                                        PASCP 1381
1472:             UNTIL FWPTR = NIL;                                            PASCP 1382
1473:             IF NOT EOL THEN WRITE(OUTPUT,' ': CHCNT+16)                   PASCP 1383
1474:           END                                                             PASCP 1384
1475:       END (*TYPEDECLARATION*) ;                                           PASCP 1385
1476:                                                                           PASCP 1386
1477:       PROCEDURE VARDECLARATION;                                           PASCP 1387
1478:         VAR LCP,NXT: CTP; LSP: STP; LSIZE: ADDRRANGE;                     PASCP 1388
1479:       BEGIN NXT := NIL;                                                   PASCP 1389
1480:         REPEAT                                                            PASCP 1390
1481:           REPEAT                                                          PASCP 1391
1482:             IF SY = IDENT THEN                                            PASCP 1392
1483:               BEGIN NEW(LCP,VARS);                                        PASCP 1393
1484:                 WITH LCP^ DO                                              PASCP 1394
1485:                  BEGIN NAME := ID; NEXT := NXT; KLASS := VARS;            PASCP 1395
1486:                     IDTYPE := NIL; VKIND := ACTUAL; VLEV := LEVEL         PASCP 1396
1487:                  END;                                                     PASCP 1397
1488:                 ENTERID(LCP);                                             PASCP 1398
1489:                 NXT := LCP;                                               PASCP 1399
1490:                 INSYMBOL;                                                 PASCP 1400
1491:               END                                                         PASCP 1401
1492:             ELSE ERROR(2);                                                PASCP 1402
1493:             IF NOT (SY IN FSYS + [COMMA,COLON] + TYPEDELS) THEN           PASCP 1403
1494:               BEGIN ERROR(6); SKIP(FSYS+[COMMA,COLON,SEMICOLON]+TYPEDELS) END; PASCP 1404
1495:             TEST := SY <> COMMA;                                          PASCP 1405
```

```
1496:           IF NOT TEST THEN INSYMBOL                                                PASCP 1406
1497:         UNTIL TEST;                                                                PASCP 1407
1498:         IF SY = COLON THEN INSYMBOL ELSE ERROR(5);                                 PASCP 1408
1499:         TYP(FSYS + [SEMICOLON] + TYPEDELS,LSP,LSIZE);                              PASCP 1409
1500:         WHILE NXT <> NIL DO                                                        PASCP 1410
1501:           WITH  NXT^ DO                                                            PASCP 1411
1502:             BEGIN ALIGN(LSP,LC);                                                   P      164
1503:               IDTYPE := LSP; VADDR := LC;                                          P      165
1504:               LC := LC + LSIZE; NXT := NEXT                                        PASCP 1413
1505:             END;                                                                   PASCP 1414
1506:         IF SY = SEMICOLON THEN                                                     PASCP 1415
1507:           BEGIN INSYMBOL;                                                          PASCP 1416
1508:             IF NOT (SY IN FSYS + [IDENT]) THEN                                     PASCP 1417
1509:               BEGIN ERROR(6); SKIP(FSYS + [IDENT]) END                             PASCP 1418
1510:           END                                                                      PASCP 1419
1511:         ELSE ERROR(14)                                                             PASCP 1420
1512:       UNTIL (SY <> IDENT) AND NOT (SY IN TYPEDELS);                                PASCP 1421
1513:       IF FWPTR <> NIL THEN                                                         PASCP 1422
1514:         BEGIN ERROR(117); WRITELN(OUTPUT);                                         PASCP 1423
1515:           REPEAT WRITELN(OUTPUT,' TYPE-ID ',FWPTR^.NAME);                          PASCP 1424
1516:             FWPTR := FWPTR^.NEXT                                                   PASCP 1425
1517:           UNTIL FWPTR = NIL;                                                       PASCP 1426
1518:           IF NOT EOL THEN WRITE(OUTPUT,' ': CHCNT+16)                              PASCP 1427
1519:         END                                                                        PASCP 1428
1520:     END (*VARDECLARATION*) ;                                                       PASCP 1429
1521:                                                                                    PASCP 1430
1522:     PROCEDURE PROCDECLARATION(FSY: SYMBOL);                                        PASCP 1431
1523:       VAR OLDLEV: 0..MAXLEVEL; LSY: SYMBOL; LCP,LCP1: CTP; LSP: STP;               PASCP 1432
1524:           FORW: BOOLEAN; OLDTOP: DISPRANGE; PARCNT: INTEGER;                       PASCP 1433
1525:           LLC,LCM: ADDRRANGE; LBNAME: INTEGER; MARKP: ^INTEGER;                    PASCP 1434
1526:                                                                                    PASCP 1435
1527:       PROCEDURE PARAMETERLIST(FSY: SETOFSYS; VAR FPAR: CTP);                       PASCP 1436
1528:         VAR LCP,LCP1,LCP2,LCP3: CTP; LSP: STP; LKIND: IDKIND;                      PASCP 1437
1529:           LLC: ADDRRANGE; COUNT,LSIZE: INTEGER;                                    P      166
1530:       BEGIN LCP1 := NIL;                                                           PASCP 1439
1531:         IF NOT (SY IN FSY + [LPARENT]) THEN                                        PASCP 1440
1532:           BEGIN ERROR(7); SKIP(FSYS + FSY + [LPARENT]) END;                        PASCP 1441
1533:         IF SY = LPARENT THEN                                                       PASCP 1442
1534:           BEGIN IF FORW THEN ERROR(119);                                           PASCP 1443
1535:             INSYMBOL;                                                              PASCP 1444
1536:             IF NOT (SY IN [IDENT,VARSY,PROCSY,FUNCSY]) THEN                        PASCP 1445
1537:               BEGIN ERROR(7); SKIP(FSYS + [IDENT,RPARENT]) END;                    PASCP 1446
1538:             WHILE SY IN [IDENT,VARSY,PROCSY,FUNCSY] DO                             PASCP 1447
1539:               BEGIN                                                                PASCP 1448
1540:                 IF SY = PROCSY THEN                                                PASCP 1449
1541:                   BEGIN ERROR(399);                                                PASCP 1450
1542:                     REPEAT INSYMBOL;                                               PASCP 1451
1543:                       IF SY = IDENT THEN                                           PASCP 1452
1544:                       BEGIN NEW(LCP,PROC,DECLARED,FORMAL);                         PASCP 1453
1545:                           WITH LCP^ DO                                             PASCP 1454
1546:                             BEGIN NAME := ID; IDTYPE := NIL; NEXT := LCP1;         PASCP 1455
1547:                               PFLEV := LEVEL (*BEWARE OF PARAMETER PROCEDURES*);PASCP 1456
1548:                               KLASS:=PROC;PFDECKIND:=DECLARED;PFKIND:=FORMAL       PASCP 1457
1549:                             END;                                                   PASCP 1458
1550:                           ENTERID(LCP);                                            PASCP 1459
1551:                           LCP1 := LCP;                                             P      167
1552:                           ALIGN(PARMPTR,LC);                                       P      168
1553:                           (*LC := LC + SOME SIZE *)                                P      169
1554:                           INSYMBOL                                                 PASCP 1461
1555:                       END                                                          PASCP 1462
1556:                       ELSE ERROR(2);                                               PASCP 1463
1557:                       IF NOT (SY IN FSYS + [COMMA,SEMICOLON,RPARENT]) THEN         PASCP 1464
1558:                         BEGIN ERROR(7);SKIP(FSYS+[COMMA,SEMICOLON,RPARENT])END     PASCP 1465
1559:                     UNTIL SY <> COMMA                                              PASCP 1466
1560:                   END                                                              PASCP 1467
```

```
1561:                ELSE                                                          PASCP 1468
1562:                  BEGIN                                                       PASCP 1469
1563:                    IF SY = FUNCSY THEN                                       PASCP 1470
1564:                      BEGIN ERROR(399); LCP2 := NIL;                          PASCP 1471
1565:                        REPEAT INSYMBOL;                                      PASCP 1472
1566:                          IF SY = IDENT THEN                                  PASCP 1473
1567:                            BEGIN NEW(LCP,FUNC,DECLARED,FORMAL);              PASCP 1474
1568:                              WITH LCP^ DO                                    PASCP 1475
1569:                                BEGIN NAME := ID; IDTYPE := NIL; NEXT := LCP2;PASCP 1476
1570:                                  PFLEV := LEVEL (*BEWARE PARAM FUNCS*);      PASCP 1477
1571:                                  KLASS:=FUNC;PFDECKIND:=DECLARED;            PASCP 1478
1572:                                  PFKIND:=FORMAL                              PASCP 1479
1573:                                END;                                          PASCP 1480
1574:                              ENTERID(LCP);                                   PASCP 1481
1575:                             LCP2 := LCP;                                     P      170
1576:                             ALIGN(PARMPTR,LC);                               P      171
1577:                             (*LC := LC + SOME SIZE*)                         P      172
1578:                              INSYMBOL;                                       PASCP 1483
1579:                            END;                                              PASCP 1484
1580:                          IF NOT (SY IN [COMMA,COLON] + FSYS) THEN            PASCP 1485
1581:                           BEGIN ERROR(7);SKIP(FSYS+[COMMA,SEMICOLON,RPARENT])PASCP 1486
1582:                            END                                               PASCP 1487
1583:                        UNTIL SY <> COMMA;                                    PASCP 1488
1584:                        IF SY = COLON THEN                                    PASCP 1489
1585:                          BEGIN INSYMBOL;                                     PASCP 1490
1586:                            IF SY = IDENT THEN                                PASCP 1491
1587:                              BEGIN SEARCHID([TYPES],LCP);                    PASCP 1492
1588:                                LSP := LCP^.IDTYPE;                           PASCP 1493
1589:                                IF LSP <> NIL THEN                            PASCP 1494
1590:                                 IF NOT(LSP^.FORM IN[SCALAR,SUBRANGE,POINTER])PASCP 1495
1591:                                    THEN BEGIN ERROR(120); LSP := NIL END;    PASCP 1496
1592:                                LCP3 := LCP2;                                 PASCP 1497
1593:                                WHILE LCP2 <> NIL DO                          PASCP 1498
1594:                                  BEGIN LCP2^.IDTYPE := LSP; LCP := LCP2;     PASCP 1499
1595:                                    LCP2 := LCP2^.NEXT                        PASCP 1500
1596:                                  END;                                        PASCP 1501
1597:                                LCP^.NEXT := LCP1; LCP1 := LCP3;              PASCP 1502
1598:                                INSYMBOL                                      PASCP 1503
1599:                              END                                             PASCP 1504
1600:                            ELSE ERROR(2);                                    PASCP 1505
1601:                            IF NOT (SY IN FSYS + [SEMICOLON,RPARENT]) THEN    PASCP 1506
1602:                              BEGIN ERROR(7);SKIP(FSYS+[SEMICOLON,RPARENT])END PASCP 1507
1603:                          END                                                 PASCP 1508
1604:                        ELSE ERROR(5)                                         PASCP 1509
1605:                      END                                                     PASCP 1510
1606:                    ELSE                                                      PASCP 1511
1607:                      BEGIN                                                   PASCP 1512
1608:                        IF SY = VARSY THEN                                    PASCP 1513
1609:                          BEGIN LKIND := FORMAL; INSYMBOL END                 PASCP 1514
1610:                        ELSE LKIND := ACTUAL;                                 PASCP 1515
1611:                        LCP2 := NIL;                                          PASCP 1516
1612:                        COUNT := 0;                                           PASCP 1517
1613:                        REPEAT                                                PASCP 1518
1614:                          IF SY = IDENT THEN                                  PASCP 1519
1615:                            BEGIN NEW(LCP,VARS);                              PASCP 1520
1616:                              WITH LCP^ DO                                    PASCP 1521
1617:                                BEGIN NAME:=ID; IDTYPE:=NIL; KLASS:=VARS;     PASCP 1522
1618:                                  VKIND := LKIND; NEXT := LCP2; VLEV := LEVEL;PASCP 1523
1619:                                END;                                          PASCP 1524
1620:                              ENTERID(LCP);                                   PASCP 1525
1621:                              LCP2 := LCP; COUNT := COUNT+1;                  PASCP 1526
1622:                              INSYMBOL;                                       PASCP 1527
1623:                            END;                                              PASCP 1528
1624:                          IF NOT (SY IN [COMMA,COLON] + FSYS) THEN            PASCP 1529
1625:                           BEGIN ERROR(7);SKIP(FSYS+[COMMA,SEMICOLON,RPARENT])PASCP 1530
```

```
1626:                           END;                                                    PASCP 1531
1627:                         TEST := SY <> COMMA;                                      PASCP 1532
1628:                         IF NOT TEST THEN INSYMBOL                                 PASCP 1533
1629:                       UNTIL TEST;                                                 PASCP 1534
1630:                       IF SY = COLON THEN                                          PASCP 1535
1631:                         BEGIN INSYMBOL;                                           PASCP 1536
1632:                           IF SY = IDENT THEN                                      PASCP 1537
1633:                             BEGIN SEARCHID([TYPES],LCP);                          PASCP 1538
1634:                               LSP := LCP^.IDTYPE;                                 PASCP 1539
1635:                               LSIZE := PTRSIZE;                                   P      173
1636:                               IF LSP <> NIL THEN                                  PASCP 1540
1637:                                 IF LKIND=ACTUAL THEN                              P      174
1638:                                   IF LSP^.FORM<=POWER THEN LSIZE := LSP^.SIZE     P      175
1639:                                   ELSE IF LSP^.FORM=FILES THEN ERROR(121);        P      176
1640:                               ALIGN(PARMPTR,LSIZE);                               P      177
1641:                               LCP3 := LCP2;                                       PASCP 1543
1642:                               ALIGN(PARMPTR,LC);                                  P      178
1643:                               LC := LC+COUNT*LSIZE;                               P      179
1644:                               LLC := LC;                                          PASCP 1548
1645:                               WHILE LCP2 <> NIL DO                                PASCP 1549
1646:                                 BEGIN LCP := LCP2;                                PASCP 1550
1647:                                   WITH LCP2^ DO                                   PASCP 1551
1648:                                     BEGIN IDTYPE := LSP;                          P      180
1649:                                       LLC := LLC-LSIZE;                           P      181
1650:                                       VADDR := LLC;                               PASCP 1553
1651:                                     END;                                          PASCP 1554
1652:                                   LCP2 := LCP2^.NEXT                              PASCP 1555
1653:                                 END;                                              PASCP 1556
1654:                               LCP^.NEXT := LCP1; LCP1 := LCP3;                    PASCP 1557
1655:                               INSYMBOL                                            PASCP 1558
1656:                             END                                                   PASCP 1559
1657:                           ELSE ERROR(2);                                          PASCP 1560
1658:                           IF NOT (SY IN FSYS + [SEMICOLON,RPARENT]) THEN          PASCP 1561
1659:                             BEGIN ERROR(7);SKIP(FSYS+[SEMICOLON,RPARENT])END      PASCP 1562
1660:                         END                                                       PASCP 1563
1661:                       ELSE ERROR(5);                                              PASCP 1564
1662:                     END;                                                          PASCP 1565
1663:                 END;                                                            PASCP 1566
1664:               IF SY = SEMICOLON THEN                                            PASCP 1567
1665:                 BEGIN INSYMBOL;                                                 PASCP 1568
1666:                   IF NOT (SY IN FSYS + [IDENT,VARSY,PROCSY,FUNCSY]) THEN        PASCP 1569
1667:                     BEGIN ERROR(7); SKIP(FSYS + [IDENT,RPARENT]) END            PASCP 1570
1668:                 END                                                             PASCP 1571
1669:             END (*WHILE*) ;                                                     PASCP 1572
1670:           IF SY = RPARENT THEN                                                  PASCP 1573
1671:             BEGIN INSYMBOL;                                                     PASCP 1574
1672:               IF NOT (SY IN FSY + FSYS) THEN                                    PASCP 1575
1673:                 BEGIN ERROR(6); SKIP(FSY + FSYS) END                            PASCP 1576
1674:             END                                                                 PASCP 1577
1675:           ELSE ERROR(4);                                                        PASCP 1578
1676:           LCP3 := NIL;                                                          PASCP 1579
1677:           (*REVERSE POINTERS AND RESERVE LOCAL CELLS FOR COPIES OF MULTIPLE     PASCP 1580
1678:            VALUES*)                                                             PASCP 1581
1679:           WHILE LCP1 <> NIL DO                                                  PASCP 1582
1680:             WITH LCP1^ DO                                                       PASCP 1583
1681:               BEGIN LCP2 := NEXT; NEXT := LCP3;                                 PASCP 1584
1682:                 IF KLASS = VARS THEN                                            PASCP 1585
1683:                   IF IDTYPE <> NIL THEN                                         PASCP 1586
1684:                     IF (VKIND=ACTUAL)AND(IDTYPE^.FORM>POWER) THEN               P      182
1685:                       BEGIN ALIGN(IDTYPE,LC);                                   P      183
1686:                         VADDR := LC;                                            P      184
1687:                         LC := LC+IDTYPE^.SIZE;                                  P      185
1688:                       END;                                                      PASCP 1589
1689:                 LCP3 := LCP1; LCP1 := LCP2                                      PASCP 1590
1690:               END;                                                              PASCP 1591
```

```
1691:             FPAR := LCP3                                                    PASCP 1592
1692:           END                                                               PASCP 1593
1693:             ELSE FPAR := NIL                                                PASCP 1594
1694:     END (*PARAMETERLIST*) ;                                                 PASCP 1595
1695:                                                                             PASCP 1596
1696:     BEGIN (*PROCDECLARATION*)                                               PASCP 1597
1697:       LLC := LC; LC := LCAFTERMARKSTACK; FORW := FALSE;                     P      186
1698:       IF SY = IDENT THEN                                                    PASCP 1599
1699:         BEGIN SEARCHSECTION(DISPLAY[TOP].FNAME,LCP); (*DECIDE WHETHER FORW.*)  PASCP 1600
1700:           IF LCP <> NIL THEN                                                PASCP 1601
1701:           BEGIN                                                             PASCP 1602
1702:             IF LCP^.KLASS = PROC THEN                                       PASCP 1603
1703:               FORW := LCP^.FORWDECL AND(FSY = PROCSY)AND(LCP^.PFKIND = ACTUAL)  PASCP 1604
1704:             ELSE                                                            PASCP 1605
1705:               IF LCP^.KLASS = FUNC THEN                                     PASCP 1606
1706:                 FORW:=LCP^.FORWDECL AND(FSY=FUNCSY)AND(LCP^.PFKIND=ACTUAL)  PASCP 1607
1707:               ELSE FORW := FALSE;                                           PASCP 1608
1708:             IF NOT FORW THEN ERROR(160)                                     PASCP 1609
1709:           END;                                                              P      187
1710:           IF NOT FORW THEN                                                  PASCP 1612
1711:             BEGIN                                                           PASCP 1613
1712:               IF FSY = PROCSY THEN NEW(LCP,PROC,DECLARED,ACTUAL)            PASCP 1614
1713:               ELSE NEW(LCP,FUNC,DECLARED,ACTUAL);                           PASCP 1615
1714:               WITH LCP^ DO                                                  PASCP 1616
1715:                 BEGIN NAME := ID; IDTYPE := NIL;                            PASCP 1617
1716:                   EXTERN := FALSE; PFLEV := LEVEL; GENLABEL(LBNAME);        PASCP 1618
1717:                   PFDECKIND := DECLARED; PFKIND := ACTUAL; PFNAME := LBNAME;  PASCP 1619
1718:                   IF FSY = PROCSY THEN KLASS := PROC                        PASCP 1620
1719:                   ELSE KLASS := FUNC                                        PASCP 1621
1720:                 END;                                                        PASCP 1622
1721:               ENTERID(LCP)                                                  PASCP 1623
1722:             END                                                             PASCP 1624
1723:           ELSE                                                              PASCP 1625
1724:             BEGIN LCP1 := LCP^.NEXT;                                        PASCP 1626
1725:               WHILE LCP1 <> NIL DO                                          PASCP 1627
1726:                 BEGIN                                                       PASCP 1628
1727:                   WITH LCP1^ DO                                             PASCP 1629
1728:                     IF KLASS = VARS THEN                                    PASCP 1630
1729:                       IF IDTYPE <> NIL THEN                                 PASCP 1631
1730:                         BEGIN LCM := VADDR + IDTYPE^.SIZE;                  PASCP 1632
1731:                           IF LCM > LC THEN LC := LCM                        PASCP 1633
1732:                         END;                                                PASCP 1634
1733:                   LCP1 := LCP1^.NEXT                                        PASCP 1635
1734:                 END                                                         PASCP 1636
1735:               END;                                                          PASCP 1637
1736:           INSYMBOL                                                          PASCP 1638
1737:         END                                                                 PASCP 1639
1738:       ELSE                                                                  P      188
1739:         BEGIN ERROR(2); LCP := UFCTPTR END;                                 P      189
1740:       OLDLEV := LEVEL; OLDTOP := TOP;                                       PASCP 1641
1741:       IF LEVEL < MAXLEVEL THEN LEVEL := LEVEL + 1 ELSE ERROR(251);          PASCP 1642
1742:       IF TOP < DISPLIMIT THEN                                               PASCP 1643
1743:         BEGIN TOP := TOP + 1;                                               PASCP 1644
1744:           WITH DISPLAY[TOP] DO                                              PASCP 1645
1745:             BEGIN                                                           PASCP 1646
1746:               IF FORW THEN FNAME := LCP^.NEXT                               PASCP 1647
1747:               ELSE FNAME := NIL;                                            PASCP 1648
1748:               FLABEL := NIL;                                                PASCP 1649
1749:               OCCUR := BLCK                                                 PASCP 1650
1750:             END                                                             PASCP 1651
1751:         END                                                                 PASCP 1652
1752:       ELSE ERROR(250);                                                      PASCP 1653
1753:       IF FSY = PROCSY THEN                                                  PASCP 1654
1754:         BEGIN PARAMETERLIST([SEMICOLON],LCP1);                              PASCP 1655
1755:           IF NOT FORW THEN LCP^.NEXT := LCP1                                PASCP 1656
```

```
1756:          END                                                                    PASCP 1657
1757:        ELSE                                                                     PASCP 1658
1758:          BEGIN PARAMETERLIST([SEMICOLON,COLON],LCP1);                           PASCP 1659
1759:            IF NOT FORW THEN LCP^.NEXT := LCP1;                                  PASCP 1660
1760:            IF SY = COLON THEN                                                   PASCP 1661
1761:              BEGIN INSYMBOL;                                                    PASCP 1662
1762:                IF SY = IDENT THEN                                               PASCP 1663
1763:                  BEGIN IF FORW THEN ERROR(122);                                 PASCP 1664
1764:                    SEARCHID([TYPES],LCP1);                                      PASCP 1665
1765:                    LSP := LCP1^.IDTYPE;                                         PASCP 1666
1766:                    LCP^.IDTYPE := LSP;                                          PASCP 1667
1767:                    IF LSP <> NIL THEN                                           PASCP 1668
1768:                      IF NOT (LSP^.FORM IN [SCALAR,SUBRANGE,POINTER]) THEN       PASCP 1669
1769:                        BEGIN ERROR(120); LCP^.IDTYPE := NIL END;                PASCP 1670
1770:                    INSYMBOL                                                     PASCP 1671
1771:                  END                                                            PASCP 1672
1772:                ELSE BEGIN ERROR(2); SKIP(FSYS + [SEMICOLON]) END                PASCP 1673
1773:              END                                                                PASCP 1674
1774:            ELSE                                                                 PASCP 1675
1775:              IF NOT FORW THEN ERROR(123)                                        PASCP 1676
1776:          END;                                                                   PASCP 1677
1777:        IF SY = SEMICOLON THEN INSYMBOL ELSE ERROR(14);                          PASCP 1678
1778:        IF SY = FORWARDSY THEN                                                   PASCP 1679
1779:          BEGIN                                                                  PASCP 1680
1780:            IF FORW THEN ERROR(161)                                              PASCP 1681
1781:            ELSE LCP^.FORWDECL := TRUE;                                          PASCP 1682
1782:            INSYMBOL;                                                            PASCP 1683
1783:            IF SY = SEMICOLON THEN INSYMBOL ELSE ERROR(14);                      PASCP 1684
1784:            IF NOT (SY IN FSYS) THEN                                             PASCP 1685
1785:              BEGIN ERROR(6); SKIP(FSYS) END                                     PASCP 1686
1786:          END                                                                    PASCP 1687
1787:        ELSE                                                                     PASCP 1688
1788:          BEGIN LCP^.FORWDECL := FALSE; MARK(MARKP);                             BOOT     3
1789:            REPEAT BLOCK(FSYS,SEMICOLON,LCP);                                    PASCP 1690
1790:              IF SY = SEMICOLON THEN                                             PASCP 1691
1791:                BEGIN IF PRTABLES THEN PRINTTABLES(FALSE); INSYMBOL;             PASCP 1692
1792:                  IF NOT (SY IN [BEGINSY,PROCSY,FUNCSY]) THEN                    PASCP 1693
1793:                    BEGIN ERROR(6); SKIP(FSYS) END                               PASCP 1694
1794:                END                                                              PASCP 1695
1795:              ELSE ERROR(14)                                                     PASCP 1696
1796:            UNTIL (SY IN [BEGINSY,PROCSY,FUNCSY]) OR EOF(INPUT);                 P      190
1797:            RELEASE(MARKP); (* RETURN LOCAL ENTRIES ON RUNTIME HEAP *)           PASCP 1698
1798:          END;                                                                   PASCP 1699
1799:        LEVEL := OLDLEV; TOP := OLDTOP; LC := LLC;                               PASCP 1700
1800:      END (*PROCDECLARATION*) ;                                                  PASCP 1701
1801:                                                                                 PASCP 1702
1802:      PROCEDURE BODY(FSYS: SETOFSYS);                                            PASCP 1703
1803:        CONST CSTOCCMAX=65; CIXMAX=1000;                                         J        3
1804:        TYPE OPRANGE = 0..63;                                                    PASCP 1705
1805:        VAR                                                                      PASCP 1706
1806:            LLCP:CTP; SAVEID:ALPHA;                                              PASCP 1707
1807:            CSTPTR: ARRAY [1..CSTOCCMAX] OF CSP;                                 PASCP 1708
1808:            CSTPTRIX: 0..CSTOCCMAX;                                              PASCP 1709
1809:            (*ALLOWS REFERENCING OF NONINTEGER CONSTANTS BY AN INDEX              PASCP 1710
1810:             (INSTEAD OF A POINTER), WHICH CAN BE STORED IN THE P2-FIELD          PASCP 1711
1811:             OF THE INSTRUCTION RECORD UNTIL WRITEOUT.                           PASCP 1712
1812:             --> PROCEDURE LOAD, PROCEDURE WRITEOUT*)                            PASCP 1713
1813:            I, ENTNAME, SEGSIZE: INTEGER;                                        PASCP 1714
1814:            STACKTOP, TOPNEW, TOPMAX: INTEGER;                                   P      191
1815:            LCMAX,LLC1: ADDRRANGE; LCP: CTP;                                     PASCP 1715
1816:            LLP: LBP;                                                            PASCP 1716
1817:                                                                                 PASCP 1717
1818:                                                                                 PASCP 1718
1819:        PROCEDURE MES(I: INTEGER);                                               P      192
1820:        BEGIN TOPNEW := TOPNEW + CDX[I]*MAXSTACK;                                P      193
```

```
1821:         IF TOPNEW > TOPMAX THEN TOPMAX := TOPNEW                          P      194
1822:       END;                                                                P      195
1823:       PROCEDURE PUTIC;                                                    PASCP 1719
1824:       BEGIN IF IC MOD 10 = 0 THEN WRITELN(PRR,'I',IC:5) END;              PASCP 1720
1825:                                                                           PASCP 1721
1826:                                                                           PASCP 1722
1827:       PROCEDURE GEN0(FOP: OPRANGE);                                       PASCP 1723
1828:       BEGIN                                                               PASCP 1724
1829:         IF PRCODE THEN BEGIN PUTIC; WRITELN(PRR,MN[FOP]:4) END;           PASCP 1725
1830:         IC := IC + 1; MES(FOP)                                            P      196
1831:       END (*GEN0*) ;                                                      PASCP 1727
1832:                                                                           PASCP 1728
1833:       PROCEDURE GEN1(FOP: OPRANGE; FP2: INTEGER);                         PASCP 1729
1834:         VAR K: INTEGER;                                                   PASCP 1730
1835:       BEGIN                                                               PASCP 1731
1836:         IF PRCODE THEN                                                    PASCP 1732
1837:           BEGIN PUTIC; WRITE(PRR,MN[FOP]:4);                              PASCP 1733
1838:             IF FOP = 30 THEN                                              P      197
1839:               BEGIN WRITELN(PRR,SNA[FP2]:12);                             P      198
1840:                 TOPNEW := TOPNEW + PDX[FP2]*MAXSTACK;                     P      199
1841:                 IF TOPNEW > TOPMAX THEN TOPMAX := TOPNEW                  P      200
1842:               END                                                         P      201
1843:             ELSE                                                          P      202
1844:               BEGIN                                                       P      203
1845:                 IF FOP = 38 THEN                                          P      204
1846:                    BEGIN WRITE(PRR,'''');                                 PASCP 1736
1847:                      WITH CSTPTR[FP2]^ DO                                 PASCP 1737
1848:                      BEGIN                                                P      205
1849:                        FOR K := 1 TO SLGTH DO WRITE(PRR,SVAL[K]:1);       PASCP 1738
1850:                       FOR K := SLGTH+1 TO STRGLGTH DO WRITE(PRR,' ');     P      206
1851:                      END;                                                 P      207
1852:                      WRITELN(PRR,'''')                                    PASCP 1739
1853:                    END                                                    PASCP 1740
1854:                 ELSE IF FOP = 42 THEN WRITELN(PRR,CHR(FP2))               PASCP 1741
1855:                      ELSE WRITELN(PRR,FP2:12);                            P      208
1856:                 MES(FOP)                                                  P      209
1857:               END                                                         P      210
1858:           END;                                                            PASCP 1743
1859:         IC := IC + 1                                                      PASCP 1744
1860:       END (*GEN1*) ;                                                      PASCP 1745
1861:                                                                           PASCP 1746
1862:       PROCEDURE GEN2(FOP: OPRANGE; FP1,FP2: INTEGER);                     PASCP 1747
1863:         VAR K : INTEGER;                                                  PASCP 1748
1864:       BEGIN                                                               PASCP 1749
1865:         IF PRCODE THEN                                                    PASCP 1750
1866:           BEGIN PUTIC; WRITE(PRR,MN[FOP]:4);                              PASCP 1751
1867:             CASE FOP OF                                                   PASCP 1752
1868:               45,50,54,56:                                                PASCP 1753
1869:                 WRITELN(PRR,' ',FP1:3,FP2:8);                             PASCP 1754
1870:               47,48,49,52,53,55:                                          PASCP 1755
1871:                 BEGIN WRITE(PRR,CHR(FP1));                                PASCP 1756
1872:                   IF CHR(FP1) = 'M' THEN WRITE(PRR,FP2:11);               PASCP 1757
1873:                   WRITELN(PRR)                                            PASCP 1758
1874:                 END;                                                      PASCP 1759
1875:               51:                                                         PASCP 1760
1876:                 CASE FP1 OF                                               PASCP 1761
1877:                   1: WRITELN(PRR,'I ',FP2);                               PASCP 1762
1878:                   2: BEGIN WRITE(PRR,'R ');                               PASCP 1763
1879:                        WITH CSTPTR[FP2]^ DO                               PASCP 1764
1880:                          FOR K := 1 TO STRGLGTH DO WRITE(PRR,RVAL[K]);    PASCP 1765
1881:                        WRITELN(PRR)                                       PASCP 1766
1882:                      END;                                                 PASCP 1767
1883:                   3: WRITELN(PRR,'B ',FP2);                               PASCP 1768
1884:                   4: WRITELN(PRR,'N');                                    PASCP 1769
1885:                   6: WRITELN(PRR,'C ''':3,CHR(FP2),'''');                 P      211
```

```
1886:                   5: BEGIN WRITE(PRR,'(');                                  PASCP 1770
1887:                        WITH CSTPTR[FP2]^ DO                                 PASCP 1771
1888:                          FOR K := 0 TO 58 DO                                PASCP 1772
1889:                            IF K IN PVAL THEN WRITE(PRR,K:3);                PASCP 1773
1890:                        WRITELN(PRR,')')                                     PASCP 1774
1891:                      END                                                    PASCP 1775
1892:                END                                                          PASCP 1776
1893:            END;                                                             PASCP 1777
1894:          END;                                                               PASCP 1778
1895:        IC := IC + 1; MES(FOP)                                               P      212
1896:      END (*GEN2*) ;                                                         PASCP 1780
1897:                                                                             PASCP 1781
1898:      PROCEDURE GENTYPINDICATOR(FSP: STP);                                   P      213
1899:      BEGIN                                                                  P      214
1900:        IF FSP<>NIL THEN                                                     P      215
1901:          WITH FSP^ DO                                                       P      216
1902:            CASE FORM OF                                                     P      217
1903:             SCALAR: IF FSP=INTPTR THEN WRITE(PRR,'I')                       P      218
1904:                     ELSE                                                    P      219
1905:                       IF FSP=BOOLPTR THEN WRITE(PRR,'B')                    P      220
1906:                       ELSE                                                  P      221
1907:                         IF FSP=CHARPTR THEN WRITE(PRR,'C')                  P      222
1908:                         ELSE                                                P      223
1909:                           IF SCALKIND = DECLARED THEN WRITE(PRR,'I')        P      224
1910:                           ELSE WRITE(PRR,'R');                              P      225
1911:             SUBRANGE: GENTYPINDICATOR(RANGETYPE);                           P      226
1912:             POINTER:  WRITE(PRR,'A');                                       P      227
1913:             POWER:    WRITE(PRR,'S');                                       P      228
1914:             RECORDS,ARRAYS: WRITE(PRR,'M');                                 P      229
1915:             FILES,TAGFLD,VARIANT: ERROR(500)                                P      230
1916:            END                                                              P      231
1917:      END (*TYPINDICATOR*);                                                  P      232
1918:                                                                             P      233
1919:      PROCEDURE GEN0T(FOP: OPRANGE; FSP: STP);                               P      234
1920:      BEGIN                                                                  P      235
1921:        IF PRCODE THEN                                                       P      236
1922:          BEGIN PUTIC;                                                       P      237
1923:            WRITE(PRR,MN[FOP]:4);                                            P      238
1924:            GENTYPINDICATOR(FSP);                                            P      239
1925:            WRITELN(PRR);                                                    P      240
1926:          END;                                                               P      241
1927:        IC := IC + 1; MES(FOP)                                               P      242
1928:      END (*GEN0T*);                                                         P      243
1929:                                                                             P      244
1930:      PROCEDURE GEN1T(FOP: OPRANGE; FP2: INTEGER; FSP: STP);                 P      245
1931:      BEGIN                                                                  P      246
1932:        IF PRCODE THEN                                                       P      247
1933:          BEGIN PUTIC;                                                       P      248
1934:            WRITE(PRR,MN[FOP]:4);                                            P      249
1935:            GENTYPINDICATOR(FSP);                                            P      250
1936:            WRITELN(PRR,FP2:11)                                              P      251
1937:          END;                                                               P      252
1938:        IC := IC + 1; MES(FOP)                                               P      253
1939:      END (*GEN1T*);                                                         P      254
1940:                                                                             P      255
1941:      PROCEDURE GEN2T(FOP: OPRANGE; FP1,FP2: INTEGER; FSP: STP);             P      256
1942:      BEGIN                                                                  P      257
1943:        IF PRCODE THEN                                                       P      258
1944:          BEGIN PUTIC;                                                       P      259
1945:            WRITE(PRR,MN[FOP]: 4);                                           P      260
1946:            GENTYPINDICATOR(FSP);                                            P      261
1947:            WRITELN(PRR,FP1:3,FP2:8);                                        P      262
1948:          END;                                                               P      263
1949:        IC := IC + 1; MES(FOP)                                               P      264
1950:      END (*GEN2T*);                                                         P      265
```

```
1951:                                                                                P      266
1952:       PROCEDURE LOAD;                                                          PASCP 1782
1953:       BEGIN                                                                    PASCP 1783
1954:         WITH GATTR DO                                                          PASCP 1784
1955:           IF TYPTR <> NIL THEN                                                 PASCP 1785
1956:             BEGIN                                                              PASCP 1786
1957:               CASE KIND OF                                                     PASCP 1787
1958:                 CST:   IF (TYPTR^.FORM = SCALAR) AND (TYPTR <> REALPTR) THEN   PASCP 1788
1959:                          IF TYPTR = BOOLPTR THEN GEN2(51(*LDC*),3,CVAL.IVAL)   PASCP 1789
1960:                          ELSE                                                  P      267
1961:                            IF TYPTR=CHARPTR THEN                               P      268
1962:                              GEN2(51(*LDC*),6,CVAL.IVAL)                       P      269
1963:                            ELSE GEN2(51(*LDC*),1,CVAL.IVAL)                    P      270
1964:                        ELSE                                                    PASCP 1791
1965:                          IF TYPTR = NILPTR THEN GEN2(51(*LDC*),4,0)            PASCP 1792
1966:                          ELSE                                                  PASCP 1793
1967:                            IF CSTPTRIX >= CSTOCCMAX THEN ERROR(254)            PASCP 1794
1968:                            ELSE                                                PASCP 1795
1969:                              BEGIN CSTPTRIX := CSTPTRIX + 1;                   PASCP 1796
1970:                                CSTPTR[CSTPTRIX] := CVAL.VALP;                  PASCP 1797
1971:                                IF TYPTR = REALPTR THEN                         PASCP 1798
1972:                                  GEN2(51(*LDC*),2,CSTPTRIX)                    PASCP 1799
1973:                                ELSE                                            PASCP 1800
1974:                                   GEN2(51(*LDC*),5,CSTPTRIX)                   PASCP 1801
1975:                              END;                                              PASCP 1802
1976:                 VARBL: CASE ACCESS OF                                          PASCP 1803
1977:                          DRCT:   IF VLEVEL<=1 THEN                             P      271
1978:                                    GEN1T(39(*LDO*),DPLMT,TYPTR)                P      272
1979:                                  ELSE GEN2T(54(*LOD*),LEVEL-VLEVEL,DPLMT,TYPTR);P      273
1980:                          INDRCT: GEN1T(35(*IND*),IDPLMT,TYPTR);                P      274
1981:                          INXD:   ERROR(400)                                    PASCP 1807
1982:                        END;                                                    PASCP 1808
1983:                 EXPR:                                                          PASCP 1809
1984:               END;                                                             PASCP 1810
1985:               KIND := EXPR                                                     PASCP 1811
1986:             END                                                                PASCP 1812
1987:       END (*LOAD*) ;                                                           PASCP 1813
1988:                                                                                PASCP 1814
1989:       PROCEDURE STORE(VAR FATTR: ATTR);                                        PASCP 1815
1990:       BEGIN                                                                    PASCP 1816
1991:         WITH FATTR DO                                                          PASCP 1817
1992:           IF TYPTR <> NIL THEN                                                 PASCP 1818
1993:             CASE ACCESS OF                                                     PASCP 1819
1994:               DRCT:   IF VLEVEL <= 1 THEN GEN1T(43(*SRO*),DPLMT,TYPTR)         P      275
1995:                       ELSE GEN2T(56(*STR*),LEVEL-VLEVEL,DPLMT,TYPTR);          P      276
1996:               INDRCT: IF IDPLMT <> 0 THEN ERROR(400)                           PASCP 1822
1997:                       ELSE GEN0T(26(*STO*),TYPTR);                             P      277
1998:               INXD:   ERROR(400)                                               PASCP 1824
1999:             END                                                                PASCP 1825
2000:       END (*STORE*) ;                                                          PASCP 1826
2001:                                                                                PASCP 1827
2002:       PROCEDURE LOADADDRESS;                                                   PASCP 1828
2003:       BEGIN                                                                    PASCP 1829
2004:         WITH GATTR DO                                                          PASCP 1830
2005:           IF TYPTR <> NIL THEN                                                 PASCP 1831
2006:             BEGIN                                                              PASCP 1832
2007:               CASE KIND OF                                                     PASCP 1833
2008:                 CST:   IF STRING(TYPTR) THEN                                   PASCP 1834
2009:                          IF CSTPTRIX >= CSTOCCMAX THEN ERROR(254)              PASCP 1835
2010:                          ELSE                                                  PASCP 1836
2011:                            BEGIN CSTPTRIX := CSTPTRIX + 1;                     PASCP 1837
2012:                              CSTPTR[CSTPTRIX] := CVAL.VALP;                    PASCP 1838
2013:                              GEN1(38(*LCA*),CSTPTRIX)                          PASCP 1839
2014:                            END                                                 PASCP 1840
2015:                        ELSE ERROR(400);                                        PASCP 1841
```

```
2016:                 VARBL: CASE ACCESS OF                                          PASCP 1842
2017:                          DRCT:   IF VLEVEL <= 1 THEN GEN1(37(*LAO*),DPLMT)     PASCP 1843
2018:                                  ELSE GEN2(50(*LDA*),LEVEL-VLEVEL,DPLMT);      PASCP 1844
2019:                          INDRCT: IF IDPLMT <> 0 THEN                           P      278
2020:                                    GEN1T(34(*INC*),IDPLMT,NILPTR);             P      279
2021:                          INXD:   ERROR(400)                                    PASCP 1846
2022:                        END;                                                    PASCP 1847
2023:                 EXPR:  ERROR(400)                                             PASCP 1848
2024:               END;                                                            PASCP 1849
2025:               KIND := VARBL; ACCESS := INDRCT; IDPLMT := 0                    PASCP 1850
2026:             END                                                               PASCP 1851
2027:       END (*LOADADDRESS*) ;                                                   PASCP 1852
2028:                                                                               PASCP 1853
2029:                                                                               PASCP 1854
2030:       PROCEDURE GENFJP(FADDR: INTEGER);                                       PASCP 1855
2031:       BEGIN LOAD;                                                             PASCP 1856
2032:         IF GATTR.TYPTR <> NIL THEN                                            PASCP 1857
2033:           IF GATTR.TYPTR <> BOOLPTR THEN ERROR(144);                          PASCP 1858
2034:         IF PRCODE THEN BEGIN PUTIC; WRITELN(PRR,MN[33]:4,' L':8,FADDR:4) END; PASCP 1859
2035:         IC := IC + 1; MES(33)                                                 P      280
2036:       END (*GENFJP*) ;                                                        PASCP 1861
2037:                                                                               PASCP 1862
2038:       PROCEDURE GENUJPXJP(FOP: OPRANGE; FP2: INTEGER);                        P      281
2039:      BEGIN                                                                    PASCP 1864
2040:        IF PRCODE THEN                                                         PASCP 1865
2041:           BEGIN PUTIC; WRITELN(PRR, MN[FOP]:4, ' L':8,FP2:4) END;             PASCP 1866
2042:         IC := IC + 1; MES(FOP)                                                P      282
2043:       END (*GENUJPENT*);                                                      PASCP 1868
2044:                                                                               PASCP 1869
2045:                                                                               PASCP 1870
2046:       PROCEDURE GENCUPENT(FOP: OPRANGE; FP1,FP2: INTEGER);                    P      283
2047:       BEGIN                                                                   P      284
2048:         IF PRCODE THEN                                                        P      285
2049:           BEGIN PUTIC;                                                        P      286
2050:             WRITELN(PRR,MN[FOP]:4,FP1:4,'L':4,FP2:4)                          P      287
2051:           END;                                                                P      288
2052:         IC := IC + 1; MES(FOP)                                                P      289
2053:       END;                                                                    P      290
2054:                                                                               PASCP 1877
2055:                                                                               P      291
2056:       PROCEDURE CHECKBNDS(FSP: STP);                                          P      292
2057:         VAR LMIN,LMAX: INTEGER;                                               P      293
2058:       BEGIN                                                                   P      294
2059:         IF FSP <> NIL THEN                                                    P      295
2060:           IF FSP <> INTPTR THEN                                               P      296
2061:             IF FSP <> REALPTR THEN                                            P      297
2062:               IF FSP^.FORM <= SUBRANGE THEN                                   P      298
2063:                 BEGIN                                                         P      299
2064:                   GETBOUNDS(FSP,LMIN,LMAX);                                   P      300
2065:                   GEN2T(45(*CHK*),LMIN,LMAX,FSP)                              P      301
2066:                 END                                                           P      302
2067:       END (*CHECKBNDS*);                                                      P      303
2068:                                                                               P      304
2069:                                                                               PASCP 1878
2070:       PROCEDURE PUTLABEL(LABNAME: INTEGER);                                   PASCP 1879
2071:       BEGIN IF PRCODE THEN WRITELN(PRR, 'L', LABNAME:4)                       PASCP 1880
2072:       END (*PUTLABEL*);                                                       PASCP 1881
2073:                                                                               PASCP 1882
2074:       PROCEDURE STATEMENT(FSYS: SETOFSYS);                                    PASCP 1883
2075:         LABEL 1;                                                              PASCP 1884
2076:         VAR LCP: CTP; LLP: LBP;                                               PASCP 1885
2077:                                                                               PASCP 1886
2078:         PROCEDURE EXPRESSION(FSYS: SETOFSYS); FORWARD;                        PASCP 1887
2079:                                                                               PASCP 1888
2080:         PROCEDURE SELECTOR(FSYS: SETOFSYS; FCP: CTP);                         PASCP 1889
```

```
2081:         VAR LATTR: ATTR; LCP: CTP; LSIZE,LMIN,LMAX: INTEGER;                    P      305
2082:         BEGIN                                                                   PASCP 1891
2083:           WITH FCP^, GATTR DO                                                   PASCP 1892
2084:             BEGIN TYPTR := IDTYPE; KIND := VARBL;                               PASCP 1893
2085:               CASE KLASS OF                                                     PASCP 1894
2086:                 VARS:                                                           PASCP 1895
2087:                   IF VKIND = ACTUAL THEN                                        PASCP 1896
2088:                     BEGIN ACCESS := DRCT; VLEVEL := VLEV;                       PASCP 1897
2089:                       DPLMT := VADDR                                            PASCP 1898
2090:                     END                                                         PASCP 1899
2091:                   ELSE                                                          PASCP 1900
2092:                     BEGIN GEN2T(54(*LOD*),LEVEL-VLEV,VADDR,NILPTR);           P      306
2093:                       ACCESS := INDRCT; IDPLMT := 0                             PASCP 1902
2094:                     END;                                                        PASCP 1903
2095:                 FIELD:                                                          PASCP 1904
2096:                   WITH DISPLAY[DISX] DO                                         PASCP 1905
2097:                     IF OCCUR = CREC THEN                                        PASCP 1906
2098:                       BEGIN ACCESS := DRCT; VLEVEL := CLEV;                     PASCP 1907
2099:                         DPLMT := CDSPL + FLDADDR                                PASCP 1908
2100:                       END                                                       PASCP 1909
2101:                     ELSE                                                        PASCP 1910
2102:                       BEGIN                                                     PASCP 1911
2103:                         IF LEVEL = 1 THEN GEN1T(39(*LOD*),VDSPL,NILPTR)       P      307
2104:                         ELSE GEN2T(54(*LOD*),0,VDSPL,NILPTR);                 P      308
2105:                         ACCESS := INDRCT; IDPLMT := FLDADDR                   PASCP 1914
2106:                       END;                                                      PASCP 1915
2107:                 FUNC:                                                         PASCP 1916
2108:                   IF PFDECKIND = STANDARD THEN                                P      309
2109:                     BEGIN ERROR(150); TYPTR := NIL END                        P      310
2110:                   ELSE                                                        P      311
2111:                     BEGIN                                                     P      312
2112:                       IF PFKIND = FORMAL THEN ERROR(151)                      P      313
2113:                       ELSE                                                    P      314
2114:                         IF (PFLEV+1<>LEVEL)OR(FPROCP<>FCP) THEN ERROR(177);   P      315
2115:                         BEGIN ACCESS := DRCT; VLEVEL := PFLEV + 1;            PASCP 1923
2116:                           DPLMT := 0   (*IMPL. RELAT. ADDR. OF FCT. RESULT*)  PASCP 1924
2117:                         END                                                   PASCP 1925
2118:                     END                                                       P      316
2119:               END (*CASE*)                                                    PASCP 1926
2120:             END (*WITH*);                                                     PASCP 1927
2121:           IF NOT (SY IN SELECTSYS + FSYS) THEN                                PASCP 1928
2122:             BEGIN ERROR(59); SKIP(SELECTSYS + FSYS) END;                      PASCP 1929
2123:           WHILE SY IN SELECTSYS DO                                            PASCP 1930
2124:             BEGIN                                                             PASCP 1931
2125:         (*[*)   IF SY = LBRACK THEN                                           PASCP 1932
2126:                 BEGIN                                                         PASCP 1933
2127:                   REPEAT LATTR := GATTR;                                      PASCP 1934
2128:                     WITH LATTR DO                                             PASCP 1935
2129:                       IF TYPTR <> NIL THEN                                    PASCP 1936
2130:                         IF TYPTR^.FORM <> ARRAYS THEN                         PASCP 1937
2131:                           BEGIN ERROR(138); TYPTR := NIL END;                 PASCP 1938
2132:                     LOADADDRESS;                                              PASCP 1939
2133:                     INSYMBOL; EXPRESSION(FSYS + [COMMA,RBRACK]);              PASCP 1940
2134:                     LOAD;                                                     PASCP 1941
2135:                     IF GATTR.TYPTR <> NIL THEN                                PASCP 1942
2136:                       IF GATTR.TYPTR^.FORM<>SCALAR THEN ERROR(113)            P      317
2137:                       ELSE IF NOT COMPTYPES(GATTR.TYPTR,INTPTR) THEN          P      318
2138:                              GEN0T(58(*ORD*),GATTR.TYPTR);                    P      319
2139:                     IF LATTR.TYPTR <> NIL THEN                                PASCP 1944
2140:                       WITH LATTR.TYPTR^ DO                                    PASCP 1945
2141:                         BEGIN                                                 PASCP 1946
2142:                           IF COMPTYPES(INXTYPE,GATTR.TYPTR) THEN              PASCP 1947
2143:                             BEGIN                                             PASCP 1948
2144:                               IF INXTYPE <> NIL THEN                          PASCP 1949
2145:                                 BEGIN GETBOUNDS(INXTYPE,LMIN,LMAX);           PASCP 1950
```

```
2146:                                 IF DEBUG THEN                                    P      320
2147:                                 GEN2T(45(*CHK*),LMIN,LMAX,INTPTR);               J        4
2148:                               IF LMIN>0 THEN GEN1T(31(*DEC*),LMIN,INTPTR)        J        5
2149:                                 ELSE IF LMIN<0 THEN                              P      323
2150:                                 GEN1T(34(*INC*),-LMIN,INTPTR);                   J        6
2151:                                 (*OR SIMPLY GEN1(31,LMIN)*)                      PASCP 1953
2152:                               END                                                PASCP 1954
2153:                             END                                                  PASCP 1955
2154:                           ELSE ERROR(139);                                       PASCP 1956
2155:                           WITH GATTR DO                                          PASCP 1957
2156:                             BEGIN TYPTR := AELTYPE; KIND := VARBL;               PASCP 1958
2157:                               ACCESS := INDRCT; IDPLMT := 0                      PASCP 1959
2158:                             END;                                                 PASCP 1960
2159:                           IF GATTR.TYPTR <> NIL THEN                             PASCP 1961
2160:                             BEGIN                                                P      325
2161:                               LSIZE := GATTR.TYPTR^.SIZE;                        P      326
2162:                               ALIGN(GATTR.TYPTR,LSIZE);                          P      327
2163:                               GEN1(36(*IXA*),LSIZE)                              P      328
2164:                             END                                                  P      329
2165:                         END                                                      PASCP 1963
2166:                   UNTIL SY <> COMMA;                                             PASCP 1964
2167:                   IF SY = RBRACK THEN INSYMBOL ELSE ERROR(12)                    PASCP 1965
2168:                 END (*IF SY = LBRACK*)                                           PASCP 1966
2169:               ELSE                                                               PASCP 1967
2170:         (*.*)     IF SY = PERIOD THEN                                          PASCP 1968
2171:                   BEGIN                                                        PASCP 1969
2172:                     WITH GATTR DO                                              PASCP 1970
2173:                       BEGIN                                                    PASCP 1971
2174:                         IF TYPTR <> NIL THEN                                   PASCP 1972
2175:                           IF TYPTR^.FORM <> RECORDS THEN                       PASCP 1973
2176:                             BEGIN ERROR(140); TYPTR := NIL END;                PASCP 1974
2177:                         INSYMBOL;                                              PASCP 1975
2178:                         IF SY = IDENT THEN                                     PASCP 1976
2179:                           BEGIN                                                PASCP 1977
2180:                             IF TYPTR <> NIL THEN                               PASCP 1978
2181:                               BEGIN SEARCHSECTION(TYPTR^.FSTFLD,LCP);          PASCP 1979
2182:                                 IF LCP = NIL THEN                              PASCP 1980
2183:                                   BEGIN ERROR(152); TYPTR := NIL END           PASCP 1981
2184:                                 ELSE                                           PASCP 1982
2185:                                   WITH LCP^ DO                                 PASCP 1983
2186:                                     BEGIN TYPTR := IDTYPE;                     PASCP 1984
2187:                                       CASE ACCESS OF                           PASCP 1985
2188:                                         DRCT:   DPLMT := DPLMT + FLDADDR;      PASCP 1986
2189:                                         INDRCT: IDPLMT := IDPLMT + FLDADDR;    PASCP 1987
2190:                                         INXD:   ERROR(400)                     PASCP 1988
2191:                                       END                                      PASCP 1989
2192:                                     END                                        PASCP 1990
2193:                             END;                                               PASCP 1991
2194:                           INSYMBOL                                             PASCP 1992
2195:                         END (*SY = IDENT*)                                     PASCP 1993
2196:                       ELSE ERROR(2)                                            PASCP 1994
2197:                     END (*WITH GATTR*)                                         PASCP 1995
2198:                   END (*IF SY = PERIOD*)                                       PASCP 1996
2199:                 ELSE                                                         PASCP 1997
2200:         (*^*)       BEGIN                                                    PASCP 1998
2201:                     IF GATTR.TYPTR <> NIL THEN                               PASCP 1999
2202:                       WITH GATTR,TYPTR^ DO                                   PASCP 2000
2203:                         IF FORM = POINTER THEN                               PASCP 2001
2204:                           BEGIN LOAD; TYPTR := ELTYPE;                       P      330
2205:                             IF DEBUG THEN GEN2T(45(*CHK*),1,MAXADDR,NILPTR); P      331
2206:                             WITH GATTR DO                                    PASCP 2003
2207:                               BEGIN KIND := VARBL; ACCESS := INDRCT;         PASCP 2004
2208:                                 IDPLMT := 0                                  PASCP 2005
2209:                               END                                            PASCP 2006
2210:                           END                                                PASCP 2007
```

```
2211:                       ELSE                                                PASCP 2008
2212:                         IF FORM = FILES THEN TYPTR := FILTYPE             PASCP 2009
2213:                         ELSE ERROR(141);                                  PASCP 2010
2214:                     INSYMBOL                                              PASCP 2011
2215:                   END;                                                    PASCP 2012
2216:               IF NOT (SY IN FSYS + SELECTSYS) THEN                        PASCP 2013
2217:                 BEGIN ERROR(6); SKIP(FSYS + SELECTSYS) END                PASCP 2014
2218:             END (*WHILE*)                                                 PASCP 2015
2219:           END (*SELECTOR*) ;                                              PASCP 2016
2220:                                                                           PASCP 2017
2221:           PROCEDURE CALL(FSYS: SETOFSYS; FCP: CTP);                       PASCP 2018
2222:             VAR LKEY: 1..15;                                              PASCP 2019
2223:                                                                           PASCP 2020
2224:             PROCEDURE VARIABLE(FSYS: SETOFSYS);                           PASCP 2021
2225:               VAR LCP: CTP;                                               PASCP 2022
2226:             BEGIN                                                         PASCP 2023
2227:               IF SY = IDENT THEN                                          PASCP 2024
2228:                 BEGIN SEARCHID([VARS,FIELD],LCP); INSYMBOL END            PASCP 2025
2229:               ELSE BEGIN ERROR(2); LCP := UVARPTR END;                    PASCP 2026
2230:               SELECTOR(FSYS,LCP)                                          PASCP 2027
2231:             END (*VARIABLE*) ;                                            PASCP 2028
2232:                                                                           PASCP 2029
2233:             PROCEDURE GETPUTRESETREWRITE;                                 PASCP 2030
2234:             BEGIN VARIABLE(FSYS + [RPARENT]); LOADADDRESS;                PASCP 2031
2235:               IF GATTR.TYPTR <> NIL THEN                                  PASCP 2032
2236:                 IF GATTR.TYPTR^.FORM <> FILES THEN ERROR(116);            PASCP 2033
2237:               IF LKEY <= 2 THEN GEN1(30(*CSP*),LKEY(*GET,PUT*))           PASCP 2034
2238:               ELSE ERROR(399)                                             PASCP 2035
2239:             END (*GETPUTRESETREWRITE*) ;                                  PASCP 2036
2240:                                                                           PASCP 2037
2241:             PROCEDURE READ;                                               PASCP 2038
2242:               VAR LCP:CTP; LLEV:LEVRANGE; LADDR:ADDRRANGE;                PASCP 2039
2243:                   LSP : STP;                                              P      332
2244:             BEGIN                                                         P      333
2245:               LLEV := 1; LADDR := LCAFTERMARKSTACK;                       P      334
2246:               IF SY = LPARENT THEN                                        P      335
2247:               BEGIN INSYMBOL;                                             P      336
2248:               VARIABLE(FSYS + [COMMA,RPARENT]);                           P      337
2249:               LSP := GATTR.TYPTR; TEST := FALSE;                          P      338
2250:               IF LSP <> NIL THEN                                          P      339
2251:                 IF LSP^.FORM = FILES THEN                                 P      340
2252:                   WITH GATTR, LSP^ DO                                     P      341
2253:                     BEGIN                                                 P      342
2254:                       IF FILTYPE = CHARPTR THEN                           P      343
2255:                         BEGIN LLEV := VLEVEL; LADDR := DPLMT END          P      344
2256:                       ELSE ERROR(399);                                    P      345
2257:                       IF SY = RPARENT THEN                                P      346
2258:                         BEGIN IF LKEY = 8 THEN ERROR(116);                P      347
2259:                           TEST := TRUE                                    P      348
2260:                         END                                               P      349
2261:                       ELSE                                                P      350
2262:                         IF SY <> COMMA THEN                               P      351
2263:                           BEGIN ERROR(116); SKIP(FSYS + [COMMA,RPARENT]) END;  P  352
2264:                       IF SY = COMMA THEN                                  P      353
2265:                         BEGIN INSYMBOL; VARIABLE(FSYS + [COMMA,RPARENT])  P      354
2266:                         END                                               P      355
2267:                       ELSE TEST := TRUE                                   P      356
2268:                     END;                                                  P      357
2269:              IF NOT TEST THEN                                             P      358
2270:               REPEAT LOADADDRESS;                                         P      359
2271:                 GEN2(50(*LDA*),LEVEL-LLEV,LADDR);                         PASCP 2063
2272:                 IF GATTR.TYPTR <> NIL THEN                                PASCP 2064
2273:                   IF GATTR.TYPTR^.FORM <= SUBRANGE THEN                   PASCP 2065
2274:                     IF COMPTYPES(INTPTR,GATTR.TYPTR) THEN                 PASCP 2066
2275:                       GEN1(30(*CSP*),3(*RDI*))                            PASCP 2067
```

```
2276:                   ELSE                                                    PASCP 2068
2277:                     IF COMPTYPES(REALPTR,GATTR.TYPTR) THEN                PASCP 2069
2278:                       GEN1(30(*CSP*),4(*RDR*))                            PASCP 2070
2279:                     ELSE                                                  PASCP 2071
2280:                       IF COMPTYPES(CHARPTR,GATTR.TYPTR) THEN              PASCP 2072
2281:                         GEN1(30(*CSP*),5(*RDC*))                          PASCP 2073
2282:                       ELSE ERROR(399)                                     PASCP 2074
2283:                 ELSE ERROR(116);                                          PASCP 2075
2284:               TEST := SY <> COMMA;                                        PASCP 2076
2285:               IF NOT TEST THEN                                            P      360
2286:                 BEGIN INSYMBOL; VARIABLE(FSYS + [COMMA,RPARENT])          P      361
2287:                 END                                                       P      362
2288:             UNTIL TEST;                                                   P      363
2289:             IF SY = RPARENT THEN INSYMBOL ELSE ERROR(4)                   P      364
2290:             END                                                           P      365
2291:               ELSE IF LKEY = 5 THEN ERROR(116);                           P      366
2292:            IF LKEY = 11 THEN                                              PASCP 2080
2293:              BEGIN GEN2(50(*LDA*),LEVEL - LLEV, LADDR);                   PASCP 2081
2294:                GEN1(30(*CSP*),21(*RLN*))                                  PASCP 2082
2295:              END                                                          PASCP 2083
2296:           END (*READ*) ;                                                  PASCP 2084
2297:                                                                           PASCP 2085
2298:           PROCEDURE WRITE;                                                PASCP 2086
2299:             VAR LSP: STP; DEFAULT : BOOLEAN; LLKEY: 1..15;                PASCP 2087
2300:                 LCP:CTP; LLEV:LEVRANGE; LADDR,LEN:ADDRRANGE;              PASCP 2088
2301:           BEGIN LLKEY := LKEY;                                            PASCP 2089
2302:             LLEV := 1; LADDR := LCAFTERMARKSTACK + CHARMAX;               P      367
2303:             IF SY = LPARENT THEN                                          P      368
2304:             BEGIN INSYMBOL;                                               P      369
2305:             EXPRESSION(FSYS + [COMMA,COLON,RPARENT]);                     P      370
2306:             LSP := GATTR.TYPTR; TEST := FALSE;                            P      371
2307:             IF LSP <> NIL THEN                                            P      372
2308:               IF LSP^.FORM = FILES THEN                                   P      373
2309:                 WITH GATTR, LSP^ DO                                       P      374
2310:                   BEGIN                                                   P      375
2311:                     IF FILTYPE = CHARPTR THEN                             P      376
2312:                       BEGIN LLEV := VLEVEL; LADDR := DPLMT END            P      377
2313:                     ELSE ERROR(399);                                      P      378
2314:                     IF SY = RPARENT THEN                                  P      379
2315:                       BEGIN IF LLKEY = 10 THEN ERROR(116);                P      380
2316:                         TEST := TRUE                                      P      381
2317:                       END                                                 P      382
2318:                     ELSE                                                  P      383
2319:                       IF SY <> COMMA THEN                                 P      384
2320:                         BEGIN ERROR(116); SKIP(FSYS+[COMMA,RPARENT]) END; P      385
2321:                     IF SY = COMMA THEN                                    P      386
2322:                       BEGIN INSYMBOL; EXPRESSION(FSYS+[COMMA,COLON,RPARENT])  P    387
2323:                       END                                                 P      388
2324:                     ELSE TEST := TRUE                                     P      389
2325:                   END;                                                    P      390
2326:            IF NOT TEST THEN                                               P      391
2327:             REPEAT                                                        P      392
2328:               LSP := GATTR.TYPTR;                                         PASCP 2111
2329:               IF LSP <> NIL THEN                                          PASCP 2112
2330:                 IF LSP^.FORM <= SUBRANGE THEN LOAD ELSE LOADADDRESS;      PASCP 2113
2331:               IF SY = COLON THEN                                          PASCP 2114
2332:                 BEGIN INSYMBOL; EXPRESSION(FSYS + [COMMA,COLON,RPARENT]); PASCP 2115
2333:                   IF GATTR.TYPTR <> NIL THEN                              PASCP 2116
2334:                     IF GATTR.TYPTR <> INTPTR THEN ERROR(116);             PASCP 2117
2335:                   LOAD; DEFAULT := FALSE                                  PASCP 2118
2336:                 END                                                       PASCP 2119
2337:               ELSE DEFAULT := TRUE;                                       PASCP 2120
2338:               IF SY = COLON THEN                                          PASCP 2121
2339:                 BEGIN INSYMBOL; EXPRESSION(FSYS + [COMMA,RPARENT]);       PASCP 2122
2340:                   IF GATTR.TYPTR <> NIL THEN                              PASCP 2123
```

```
2341:                     IF GATTR.TYPTR <> INTPTR THEN ERROR(116);                PASCP 2124
2342:                   IF LSP <> REALPTR THEN ERROR(124);                         PASCP 2125
2343:                   LOAD; ERROR(399);                                          PASCP 2126
2344:                 END                                                          PASCP 2127
2345:               ELSE                                                           PASCP 2128
2346:                 IF LSP = INTPTR THEN                                         PASCP 2129
2347:                   BEGIN IF DEFAULT THEN GEN2(51(*LDC*),1,10);                PASCP 2130
2348:                     GEN2(50(*LDA*),LEVEL-LLEV,LADDR);                        PASCP 2131
2349:                     GEN1(30(*CSP*),6(*WRI*))                                 PASCP 2132
2350:                   END                                                        PASCP 2133
2351:                 ELSE                                                         PASCP 2134
2352:                   IF LSP = REALPTR THEN                                      PASCP 2135
2353:                     BEGIN IF DEFAULT THEN GEN2(51(*LDC*),1,20);              PASCP 2136
2354:                       GEN2(50(*LDA*),LEVEL-LLEV,LADDR);                      PASCP 2137
2355:                       GEN1(30(*CSP*),8(*WRR*))                               PASCP 2138
2356:                     END                                                      PASCP 2139
2357:                   ELSE                                                       PASCP 2140
2358:                     IF LSP = CHARPTR THEN                                    PASCP 2141
2359:                       BEGIN IF DEFAULT THEN GEN2(51(*LDC*),1,1);             PASCP 2142
2360:                         GEN2(50(*LDA*),LEVEL-LLEV,LADDR);                    PASCP 2143
2361:                         GEN1(30(*CSP*),9(*WRC*))                             PASCP 2144
2362:                       END                                                    PASCP 2145
2363:                     ELSE                                                     PASCP 2146
2364:                       IF LSP <> NIL THEN                                     PASCP 2147
2365:                         BEGIN                                                PASCP 2148
2366:                           IF LSP^.FORM = SCALAR THEN ERROR(399)              PASCP 2149
2367:                           ELSE                                               PASCP 2150
2368:                             IF STRING(LSP) THEN                              PASCP 2151
2369:                               BEGIN LEN := LSP^.SIZE DIV CHARMAX;            P      393
2370:                                 IF DEFAULT THEN                              PASCP 2153
2371:                                       GEN2(51(*LDC*),1,LEN);                 PASCP 2154
2372:                                 GEN2(51(*LDC*),1,LEN);                       PASCP 2155
2373:                                 GEN2(50(*LDA*),LEVEL-LLEV,LADDR);            PASCP 2156
2374:                                 GEN1(30(*CSP*),10(*WRS*))                    PASCP 2157
2375:                               END                                            PASCP 2158
2376:                             ELSE ERROR(116)                                  PASCP 2159
2377:                         END;                                                 PASCP 2160
2378:               TEST := SY <> COMMA;                                           PASCP 2161
2379:               IF NOT TEST THEN                                               P      394
2380:                 BEGIN INSYMBOL; EXPRESSION(FSYS + [COMMA,COLON,RPARENT])     P      395
2381:                 END                                                          P      396
2382:             UNTIL TEST;                                                    PASCP 2163
2383:             IF SY = RPARENT THEN INSYMBOL ELSE ERROR(4)                     P      397
2384:             END                                                            P      398
2385:               ELSE IF LKEY = 6 THEN ERROR(116);                            P      399
2386:             IF LLKEY = 12 THEN (*WRITELN*)                                 PASCP 2165
2387:               BEGIN GEN2(50(*LDA*),LEVEL-LLEV,LADDR);                      PASCP 2166
2388:                 GEN1(30(*CSP*),22(*WLN*))                                  PASCP 2167
2389:               END                                                          PASCP 2168
2390:           END (*WRITE*) ;                                                  PASCP 2169
2391:                                                                            PASCP 2170
2392:           PROCEDURE PACK;                                                  PASCP 2171
2393:             VAR LSP,LSP1: STP;                                             PASCP 2172
2394:           BEGIN ERROR(399); VARIABLE(FSYS + [COMMA,RPARENT]);              PASCP 2173
2395:             LSP := NIL; LSP1 := NIL;                                       PASCP 2174
2396:             IF GATTR.TYPTR <> NIL THEN                                     PASCP 2175
2397:               WITH GATTR.TYPTR^ DO                                         PASCP 2176
2398:                 IF FORM = ARRAYS THEN                                      PASCP 2177
2399:                   BEGIN LSP := INXTYPE; LSP1 := AELTYPE END                PASCP 2178
2400:                 ELSE ERROR(116);                                           PASCP 2179
2401:             IF SY = COMMA THEN INSYMBOL ELSE ERROR(20);                    PASCP 2180
2402:             EXPRESSION(FSYS + [COMMA,RPARENT]);                            PASCP 2181
2403:             IF GATTR.TYPTR <> NIL THEN                                     PASCP 2182
2404:               IF GATTR.TYPTR^.FORM <> SCALAR THEN ERROR(116)               PASCP 2183
2405:               ELSE                                                         PASCP 2184
```

```
2406:                 IF NOT COMPTYPES(LSP,GATTR.TYPTR) THEN ERROR(116);              PASCP 2185
2407:               IF SY = COMMA THEN INSYMBOL ELSE ERROR(20);                       PASCP 2186
2408:               VARIABLE(FSYS + [RPARENT]);                                       PASCP 2187
2409:               IF GATTR.TYPTR <> NIL THEN                                        PASCP 2188
2410:                 WITH GATTR.TYPTR^ DO                                            PASCP 2189
2411:                   IF FORM = ARRAYS THEN                                         PASCP 2190
2412:                     BEGIN                                                       PASCP 2191
2413:                       IF NOT COMPTYPES(AELTYPE,LSP1)                            PASCP 2192
2414:                         OR NOT COMPTYPES(INXTYPE,LSP) THEN                      PASCP 2193
2415:                         ERROR(116)                                              PASCP 2194
2416:                     END                                                         PASCP 2195
2417:                   ELSE ERROR(116)                                               PASCP 2196
2418:             END (*PACK*) ;                                                      PASCP 2197
2419:                                                                                 PASCP 2198
2420:             PROCEDURE UNPACK;                                                   PASCP 2199
2421:               VAR LSP,LSP1: STP;                                                PASCP 2200
2422:             BEGIN ERROR(399); VARIABLE(FSYS + [COMMA,RPARENT]);                 PASCP 2201
2423:               LSP := NIL; LSP1 := NIL;                                          PASCP 2202
2424:               IF GATTR.TYPTR <> NIL THEN                                        PASCP 2203
2425:                 WITH GATTR.TYPTR^ DO                                            PASCP 2204
2426:                   IF FORM = ARRAYS THEN                                         PASCP 2205
2427:                     BEGIN LSP := INXTYPE; LSP1 := AELTYPE END                   PASCP 2206
2428:                   ELSE ERROR(116);                                              PASCP 2207
2429:               IF SY = COMMA THEN INSYMBOL ELSE ERROR(20);                       PASCP 2208
2430:               VARIABLE(FSYS + [COMMA,RPARENT]);                                 PASCP 2209
2431:               IF GATTR.TYPTR <> NIL THEN                                        PASCP 2210
2432:                 WITH GATTR.TYPTR^ DO                                            PASCP 2211
2433:                   IF FORM = ARRAYS THEN                                         PASCP 2212
2434:                     BEGIN                                                       PASCP 2213
2435:                       IF NOT COMPTYPES(AELTYPE,LSP1)                            PASCP 2214
2436:                         OR NOT COMPTYPES(INXTYPE,LSP) THEN                      PASCP 2215
2437:                         ERROR(116)                                              PASCP 2216
2438:                     END                                                         PASCP 2217
2439:                   ELSE ERROR(116);                                              PASCP 2218
2440:               IF SY = COMMA THEN INSYMBOL ELSE ERROR(20);                       PASCP 2219
2441:               EXPRESSION(FSYS + [RPARENT]);                                     PASCP 2220
2442:               IF GATTR.TYPTR <> NIL THEN                                        PASCP 2221
2443:                 IF GATTR.TYPTR^.FORM <> SCALAR THEN ERROR(116)                  PASCP 2222
2444:                 ELSE                                                            PASCP 2223
2445:                   IF NOT COMPTYPES(LSP,GATTR.TYPTR) THEN ERROR(116);            PASCP 2224
2446:             END (*UNPACK*) ;                                                    PASCP 2225
2447:                                                                                 PASCP 2226
2448:             PROCEDURE NEW;                                                      PASCP 2227
2449:               LABEL 1;                                                          PASCP 2228
2450:               VAR LSP,LSP1: STP; VARTS,LMIN,LMAX: INTEGER;                      PASCP 2229
2451:                   LSIZE,LSZ: ADDRRANGE; LVAL: VALU;                             PASCP 2230
2452:             BEGIN VARIABLE(FSYS + [COMMA,RPARENT]); LOADADDRESS;                PASCP 2231
2453:               LSP := NIL; VARTS := 0; LSIZE := 0;                               PASCP 2232
2454:               IF GATTR.TYPTR <> NIL THEN                                        PASCP 2233
2455:                 WITH GATTR.TYPTR^ DO                                            PASCP 2234
2456:                   IF FORM = POINTER THEN                                        PASCP 2235
2457:                     BEGIN                                                       PASCP 2236
2458:                       IF ELTYPE <> NIL THEN                                     PASCP 2237
2459:                         BEGIN LSIZE := ELTYPE^.SIZE;                            PASCP 2238
2460:                           IF ELTYPE^.FORM = RECORDS THEN LSP := ELTYPE^.RECVAR  PASCP 2239
2461:                         END                                                     PASCP 2240
2462:                     END                                                         PASCP 2241
2463:                   ELSE ERROR(116);                                              PASCP 2242
2464:               WHILE SY = COMMA DO                                               PASCP 2243
2465:                 BEGIN INSYMBOL;CONSTANT(FSYS + [COMMA,RPARENT],LSP1,LVAL);      PASCP 2244
2466:                   VARTS := VARTS + 1;                                           PASCP 2245
2467:                   (*CHECK TO INSERT HERE: IS CONSTANT IN TAGFIELDTYPE RANGE*)   PASCP 2246
2468:                   IF LSP = NIL THEN ERROR(158)                                  PASCP 2247
2469:                   ELSE                                                          PASCP 2248
2470:                     IF LSP^.FORM <> TAGFLD THEN ERROR(162)                      PASCP 2249
```

```
2471:                 ELSE                                                    PASCP 2250
2472:                   IF LSP^.TAGFIELDP <> NIL THEN                         PASCP 2251
2473:                     IF STRING(LSP1) OR (LSP1 = REALPTR) THEN ERROR(159) PASCP 2252
2474:                     ELSE                                                PASCP 2253
2475:                       IF COMPTYPES(LSP^.TAGFIELDP^.IDTYPE,LSP1) THEN    PASCP 2254
2476:                         BEGIN                                           PASCP 2255
2477:                           LSP1 := LSP^.FSTVAR;                          PASCP 2256
2478:                           WHILE LSP1 <> NIL DO                          PASCP 2257
2479:                             WITH LSP1^ DO                               PASCP 2258
2480:                               IF VARVAL.IVAL = LVAL.IVAL THEN           PASCP 2259
2481:                                 BEGIN LSIZE := SIZE; LSP := SUBVAR;     PASCP 2260
2482:                                   GOTO 1                                PASCP 2261
2483:                                 END                                     PASCP 2262
2484:                               ELSE LSP1 := NXTVAR;                      PASCP 2263
2485:                           LSIZE := LSP^.SIZE; LSP := NIL;               PASCP 2264
2486:                         END                                             PASCP 2265
2487:                       ELSE ERROR(116);                                  PASCP 2266
2488:           1:  END (*WHILE*) ;                                           PASCP 2267
2489:             GEN2(51(*LDC*),1,LSIZE);                                    PASCP 2268
2490:             GEN1(30(*CSP*),12(*NEW*));                                  PASCP 2269
2491:           END (*NEW*) ;                                                 PASCP 2270
2492:                                                                         PASCP 2271
2493:           PROCEDURE MARK;                                               PASCP 2272
2494:           BEGIN VARIABLE(FSYS+[RPARENT]);                               PASCP 2273
2495:              IF GATTR.TYPTR <> NIL THEN                                 PASCP 2274
2496:                IF GATTR.TYPTR^.FORM = POINTER THEN                      PASCP 2275
2497:                  BEGIN LOADADDRESS; GEN1(30(*CSP*),23(*SAV*)) END       PASCP 2276
2498:                ELSE ERROR(125)                                          PASCP 2277
2499:           END(*MARK*);                                                  PASCP 2278
2500:                                                                         PASCP 2279
2501:           PROCEDURE RELEASE;                                            PASCP 2280
2502:           BEGIN VARIABLE(FSYS+[RPARENT]);                               PASCP 2281
2503:                 IF GATTR.TYPTR <> NIL THEN                              PASCP 2282
2504:                    IF GATTR.TYPTR^.FORM = POINTER THEN                  PASCP 2283
2505:                       BEGIN LOAD; GEN1(30(*CSP*),13(*RST*)) END         PASCP 2284
2506:                    ELSE ERROR(125)                                      PASCP 2285
2507:           END (*RELEASE*);                                              PASCP 2286
2508:                                                                         PASCP 2287
2509:                                                                         PASCP 2288
2510:                                                                         PASCP 2289
2511:           PROCEDURE ABS;                                                PASCP 2290
2512:           BEGIN                                                         PASCP 2291
2513:             IF GATTR.TYPTR <> NIL THEN                                  PASCP 2292
2514:               IF GATTR.TYPTR = INTPTR THEN GEN0(0(*ABI*))               PASCP 2293
2515:               ELSE                                                      PASCP 2294
2516:                 IF GATTR.TYPTR = REALPTR THEN GEN0(1(*ABR*))            PASCP 2295
2517:                 ELSE BEGIN ERROR(125); GATTR.TYPTR := INTPTR END        PASCP 2296
2518:           END (*ABS*) ;                                                 PASCP 2297
2519:                                                                         PASCP 2298
2520:           PROCEDURE SQR;                                                PASCP 2299
2521:           BEGIN                                                         PASCP 2300
2522:             IF GATTR.TYPTR <> NIL THEN                                  PASCP 2301
2523:               IF GATTR.TYPTR = INTPTR THEN GEN0(24(*SQI*))              PASCP 2302
2524:               ELSE                                                      PASCP 2303
2525:                 IF GATTR.TYPTR = REALPTR THEN GEN0(25(*SQR*))           PASCP 2304
2526:                 ELSE BEGIN ERROR(125); GATTR.TYPTR := INTPTR END        PASCP 2305
2527:           END (*SQR*) ;                                                 PASCP 2306
2528:                                                                         PASCP 2307
2529:           PROCEDURE TRUNC;                                              PASCP 2308
2530:           BEGIN                                                         PASCP 2309
2531:             IF GATTR.TYPTR <> NIL THEN                                  PASCP 2310
2532:               IF GATTR.TYPTR <> REALPTR THEN ERROR(125);                PASCP 2311
2533:             GEN0(27(*TRC*));                                            PASCP 2312
2534:             GATTR.TYPTR := INTPTR                                       PASCP 2313
2535:           END (*TRUNC*) ;                                               PASCP 2314
```

```
2536:                                                                                PASCP 2315
2537:           PROCEDURE ODD;                                                       PASCP 2316
2538:           BEGIN                                                                PASCP 2317
2539:             IF GATTR.TYPTR <> NIL THEN                                         PASCP 2318
2540:               IF GATTR.TYPTR <> INTPTR THEN ERROR(125);                        PASCP 2319
2541:             GEN0(20(*ODD*));                                                   PASCP 2320
2542:             GATTR.TYPTR := BOOLPTR                                             PASCP 2321
2543:           END (*ODD*) ;                                                        PASCP 2322
2544:                                                                                PASCP 2323
2545:           PROCEDURE ORD;                                                       PASCP 2324
2546:           BEGIN                                                                PASCP 2325
2547:             IF GATTR.TYPTR <> NIL THEN                                         PASCP 2326
2548:               IF GATTR.TYPTR^.FORM >= POWER THEN ERROR(125);                   PASCP 2327
2549:             GEN0T(58(*ORD*),GATTR.TYPTR);                                      P      400
2550:             GATTR.TYPTR := INTPTR                                              PASCP 2328
2551:           END (*ORD*) ;                                                        PASCP 2329
2552:                                                                                PASCP 2330
2553:           PROCEDURE CHR;                                                       PASCP 2331
2554:           BEGIN                                                                PASCP 2332
2555:             IF GATTR.TYPTR <> NIL THEN                                         PASCP 2333
2556:               IF GATTR.TYPTR <> INTPTR THEN ERROR(125);                        PASCP 2334
2557:             GEN0(59(*CHR*));                                                   P      401
2558:             GATTR.TYPTR := CHARPTR                                             PASCP 2335
2559:           END (*CHR*) ;                                                        PASCP 2336
2560:                                                                                PASCP 2337
2561:                                                                                PASCP 2338
2562:                                                                                PASCP 2339
2563:           PROCEDURE PREDSUCC;                                                  PASCP 2340
2564:           BEGIN                                                                P      402
2565:             IF GATTR.TYPTR <> NIL THEN                                         PASCP 2342
2566:               IF GATTR.TYPTR^.FORM <> SCALAR THEN ERROR(125);                  PASCP 2343
2567:             IF LKEY = 7 THEN GEN1T(31(*DEC*),1,GATTR.TYPTR)                    X1       1
2568:             ELSE GEN1T(34(*INC*),1,GATTR.TYPTR)                                X1       2
2569:           END (*PREDSUCC*) ;                                                   PASCP 2344
2570:                                                                                PASCP 2345
2571:           PROCEDURE EOF;                                                       PASCP 2346
2572:           BEGIN                                                                P      405
2573:             IF SY = LPARENT THEN                                               P      406
2574:               BEGIN INSYMBOL; VARIABLE(FSYS + [RPARENT]);                      P      407
2575:                 IF SY = RPARENT THEN INSYMBOL ELSE ERROR(4)                    P      408
2576:               END                                                            P      409
2577:             ELSE                                                             P      410
2578:               WITH GATTR DO                                                  P      411
2579:                 BEGIN TYPTR := TEXTPTR; KIND := VARBL; ACCESS := DRCT;       P      412
2580:                   VLEVEL := 1; DPLMT := LCAFTERMARKSTACK                     P      413
2581:                 END;                                                         P      414
2582:             LOADADDRESS;                                                     P      415
2583:             IF GATTR.TYPTR <> NIL THEN                                       PASCP 2348
2584:               IF GATTR.TYPTR^.FORM <> FILES THEN ERROR(125);                 PASCP 2349
2585:             IF LKEY = 9 THEN GEN0(8(*EOF*)) ELSE GEN1(30(*CSP*),14(*ELN*));  PASCP 2350
2586:               GATTR.TYPTR := BOOLPTR                                         PASCP 2351
2587:           END (*EOF*) ;                                                        PASCP 2352
2588:                                                                                PASCP 2353
2589:           PROCEDURE CALLNONSTANDARD;                                           PASCP 2354
2590:             VAR NXT,LCP: CTP; LSP: STP; LKIND: IDKIND; LB: BOOLEAN;            PASCP 2355
2591:                 LOCPAR, LLC: ADDRRANGE;                                      PASCP 2356
2592:           BEGIN LOCPAR := 0;                                                   PASCP 2357
2593:             WITH FCP^ DO                                                       PASCP 2358
2594:               BEGIN NXT := NEXT; LKIND := PFKIND;                            PASCP 2359
2595:                 IF NOT EXTERN THEN GEN1(41(*MST*),LEVEL-PFLEV)               PASCP 2360
2596:               END;                                                           PASCP 2361
2597:             IF SY = LPARENT THEN                                             PASCP 2362
2598:               BEGIN LLC := LC;                                               PASCP 2363
2599:                 REPEAT LB := FALSE; (*DECIDE WHETHER PROC/FUNC MUST BE PASSED*) PASCP 2364
2600:                   IF LKIND = ACTUAL THEN                                     PASCP 2365
```

```
2601:                     BEGIN                                                  PASCP 2366
2602:                       IF NXT = NIL THEN ERROR(126)                         PASCP 2367
2603:                       ELSE LB := NXT^.KLASS IN [PROC,FUNC]                 PASCP 2368
2604:                     END ELSE ERROR(399);                                   PASCP 2369
2605:                   (*FOR FORMAL PROC/FUNC LB IS FALSE AND EXPRESSION        PASCP 2370
2606:                    WILL BE CALLED, WHICH WILL ALLWAYS INTERPRET A PROC/FUNC ID PASCP 2371
2607:                   AT ITS BEGINNING AS A CALL RATHER THAN A PARAMETER PASSING. PASCP 2372
2608:                   IN THIS IMPLEMENTATION, PARAMETER PROCEDURES/FUNCTIONS   PASCP 2373
2609:                   ARE THEREFORE NOT ALLOWED TO HAVE PROCEDURE/FUNCTION     PASCP 2374
2610:                   PARAMETERS*)                                             PASCP 2375
2611:                   INSYMBOL;                                                PASCP 2376
2612:                   IF LB THEN   (*PASS FUNCTION OR PROCEDURE*)              PASCP 2377
2613:                     BEGIN ERROR(399);                                      PASCP 2378
2614:                       IF SY <> IDENT THEN                                  PASCP 2379
2615:                         BEGIN ERROR(2); SKIP(FSYS + [COMMA,RPARENT]) END   PASCP 2380
2616:                       ELSE                                                 PASCP 2381
2617:                         BEGIN                                              PASCP 2382
2618:                           IF NXT^.KLASS = PROC THEN SEARCHID([PROC],LCP)   PASCP 2383
2619:                           ELSE                                             PASCP 2384
2620:                             BEGIN SEARCHID([FUNC],LCP);                    PASCP 2385
2621:                               IF NOT COMPTYPES(LCP^.IDTYPE,NXT^.IDTYPE) THEN PASCP 2386
2622:                                 ERROR(128)                                 PASCP 2387
2623:                             END;                                           PASCP 2388
2624:                           INSYMBOL;                                        PASCP 2389
2625:                           IF NOT (SY IN FSYS + [COMMA,RPARENT]) THEN       PASCP 2390
2626:                             BEGIN ERROR(6); SKIP(FSYS + [COMMA,RPARENT]) END PASCP 2391
2627:                         END                                                PASCP 2392
2628:                     END (*IF LB*)                                          PASCP 2393
2629:                   ELSE                                                     PASCP 2394
2630:                     BEGIN EXPRESSION(FSYS + [COMMA,RPARENT]);              PASCP 2395
2631:                       IF GATTR.TYPTR <> NIL THEN                           PASCP 2396
2632:                         IF LKIND = ACTUAL THEN                             PASCP 2397
2633:                           BEGIN                                            PASCP 2398
2634:                             IF NXT <> NIL THEN                             PASCP 2399
2635:                               BEGIN LSP := NXT^.IDTYPE;                    PASCP 2400
2636:                                 IF LSP <> NIL THEN                         PASCP 2401
2637:                                   BEGIN                                    PASCP 2402
2638:                                     IF (NXT^.VKIND = ACTUAL) THEN          PASCP 2403
2639:                                       IF LSP^.FORM <= POWER THEN           P      416
2640:                                       BEGIN LOAD;                          PASCP 2405
2641:                                         IF DEBUG THEN CHECKBNDS(LSP);      P      417
2642:                                         IF COMPTYPES(REALPTR,LSP)          PASCP 2406
2643:                                            AND (GATTR.TYPTR = INTPTR) THEN PASCP 2407
2644:                                           BEGIN GEN0(10(*FLT*));           PASCP 2408
2645:                                             GATTR.TYPTR := REALPTR         PASCP 2409
2646:                                           END;                             PASCP 2410
2647:                                         LOCPAR := LOCPAR+LSP^.SIZE;        P      418
2648:                                         ALIGN(PARMPTR,LOCPAR);             P      419
2649:                                       END                                  PASCP 2412
2650:                                       ELSE                                 PASCP 2413
2651:                                       BEGIN                                PASCP 2414
2652:                                         LOADADDRESS;                       P      420
2653:                                         LOCPAR := LOCPAR+PTRSIZE;          P      421
2654:                                         ALIGN(PARMPTR,LOCPAR)              P      422
2655:                                       END                                  PASCP 2440
2656:                                     ELSE                                   PASCP 2441
2657:                                       IF GATTR.KIND = VARBL THEN           PASCP 2442
2658:                                        BEGIN LOADADDRESS;                  P      423
2659:                                          LOCPAR := LOCPAR+PARMSIZE         P      424
2660:                                        END                                 PASCP 2444
2661:                                       ELSE ERROR(154);                     PASCP 2445
2662:                                     IF NOT COMPTYPES(LSP,GATTR.TYPTR) THEN PASCP 2446
2663:                                       ERROR(142)                           PASCP 2447
2664:                                   END                                      PASCP 2448
2665:                               END                                          PASCP 2449
```

```
2666:                         END                                                                PASCP 2450
2667:                       ELSE (*LKIND = FORMAL*)                                              PASCP 2451
2668:                         BEGIN (*PASS FORMAL PARAM*)                                        PASCP 2452
2669:                         END                                                                PASCP 2453
2670:                     END;                                                                   PASCP 2454
2671:                   IF (LKIND = ACTUAL) AND (NXT <> NIL) THEN NXT := NXT^.NEXT               PASCP 2455
2672:                 UNTIL SY <> COMMA;                                                         PASCP 2456
2673:                 LC := LLC;                                                                 PASCP 2457
2674:               IF SY = RPARENT THEN INSYMBOL ELSE ERROR(4)                                  PASCP 2458
2675:             END (*IF LPARENT*);                                                            PASCP 2459
2676:             IF LKIND = ACTUAL THEN                                                         PASCP 2460
2677:               BEGIN IF NXT <> NIL THEN ERROR(126);                                         PASCP 2461
2678:                 WITH FCP^ DO                                                               PASCP 2462
2679:                   BEGIN                                                                    PASCP 2463
2680:                     IF EXTERN THEN GEN1(30(*CSP*),PFNAME)                                  PASCP 2464
2681:                     ELSE GENCUPENT(46(*CUP*),LOCPAR,PFNAME);                               P      425
2682:                   END                                                                      PASCP 2466
2683:               END;                                                                         PASCP 2467
2684:             GATTR.TYPTR := FCP^.IDTYPE                                                     PASCP 2468
2685:           END (*CALLNONSTANDARD*) ;                                                        PASCP 2469
2686:                                                                                            PASCP 2470
2687:         BEGIN (*CALL*)                                                                     PASCP 2471
2688:           IF FCP^.PFDECKIND = STANDARD THEN                                                PASCP 2472
2689:             BEGIN LKEY := FCP^.KEY;                                                        P      426
2690:               IF FCP^.KLASS = PROC THEN                                                    PASCP 2475
2691:                BEGIN                                                                       P      427
2692:                 IF NOT(LKEY IN [5,6,11,12]) THEN                                           P      428
2693:                   IF SY = LPARENT THEN INSYMBOL ELSE ERROR(9);                             P      429
2694:                 CASE LKEY OF                                                               PASCP 2476
2695:                   1,2,                                                                     PASCP 2477
2696:                   3,4:  GETPUTRESETREWRITE;                                                PASCP 2478
2697:                   5,11:    READ;                                                           PASCP 2479
2698:                   6,12:    WRITE;                                                          PASCP 2480
2699:                   7:    PACK;                                                              PASCP 2481
2700:                   8:    UNPACK;                                                            PASCP 2482
2701:                   9:    NEW;                                                               PASCP 2483
2702:                   10:   RELEASE;                                                          PASCP 2484
2703:                   13:   MARK                                                             PASCP 2485
2704:                 END;                                                                       P      430
2705:                 IF NOT(LKEY IN [5,6,11,12]) THEN                                           P      431
2706:                   IF SY = RPARENT THEN INSYMBOL ELSE ERROR(4)                              P      432
2707:                END                                                                         P      433
2708:               ELSE                                                                         PASCP 2487
2709:                 BEGIN                                                                      P      434
2710:                   IF LKEY <= 8 THEN                                                        P      435
2711:                     BEGIN                                                                  P      436
2712:                       IF SY = LPARENT THEN INSYMBOL ELSE ERROR(9);                         P      437
2713:                       EXPRESSION(FSYS+[RPARENT]); LOAD                                     P      438
2714:                     END;                                                                   P      439
2715:                   CASE LKEY OF                                                             PASCP 2490
2716:                     1:    ABS;                                                             PASCP 2491
2717:                     2:    SQR;                                                             PASCP 2492
2718:                     3:    TRUNC;                                                           PASCP 2493
2719:                     4:    ODD;                                                             PASCP 2494
2720:                     5:    ORD;                                                             PASCP 2495
2721:                     6:    CHR;                                                             PASCP 2496
2722:                     7,8:  PREDSUCC;                                                        PASCP 2497
2723:                     9,10:    EOF                                                           PASCP 2498
2724:                   END;                                                                     P      440
2725:                   IF LKEY <= 8 THEN                                                        P      441
2726:                     IF SY = RPARENT THEN INSYMBOL ELSE ERROR(4)                            P      442
2727:                 END;                                                                       PASCP 2500
2728:             END (*STANDARD PROCEDURES AND FUNCTIONS*)                                      PASCP 2502
2729:           ELSE CALLNONSTANDARD                                                             PASCP 2503
2730:         END (*CALL*) ;                                                                     PASCP 2504
```

```
2731:                                                                                PASCP 2505
2732:          PROCEDURE EXPRESSION;                                                 PASCP 2506
2733:            VAR LATTR: ATTR; LOP: OPERATOR; TYPIND: CHAR; LSIZE: ADDRRANGE;     PASCP 2507
2734:                                                                                PASCP 2508
2735:            PROCEDURE SIMPLEEXPRESSION(FSYS: SETOFSYS);                         PASCP 2509
2736:              VAR LATTR: ATTR; LOP: OPERATOR; SIGNED: BOOLEAN;                  PASCP 2510
2737:                                                                                PASCP 2511
2738:              PROCEDURE TERM(FSYS: SETOFSYS);                                   PASCP 2512
2739:                VAR LATTR: ATTR; LOP: OPERATOR;                                 PASCP 2513
2740:                                                                                PASCP 2514
2741:                PROCEDURE FACTOR(FSYS: SETOFSYS);                               PASCP 2515
2742:                  VAR LCP: CTP; LVP: CSP; VARPART: BOOLEAN;                     PASCP 2516
2743:                      CSTPART: SET OF 0..58; LSP: STP;                          PASCP 2517
2744:                BEGIN                                                           PASCP 2518
2745:                  IF NOT (SY IN FACBEGSYS) THEN                                 PASCP 2519
2746:                    BEGIN ERROR(58); SKIP(FSYS + FACBEGSYS);                    PASCP 2520
2747:                      GATTR.TYPTR := NIL                                        PASCP 2521
2748:                    END;                                                        PASCP 2522
2749:                  WHILE SY IN FACBEGSYS DO                                      PASCP 2523
2750:                    BEGIN                                                       PASCP 2524
2751:                      CASE SY OF                                                PASCP 2525
2752:                (*ID*)    IDENT:                                                PASCP 2526
2753:                          BEGIN SEARCHID([KONST,VARS,FIELD,FUNC],LCP);          PASCP 2527
2754:                            INSYMBOL;                                           PASCP 2528
2755:                            IF LCP^.KLASS = FUNC THEN                           PASCP 2529
2756:                              BEGIN CALL(FSYS,LCP);                             P      443
2757:                                WITH GATTR DO                                   P      444
2758:                                  BEGIN KIND := EXPR;                           P      445
2759:                                    IF TYPTR <> NIL THEN                        P      446
2760:                                      IF TYPTR^.FORM=SUBRANGE THEN              P      447
2761:                                        TYPTR := TYPTR^.RANGETYPE               P      448
2762:                                  END                                           P      449
2763:                              END                                               P      450
2764:                            ELSE                                                PASCP 2531
2765:                              IF LCP^.KLASS = KONST THEN                        PASCP 2532
2766:                                WITH GATTR, LCP^ DO                             PASCP 2533
2767:                                  BEGIN TYPTR := IDTYPE; KIND := CST;           PASCP 2534
2768:                                    CVAL := VALUES                              PASCP 2535
2769:                                  END                                           PASCP 2536
2770:                              ELSE                                              PASCP 2537
2771:                                BEGIN SELECTOR(FSYS,LCP);                       PASCP 2538
2772:                                  IF GATTR.TYPTR<>NIL THEN(*ELIM.SUBR.TYPES TO*) PASCP 2539
2773:                                    WITH GATTR,TYPTR^ DO(*SIMPLIFY LATER TESTS*) PASCP 2540
2774:                                      IF FORM = SUBRANGE THEN                   PASCP 2541
2775:                                        TYPTR := RANGETYPE                      PASCP 2542
2776:                                END                                             PASCP 2543
2777:                          END;                                                  PASCP 2544
2778:                (*CST*)   INTCONST:                                             PASCP 2545
2779:                          BEGIN                                                 PASCP 2546
2780:                            WITH GATTR DO                                       PASCP 2547
2781:                              BEGIN TYPTR := INTPTR; KIND := CST;               PASCP 2548
2782:                                CVAL := VAL                                     PASCP 2549
2783:                              END;                                              PASCP 2550
2784:                            INSYMBOL                                            PASCP 2551
2785:                          END;                                                  PASCP 2552
2786:                        REALCONST:                                              PASCP 2553
2787:                          BEGIN                                                 PASCP 2554
2788:                            WITH GATTR DO                                       PASCP 2555
2789:                              BEGIN TYPTR := REALPTR; KIND := CST;              PASCP 2556
2790:                                CVAL := VAL                                     PASCP 2557
2791:                              END;                                              PASCP 2558
2792:                            INSYMBOL                                            PASCP 2559
2793:                          END;                                                  PASCP 2560
2794:                        STRINGCONST:                                            PASCP 2561
2795:                          BEGIN                                                 PASCP 2562
```

```
2796:                         WITH GATTR DO                                          PASCP 2563
2797:                           BEGIN                                                PASCP 2564
2798:                             IF LGTH = 1 THEN TYPTR := CHARPTR                  PASCP 2565
2799:                             ELSE                                               PASCP 2566
2800:                               BEGIN NEW(LSP,ARRAYS);                           PASCP 2567
2801:                                 WITH LSP^ DO                                   PASCP 2568
2802:                                   BEGIN AELTYPE := CHARPTR; FORM:=ARRAYS;      PASCP 2569
2803:                                     INXTYPE := NIL; SIZE := LGTH*CHARSIZE      PASCP 2570
2804:                                   END;                                         PASCP 2571
2805:                                 TYPTR := LSP                                   PASCP 2572
2806:                               END;                                             PASCP 2573
2807:                             KIND := CST; CVAL := VAL                           PASCP 2574
2808:                           END;                                                 PASCP 2575
2809:                         INSYMBOL                                               PASCP 2576
2810:                       END;                                                     PASCP 2577
2811:               (*(*)   LPARENT:                                                 PASCP 2578
2812:                       BEGIN INSYMBOL; EXPRESSION(FSYS + [RPARENT]);            PASCP 2579
2813:                         IF SY = RPARENT THEN INSYMBOL ELSE ERROR(4)            PASCP 2580
2814:                       END;                                                     PASCP 2581
2815:               (*NOT*) NOTSY:                                                   PASCP 2582
2816:                       BEGIN INSYMBOL; FACTOR(FSYS);                            PASCP 2583
2817:                         LOAD; GEN0(19(*NOT*));                                 PASCP 2584
2818:                         IF GATTR.TYPTR <> NIL THEN                             PASCP 2585
2819:                           IF GATTR.TYPTR <> BOOLPTR THEN                       PASCP 2586
2820:                             BEGIN ERROR(135); GATTR.TYPTR := NIL END;          PASCP 2587
2821:                       END;                                                     PASCP 2588
2822:               (*[*)   LBRACK:                                                  PASCP 2589
2823:                       BEGIN INSYMBOL; CSTPART := [ ]; VARPART := FALSE;        PASCP 2590
2824:                         NEW(LSP,POWER);                                        PASCP 2591
2825:                         WITH LSP^ DO                                           PASCP 2592
2826:                           BEGIN ELSET:=NIL;SIZE:=SETSIZE;FORM:=POWER END;      PASCP 2593
2827:                         IF SY = RBRACK THEN                                    PASCP 2594
2828:                           BEGIN                                                PASCP 2595
2829:                             WITH GATTR DO                                      PASCP 2596
2830:                               BEGIN TYPTR := LSP; KIND := CST END;             PASCP 2597
2831:                             INSYMBOL                                           PASCP 2598
2832:                           END                                                  PASCP 2599
2833:                         ELSE                                                   PASCP 2600
2834:                           BEGIN                                                PASCP 2601
2835:                             REPEAT EXPRESSION(FSYS + [COMMA,RBRACK]);          PASCP 2602
2836:                               IF GATTR.TYPTR <> NIL THEN                       PASCP 2603
2837:                                 IF GATTR.TYPTR^.FORM <> SCALAR THEN            PASCP 2604
2838:                                   BEGIN ERROR(136); GATTR.TYPTR := NIL END     PASCP 2605
2839:                                 ELSE                                           PASCP 2606
2840:                                   IF COMPTYPES(LSP^.ELSET,GATTR.TYPTR) THEN    PASCP 2607
2841:                                     BEGIN                                      PASCP 2608
2842:                                       IF GATTR.KIND = CST THEN                 PASCP 2609
2843:                                         IF (GATTR.CVAL.IVAL < SETLOW) OR       P      451
2844:                                           (GATTR.CVAL.IVAL > SETHIGH) THEN     P      452
2845:                                           ERROR(304)                           P      453
2846:                                         ELSE                                   P      454
2847:                                           CSTPART := CSTPART+[GATTR.CVAL.IVAL]P      455
2848:                                       ELSE                                     PASCP 2611
2849:                                         BEGIN LOAD;                            P      456
2850:                                           IF NOT COMPTYPES(GATTR.TYPTR,INTPTR)P      457
2851:                                           THEN GEN0T(58(*ORD*),GATTR.TYPTR);   P      458
2852:                                           GEN0(23(*SGS*));                     P      459
2853:                                           IF VARPART THEN GEN0(28(*UNI*))      PASCP 2613
2854:                                           ELSE VARPART := TRUE                 PASCP 2614
2855:                                         END;                                   PASCP 2615
2856:                                       LSP^.ELSET := GATTR.TYPTR;               PASCP 2616
2857:                                       GATTR.TYPTR := LSP                       PASCP 2617
2858:                                     END                                        PASCP 2618
2859:                                   ELSE ERROR(137);                             PASCP 2619
2860:                               TEST := SY <> COMMA;                             PASCP 2620
```

```
2861:                             IF NOT TEST THEN INSYMBOL                                PASCP 2621
2862:                           UNTIL TEST;                                                PASCP 2622
2863:                           IF SY = RBRACK THEN INSYMBOL ELSE ERROR(12)                PASCP 2623
2864:                         END;                                                         PASCP 2624
2865:                       IF VARPART THEN                                                PASCP 2625
2866:                         BEGIN                                                        PASCP 2626
2867:                           IF CSTPART <> [ ] THEN                                     PASCP 2627
2868:                             BEGIN NEW(LVP,PSET); LVP^.PVAL := CSTPART;               PASCP 2628
2869:                               LVP^.CCLASS := PSET;                                   PASCP 2629
2870:                               IF CSTPTRIX = CSTOCCMAX THEN ERROR(254)                PASCP 2630
2871:                               ELSE                                                   PASCP 2631
2872:                                 BEGIN CSTPTRIX := CSTPTRIX + 1;                      PASCP 2632
2873:                                   CSTPTR[CSTPTRIX] := LVP;                           PASCP 2633
2874:                                   GEN2(51(*LDC*),5,CSTPTRIX);                        PASCP 2634
2875:                                   GEN0(28(*UNI*)); GATTR.KIND := EXPR                PASCP 2635
2876:                                 END                                                  PASCP 2636
2877:                             END                                                      PASCP 2637
2878:                         END                                                          PASCP 2638
2879:                       ELSE                                                           PASCP 2639
2880:                         BEGIN NEW(LVP,PSET); LVP^.PVAL := CSTPART;                   PASCP 2640
2881:                           LVP^.CCLASS := PSET;                                       PASCP 2641
2882:                           GATTR.CVAL.VALP := LVP                                     PASCP 2642
2883:                         END                                                          PASCP 2643
2884:                     END                                                              PASCP 2644
2885:                   END (*CASE*) ;                                                     PASCP 2645
2886:                   IF NOT (SY IN FSYS) THEN                                           PASCP 2646
2887:                     BEGIN ERROR(6); SKIP(FSYS + FACBEGSYS) END                       PASCP 2647
2888:                 END (*WHILE*)                                                        PASCP 2648
2889:               END (*FACTOR*) ;                                                       PASCP 2649
2890:                                                                                      PASCP 2650
2891:             BEGIN (*TERM*)                                                           PASCP 2651
2892:               FACTOR(FSYS + [MULOP]);                                                PASCP 2652
2893:               WHILE SY = MULOP DO                                                    PASCP 2653
2894:                       BEGIN LOAD; LATTR := GATTR; LOP := OP;                         PASCP 2654
2895:                   INSYMBOL; FACTOR(FSYS + [MULOP]); LOAD;                            PASCP 2655
2896:                   IF (LATTR.TYPTR <> NIL) AND (GATTR.TYPTR <> NIL) THEN              PASCP 2656
2897:                     CASE LOP OF                                                      PASCP 2657
2898:             (***)       MUL:  IF (LATTR.TYPTR=INTPTR)AND(GATTR.TYPTR=INTPTR)       PASCP 2658
2899:                               THEN GEN0(15(*MPI*))                                   PASCP 2659
2900:                             ELSE                                                     PASCP 2660
2901:                               BEGIN                                                  PASCP 2661
2902:                                 IF LATTR.TYPTR = INTPTR THEN                         PASCP 2662
2903:                                   BEGIN GEN0(9(*FLO*));                              PASCP 2663
2904:                                     LATTR.TYPTR := REALPTR                           PASCP 2664
2905:                                   END                                                PASCP 2665
2906:                                 ELSE                                                 PASCP 2666
2907:                                   IF GATTR.TYPTR = INTPTR THEN                       PASCP 2667
2908:                                     BEGIN GEN0(10(*FLT*));                           PASCP 2668
2909:                                       GATTR.TYPTR := REALPTR                         PASCP 2669
2910:                                     END;                                             PASCP 2670
2911:                                 IF (LATTR.TYPTR = REALPTR)                           PASCP 2671
2912:                                   AND(GATTR.TYPTR=REALPTR)THEN GEN0(16(*MPR*))       PASCP 2672
2913:                                 ELSE                                                 PASCP 2673
2914:                                   IF(LATTR.TYPTR^.FORM=POWER)                        PASCP 2674
2915:                                     AND COMPTYPES(LATTR.TYPTR,GATTR.TYPTR)THEN       PASCP 2675
2916:                                     GEN0(12(*INT*))                                  PASCP 2676
2917:                                   ELSE BEGIN ERROR(134);GATTR.TYPTR:=NIL END         PASCP 2677
2918:                               END;                                                   PASCP 2678
2919:             (*/*)       RDIV: BEGIN                                                PASCP 2679
2920:                               IF GATTR.TYPTR = INTPTR THEN                           PASCP 2684
2921:                                 BEGIN GEN0(10(*FLT*));                               PASCP 2685
2922:                                 GATTR.TYPTR := REALPTR                               PASCP 2686
2923:                               END;                                                   PASCP 2687
2924:                               IF LATTR.TYPTR = INTPTR THEN                           J       7
2925:                                 BEGIN GEN0(9(*FLO*));                                J       8
```

```
2926:                                LATTR.TYPTR := REALPTR                          J       9
2927:                              END;                                              J      10
2928:                            IF (LATTR.TYPTR = REALPTR)                          PASCP 2688
2929:                              AND (GATTR.TYPTR=REALPTR)THEN GEN0(7(*DVR*))      PASCP 2689
2930:                            ELSE BEGIN ERROR(134); GATTR.TYPTR := NIL END       PASCP 2690
2931:                          END;                                                  PASCP 2691
2932:            (*DIV*)     IDIV: IF (LATTR.TYPTR = INTPTR)                         PASCP 2692
2933:                            AND (GATTR.TYPTR = INTPTR) THEN GEN0(6(*DVI*))      PASCP 2693
2934:                          ELSE BEGIN ERROR(134); GATTR.TYPTR := NIL END;        PASCP 2694
2935:            (*MOD*)     IMOD: IF (LATTR.TYPTR = INTPTR)                         PASCP 2695
2936:                            AND (GATTR.TYPTR = INTPTR) THEN GEN0(14(*MOD*))     PASCP 2696
2937:                          ELSE BEGIN ERROR(134); GATTR.TYPTR := NIL END;        PASCP 2697
2938:            (*AND*)     ANDOP:IF (LATTR.TYPTR = BOOLPTR)                        PASCP 2698
2939:                            AND (GATTR.TYPTR = BOOLPTR) THEN GEN0(4(*AND*))     PASCP 2699
2940:                          ELSE BEGIN ERROR(134); GATTR.TYPTR := NIL END         PASCP 2700
2941:                    END (*CASE*)                                                PASCP 2701
2942:                  ELSE GATTR.TYPTR := NIL                                       PASCP 2702
2943:                END (*WHILE*)                                                   PASCP 2703
2944:            END (*TERM*) ;                                                      PASCP 2704
2945:                                                                                PASCP 2705
2946:          BEGIN (*SIMPLEEXPRESSION*)                                            PASCP 2706
2947:            SIGNED := FALSE;                                                    PASCP 2707
2948:            IF (SY = ADDOP) AND (OP IN [PLUS,MINUS]) THEN                       PASCP 2708
2949:              BEGIN SIGNED := OP = MINUS; INSYMBOL END;                         PASCP 2709
2950:            TERM(FSYS + [ADDOP]);                                               PASCP 2710
2951:            IF SIGNED THEN                                                      PASCP 2711
2952:              BEGIN LOAD;                                                       PASCP 2712
2953:                IF GATTR.TYPTR = INTPTR THEN GEN0(17(*NGI*))                    PASCP 2713
2954:                ELSE                                                            PASCP 2714
2955:                  IF GATTR.TYPTR = REALPTR THEN GEN0(18(*NGR*))                 PASCP 2715
2956:                  ELSE BEGIN ERROR(134); GATTR.TYPTR := NIL END                 PASCP 2716
2957:              END;                                                              PASCP 2717
2958:            WHILE SY = ADDOP DO                                                 PASCP 2718
2959:              BEGIN LOAD; LATTR := GATTR; LOP := OP;                            PASCP 2719
2960:                INSYMBOL; TERM(FSYS + [ADDOP]); LOAD;                           PASCP 2720
2961:                IF (LATTR.TYPTR <> NIL) AND (GATTR.TYPTR <> NIL) THEN           PASCP 2721
2962:                  CASE LOP OF                                                   PASCP 2722
2963:          (*+*)       PLUS:                                                     PASCP 2723
2964:                      IF (LATTR.TYPTR = INTPTR)AND(GATTR.TYPTR = INTPTR) THEN   PASCP 2724
2965:                        GEN0(2(*ADI*))                                          PASCP 2725
2966:                      ELSE                                                      PASCP 2726
2967:                        BEGIN                                                   PASCP 2727
2968:                          IF LATTR.TYPTR = INTPTR THEN                          PASCP 2728
2969:                            BEGIN GEN0(9(*FLO*));                               PASCP 2729
2970:                              LATTR.TYPTR := REALPTR                            PASCP 2730
2971:                            END                                                 PASCP 2731
2972:                          ELSE                                                  PASCP 2732
2973:                            IF GATTR.TYPTR = INTPTR THEN                        PASCP 2733
2974:                              BEGIN GEN0(10(*FLT*));                            PASCP 2734
2975:                                GATTR.TYPTR := REALPTR                          PASCP 2735
2976:                              END;                                              PASCP 2736
2977:                          IF (LATTR.TYPTR = REALPTR)AND(GATTR.TYPTR = REALPTR)  PASCP 2737
2978:                            THEN GEN0(3(*ADR*))                                 PASCP 2738
2979:                          ELSE IF(LATTR.TYPTR^.FORM=POWER)                      PASCP 2739
2980:                                 AND COMPTYPES(LATTR.TYPTR,GATTR.TYPTR) THEN    PASCP 2740
2981:                                 GEN0(28(*UNI*))                                PASCP 2741
2982:                               ELSE BEGIN ERROR(134);GATTR.TYPTR:=NIL END       PASCP 2742
2983:                        END;                                                    PASCP 2743
2984:          (*-*)       MINUS:                                                    PASCP 2744
2985:                      IF (LATTR.TYPTR = INTPTR)AND(GATTR.TYPTR = INTPTR) THEN   PASCP 2745
2986:                        GEN0(21(*SBI*))                                         PASCP 2746
2987:                      ELSE                                                      PASCP 2747
2988:                        BEGIN                                                   PASCP 2748
2989:                          IF LATTR.TYPTR = INTPTR THEN                          PASCP 2749
2990:                            BEGIN GEN0(9(*FLO*));                               PASCP 2750
```

```
2991:                             LATTR.TYPTR := REALPTR                                PASCP 2751
2992:                           END                                                     PASCP 2752
2993:                         ELSE                                                      PASCP 2753
2994:                           IF GATTR.TYPTR = INTPTR THEN                            PASCP 2754
2995:                           BEGIN GEN0(10(*FLT*));                                  PASCP 2755
2996:                               GATTR.TYPTR := REALPTR                              PASCP 2756
2997:                           END;                                                    PASCP 2757
2998:                         IF (LATTR.TYPTR = REALPTR)AND(GATTR.TYPTR = REALPTR)      PASCP 2758
2999:                           THEN GEN0(22(*SBR*))                                    PASCP 2759
3000:                         ELSE                                                      PASCP 2760
3001:                           IF (LATTR.TYPTR^.FORM = POWER)                          PASCP 2761
3002:                             AND COMPTYPES(LATTR.TYPTR,GATTR.TYPTR) THEN           PASCP 2762
3003:                             GEN0(5(*DIF*))                                        PASCP 2763
3004:                           ELSE BEGIN ERROR(134); GATTR.TYPTR := NIL END           PASCP 2764
3005:                       END;                                                        PASCP 2765
3006:           (*OR*)      OROP:                                                       PASCP 2766
3007:                       IF(LATTR.TYPTR=BOOLPTR)AND(GATTR.TYPTR=BOOLPTR)THEN         PASCP 2767
3008:                         GEN0(13(*IOR*))                                           PASCP 2768
3009:                       ELSE BEGIN ERROR(134); GATTR.TYPTR := NIL END             PASCP 2769
3010:                   END (*CASE*)                                                    PASCP 2770
3011:                 ELSE GATTR.TYPTR := NIL                                           PASCP 2771
3012:               END (*WHILE*)                                                       PASCP 2772
3013:           END (*SIMPLEEXPRESSION*) ;                                              PASCP 2773
3014:                                                                                   PASCP 2774
3015:         BEGIN (*EXPRESSION*)                                                      PASCP 2775
3016:           SIMPLEEXPRESSION(FSYS + [RELOP]);                                       PASCP 2776
3017:           IF SY = RELOP THEN                                                      PASCP 2777
3018:             BEGIN                                                                 PASCP 2778
3019:               IF GATTR.TYPTR <> NIL THEN                                          PASCP 2779
3020:                 IF GATTR.TYPTR^.FORM <= POWER THEN LOAD                           PASCP 2780
3021:                 ELSE LOADADDRESS;                                                 PASCP 2781
3022:                 LATTR := GATTR; LOP := OP;                                        PASCP 2782
3023:               IF LOP = INOP THEN                                                  P      460
3024:                 IF NOT COMPTYPES(GATTR.TYPTR,INTPTR) THEN                         P      461
3025:                   GEN0T(58(*ORD*),GATTR.TYPTR);                                   P      462
3026:               INSYMBOL; SIMPLEEXPRESSION(FSYS);                                   PASCP 2783
3027:               IF GATTR.TYPTR <> NIL THEN                                          PASCP 2784
3028:                 IF GATTR.TYPTR^.FORM <= POWER THEN LOAD                           PASCP 2785
3029:                 ELSE LOADADDRESS;                                                 PASCP 2786
3030:               IF (LATTR.TYPTR <> NIL) AND (GATTR.TYPTR <> NIL) THEN               PASCP 2787
3031:                 IF LOP = INOP THEN                                                PASCP 2788
3032:                   IF GATTR.TYPTR^.FORM = POWER THEN                               PASCP 2789
3033:                     IF COMPTYPES(LATTR.TYPTR,GATTR.TYPTR^.ELSET) THEN             PASCP 2790
3034:                       GEN0(11(*INN*))                                             PASCP 2791
3035:                     ELSE BEGIN ERROR(129); GATTR.TYPTR := NIL END                 PASCP 2792
3036:                   ELSE BEGIN ERROR(130); GATTR.TYPTR := NIL END                   PASCP 2793
3037:                 ELSE                                                              PASCP 2794
3038:                   BEGIN                                                           PASCP 2795
3039:                     IF LATTR.TYPTR <> GATTR.TYPTR THEN                            PASCP 2796
3040:                       IF LATTR.TYPTR = INTPTR THEN                                PASCP 2797
3041:                         BEGIN GEN0(9(*FLO*));                                     PASCP 2798
3042:                           LATTR.TYPTR := REALPTR                                  PASCP 2799
3043:                         END                                                       PASCP 2800
3044:                       ELSE                                                        PASCP 2801
3045:                         IF GATTR.TYPTR = INTPTR THEN                              PASCP 2802
3046:                           BEGIN GEN0(10(*FLT*));                                  PASCP 2803
3047:                             GATTR.TYPTR := REALPTR                                PASCP 2804
3048:                           END;                                                    PASCP 2805
3049:                     IF COMPTYPES(LATTR.TYPTR,GATTR.TYPTR) THEN                    PASCP 2806
3050:                       BEGIN LSIZE := LATTR.TYPTR^.SIZE;                           PASCP 2807
3051:                         CASE LATTR.TYPTR^.FORM OF                                 PASCP 2808
3052:                           SCALAR:                                                 PASCP 2809
3053:                             IF LATTR.TYPTR = REALPTR THEN TYPIND := 'R'           PASCP 2810
3054:                             ELSE                                                  PASCP 2811
3055:                               IF LATTR.TYPTR = BOOLPTR THEN TYPIND := 'B'         PASCP 2812
```

```
3056:                             ELSE                                                    P      463
3057:                               IF LATTR.TYPTR = CHARPTR THEN TYPIND := 'C'            P      464
3058:                               ELSE TYPIND := 'I';                                   P      465
3059:                           POINTER:                                                  PASCP 2814
3060:                             BEGIN                                                   PASCP 2815
3061:                               IF LOP IN [LTOP,LEOP,GTOP,GEOP] THEN ERROR(131);      PASCP 2816
3062:                               TYPIND := 'A'                                         PASCP 2817
3063:                             END;                                                    PASCP 2818
3064:                           POWER:                                                    PASCP 2819
3065:                             BEGIN IF LOP IN [LTOP,GTOP] THEN ERROR(132);          PASCP 2820
3066:                               TYPIND := 'S'                                         PASCP 2821
3067:                           END;                                                      PASCP 2822
3068:                           ARRAYS:                                                   PASCP 2823
3069:                             BEGIN                                                   PASCP 2824
3070:                               IF NOT STRING(LATTR.TYPTR)                            PASCP 2825
3071:                               AND(LOP IN[LTOP,LEOP,GTOP,GEOP])THEN ERROR(131);      PASCP 2826
3072:                               TYPIND := 'M'                                         PASCP 2827
3073:                             END;                                                    PASCP 2828
3074:                           RECORDS:                                                  PASCP 2829
3075:                             BEGIN                                                   PASCP 2830
3076:                               IF LOP IN [LTOP,LEOP,GTOP,GEOP] THEN ERROR(131);      PASCP 2831
3077:                               TYPIND := 'M'                                         PASCP 2832
3078:                             END;                                                    PASCP 2833
3079:                           FILES:                                                    PASCP 2834
3080:                             BEGIN ERROR(133); TYPIND := 'F' END                     PASCP 2835
3081:                         END;                                                        PASCP 2836
3082:                         CASE LOP OF                                                 PASCP 2837
3083:                           LTOP: GEN2(53(*LES*),ORD(TYPIND),LSIZE);                  PASCP 2838
3084:                           LEOP: GEN2(52(*LEQ*),ORD(TYPIND),LSIZE);                  PASCP 2839
3085:                           GTOP: GEN2(49(*GRT*),ORD(TYPIND),LSIZE);                  PASCP 2840
3086:                           GEOP: GEN2(48(*GEQ*),ORD(TYPIND),LSIZE);                  PASCP 2841
3087:                           NEOP: GEN2(55(*NEQ*),ORD(TYPIND),LSIZE);                  PASCP 2842
3088:                           EQOP: GEN2(47(*EQU*),ORD(TYPIND),LSIZE)                   PASCP 2843
3089:                         END                                                         PASCP 2844
3090:                       END                                                           PASCP 2845
3091:                     ELSE ERROR(129)                                                 PASCP 2846
3092:                   END;                                                            PASCP 2847
3093:               GATTR.TYPTR := BOOLPTR; GATTR.KIND := EXPR                          PASCP 2848
3094:             END (*SY = RELOP*)                                                    PASCP 2849
3095:         END (*EXPRESSION*) ;                                                      PASCP 2850
3096:                                                                                   PASCP 2851
3097:         PROCEDURE ASSIGNMENT(FCP: CTP);                                           PASCP 2852
3098:           VAR LATTR: ATTR;                                                        PASCP 2853
3099:         BEGIN SELECTOR(FSYS + [BECOMES],FCP);                                     PASCP 2854
3100:           IF SY = BECOMES THEN                                                    PASCP 2855
3101:             BEGIN                                                                 PASCP 2856
3102:               IF GATTR.TYPTR <> NIL THEN                                          PASCP 2857
3103:                 IF (GATTR.ACCESS<>DRCT) OR (GATTR.TYPTR^.FORM>POWER) THEN         PASCP 2858
3104:                   LOADADDRESS;                                                    PASCP 2859
3105:               LATTR := GATTR;                                                     PASCP 2860
3106:               INSYMBOL; EXPRESSION(FSYS);                                         PASCP 2861
3107:               IF GATTR.TYPTR <> NIL THEN                                          PASCP 2862
3108:                 IF GATTR.TYPTR^.FORM <= POWER THEN LOAD                           PASCP 2863
3109:                 ELSE LOADADDRESS;                                                 PASCP 2864
3110:               IF (LATTR.TYPTR <> NIL) AND (GATTR.TYPTR <> NIL) THEN               PASCP 2865
3111:                 BEGIN                                                             PASCP 2866
3112:                   IF COMPTYPES(REALPTR,LATTR.TYPTR)AND(GATTR.TYPTR=INTPTR)THEN    PASCP 2867
3113:                     BEGIN GEN0(10(*FLT*));                                        PASCP 2868
3114:                       GATTR.TYPTR := REALPTR                                      PASCP 2869
3115:                     END;                                                          PASCP 2870
3116:                   IF COMPTYPES(LATTR.TYPTR,GATTR.TYPTR) THEN                      PASCP 2871
3117:                     CASE LATTR.TYPTR^.FORM OF                                     PASCP 2872
3118:                       SCALAR,                                                     PASCP 2873
3119:                       SUBRANGE: BEGIN                                             P      466
3120:                                   IF DEBUG THEN CHECKBNDS(LATTR.TYPTR);           P      467
```

```
3121:                                 STORE(LATTR)                                  P      468
3122:                               END;                                            P      469
3123:                         POINTER: BEGIN                                        P      470
3124:                                   IF DEBUG THEN                               P      471
3125:                                     GEN2T(45(*CHK*),0,MAXADDR,NILPTR);        P      472
3126:                                   STORE(LATTR)                                P      473
3127:                                 END;                                          P      474
3128:                         POWER:   STORE(LATTR);                                PASCP 2876
3129:                         ARRAYS,                                               PASCP 2877
3130:                         RECORDS: GEN1(40(*MOV*),LATTR.TYPTR^.SIZE);           PASCP 2878
3131:                         FILES: ERROR(146)                                     PASCP 2879
3132:                       END                                                     PASCP 2880
3133:                     ELSE ERROR(129)                                           PASCP 2881
3134:                   END                                                         PASCP 2882
3135:               END (*SY = BECOMES*)                                            PASCP 2883
3136:             ELSE ERROR(51)                                                    PASCP 2884
3137:           END (*ASSIGNMENT*) ;                                                PASCP 2885
3138:                                                                               PASCP 2886
3139:           PROCEDURE GOTOSTATEMENT;                                            PASCP 2887
3140:             VAR LLP: LBP; FOUND: BOOLEAN; TTOP,TTOP1: DISPRANGE;              PASCP 2888
3141:           BEGIN                                                               PASCP 2889
3142:             IF SY = INTCONST THEN                                             PASCP 2890
3143:               BEGIN                                                           PASCP 2891
3144:                 FOUND := FALSE;                                               PASCP 2892
3145:                 TTOP := TOP;                                                  PASCP 2893
3146:                 REPEAT                                                        PASCP 2894
3147:                   WHILE DISPLAY[TTOP].OCCUR <> BLCK DO TTOP := TTOP - 1;      PASCP 2895
3148:                   TTOP1 := TTOP; LLP := DISPLAY[TTOP].FLABEL;                 PASCP 2896
3149:                   WHILE (LLP <> NIL) AND NOT FOUND DO                         PASCP 2897
3150:                     WITH LLP^ DO                                              PASCP 2898
3151:                       IF LABVAL = VAL.IVAL THEN                               PASCP 2899
3152:                         BEGIN FOUND := TRUE;                                  PASCP 2900
3153:                           IF TTOP = TTOP1 THEN                                PASCP 2901
3154:                             GENUJPXJP(57(*UJP*),LABNAME)                      P      475
3155:                           ELSE (*GOTO LEADS OUT OF PROCEDURE*) ERROR(399)     PASCP 2903
3156:                         END                                                   PASCP 2904
3157:                       ELSE LLP := NEXTLAB;                                    PASCP 2905
3158:                   TTOP := TTOP - 1                                            PASCP 2906
3159:                 UNTIL FOUND OR (TTOP = 0);                                    PASCP 2907
3160:                 IF NOT FOUND THEN ERROR(167);                                 PASCP 2908
3161:                 INSYMBOL                                                      PASCP 2909
3162:               END                                                             PASCP 2910
3163:             ELSE ERROR(15)                                                    PASCP 2911
3164:           END (*GOTOSTATEMENT*) ;                                             PASCP 2912
3165:                                                                               PASCP 2913
3166:           PROCEDURE COMPOUNDSTATEMENT;                                        PASCP 2914
3167:           BEGIN                                                               PASCP 2915
3168:             REPEAT                                                            PASCP 2916
3169:               REPEAT STATEMENT(FSYS + [SEMICOLON,ENDSY])                      PASCP 2917
3170:               UNTIL NOT (SY IN STATBEGSYS);                                   PASCP 2918
3171:               TEST := SY <> SEMICOLON;                                        PASCP 2919
3172:               IF NOT TEST THEN INSYMBOL                                       PASCP 2920
3173:             UNTIL TEST;                                                       PASCP 2921
3174:             IF SY = ENDSY THEN INSYMBOL ELSE ERROR(13)                        PASCP 2922
3175:           END (*COMPOUNDSTATEMENET*) ;                                        PASCP 2923
3176:                                                                               PASCP 2924
3177:           PROCEDURE IFSTATEMENT;                                              PASCP 2925
3178:             VAR LCIX1,LCIX2: INTEGER;                                         PASCP 2926
3179:           BEGIN EXPRESSION(FSYS + [THENSY]);                                  PASCP 2927
3180:             GENLABEL(LCIX1); GENFJP(LCIX1);                                   PASCP 2928
3181:             IF SY = THENSY THEN INSYMBOL ELSE ERROR(52);                      PASCP 2929
3182:             STATEMENT(FSYS + [ELSESY]);                                       PASCP 2931
3183:             IF SY = ELSESY THEN                                               PASCP 2932
3184:               BEGIN GENLABEL(LCIX2); GENUJPXJP(57(*UJP*),LCIX2);              P      476
3185:                 PUTLABEL(LCIX1);                                              PASCP 2934
```

```
3186:               INSYMBOL; STATEMENT(FSYS);                                  PASCP 2935
3187:               PUTLABEL(LCIX2)                                             PASCP 2936
3188:             END                                                           PASCP 2937
3189:           ELSE PUTLABEL(LCIX1)                                            PASCP 2938
3190:         END (*IFSTATEMENT*) ;                                             PASCP 2939
3191:                                                                           PASCP 2940
3192:         PROCEDURE CASESTATEMENT;                                          PASCP 2941
3193:           LABEL 1;                                                        PASCP 2942
3194:           TYPE CIP = ^CASEINFO;                                           PASCP 2943
3195:                CASEINFO = PACKED                                          PASCP 2944
3196:                           RECORD NEXT: CIP;                               PASCP 2945
3197:                             CSSTART: INTEGER;                             PASCP 2946
3198:                             CSLAB: INTEGER                                PASCP 2947
3199:                           END;                                            PASCP 2948
3200:           VAR LSP,LSP1: STP; FSTPTR,LPT1,LPT2,LPT3: CIP; LVAL: VALU;      PASCP 2949
3201:               LADDR, LCIX, LCIX1, LMIN, LMAX: INTEGER;                    PASCP 2950
3202:         BEGIN EXPRESSION(FSYS + [OFSY,COMMA,COLON]);                      PASCP 2951
3203:           LOAD; GENLABEL(LCIX); GENUJPXJP(57(*UJP*),LCIX);                P      477
3204:           LSP := GATTR.TYPTR;                                             PASCP 2953
3205:           IF LSP <> NIL THEN                                              PASCP 2954
3206:             IF (LSP^.FORM <> SCALAR) OR (LSP = REALPTR) THEN              PASCP 2955
3207:              BEGIN ERROR(144); LSP := NIL END                             P      478
3208:            ELSE IF NOT COMPTYPES(LSP,INTPTR) THEN GEN0T(58(*ORD*),LSP);  P      479
3209:           IF SY = OFSY THEN INSYMBOL ELSE ERROR(8);                       PASCP 2957
3210:           FSTPTR := NIL; GENLABEL(LADDR);                                 PASCP 2958
3211:           REPEAT                                                          PASCP 2959
3212:             LPT3 := NIL; GENLABEL(LCIX1);                                 PASCP 2960
3213:             IF NOT(SY IN [SEMICOLON,ENDSY]) THEN                          P      480
3214:             BEGIN                                                         P      481
3215:             REPEAT CONSTANT(FSYS + [COMMA,COLON],LSP1,LVAL);              PASCP 2961
3216:               IF LSP <> NIL THEN                                          PASCP 2962
3217:                 IF COMPTYPES(LSP,LSP1) THEN                               PASCP 2963
3218:                   BEGIN LPT1 := FSTPTR; LPT2 := NIL;                      PASCP 2964
3219:                     WHILE LPT1 <> NIL DO                                  PASCP 2965
3220:                       WITH LPT1^ DO                                       PASCP 2966
3221:                         BEGIN                                             PASCP 2967
3222:                           IF CSLAB <= LVAL.IVAL THEN                      PASCP 2968
3223:                             BEGIN IF CSLAB = LVAL.IVAL THEN ERROR(156);   PASCP 2969
3224:                               GOTO 1                                      PASCP 2970
3225:                             END;                                          PASCP 2971
3226:                           LPT2 := LPT1; LPT1 := NEXT                      PASCP 2972
3227:                         END;                                              PASCP 2973
3228:         1:              NEW(LPT3);                                        PASCP 2974
3229:                         WITH LPT3^ DO                                     PASCP 2975
3230:                           BEGIN NEXT := LPT1; CSLAB := LVAL.IVAL;         PASCP 2976
3231:                             CSSTART := LCIX1                              PASCP 2977
3232:                           END;                                            PASCP 2978
3233:                         IF LPT2 = NIL THEN FSTPTR := LPT3                 PASCP 2979
3234:                         ELSE LPT2^.NEXT := LPT3                           PASCP 2980
3235:                       END                                                 PASCP 2981
3236:                     ELSE ERROR(147);                                      PASCP 2982
3237:               TEST := SY <> COMMA;                                        PASCP 2983
3238:               IF NOT TEST THEN INSYMBOL                                   PASCP 2984
3239:             UNTIL TEST;                                                   PASCP 2985
3240:             IF SY = COLON THEN INSYMBOL ELSE ERROR(5);                    PASCP 2986
3241:             PUTLABEL(LCIX1);                                              PASCP 2987
3242:             REPEAT STATEMENT(FSYS + [SEMICOLON])                          PASCP 2988
3243:             UNTIL NOT (SY IN STATBEGSYS);                                 PASCP 2989
3244:             IF LPT3 <> NIL THEN                                           PASCP 2990
3245:               GENUJPXJP(57(*UJP*),LADDR);                                 P      482
3246:             END;                                                          P      483
3247:             TEST := SY <> SEMICOLON;                                      PASCP 2992
3248:             IF NOT TEST THEN INSYMBOL                                     PASCP 2993
3249:           UNTIL TEST;                                                     PASCP 2994
3250:           PUTLABEL(LCIX);                                                 PASCP 2995
```

```
3251:           IF FSTPTR <> NIL THEN                                                    PASCP 2996
3252:             BEGIN LMAX := FSTPTR^.CSLAB;                                           PASCP 2997
3253:               (*REVERSE POINTERS*)                                                 PASCP 2998
3254:               LPT1 := FSTPTR; FSTPTR := NIL;                                       PASCP 2999
3255:               REPEAT LPT2 := LPT1^.NEXT; LPT1^.NEXT := FSTPTR;                     PASCP 3000
3256:                 FSTPTR := LPT1; LPT1 := LPT2                                       PASCP 3001
3257:               UNTIL LPT1 = NIL;                                                    PASCP 3002
3258:               LMIN := FSTPTR^.CSLAB;                                               PASCP 3003
3259:               IF LMAX - LMIN < CIXMAX THEN                                         PASCP 3004
3260:                 BEGIN                                                              P      484
3261:                   GEN2T(45(*CHK*),LMIN,LMAX,INTPTR);                               P      485
3262:                   GEN2(51(*LDC*),1,LMIN); GEN0(21(*SBI*)); GENLABEL(LCIX);         PASCP 3011
3263:                   GENUJPXJP(44(*XJP*),LCIX); PUTLABEL(LCIX);                       P      486
3264:                   REPEAT                                                           PASCP 3013
3265:                     WITH FSTPTR^ DO                                                PASCP 3014
3266:                       BEGIN                                                        PASCP 3015
3267:                         WHILE CSLAB > LMIN DO                                      PASCP 3016
3268:                            BEGIN GEN0(60(*UJC ERROR*));                            P      487
3269:                              LMIN := LMIN+1                                        P      488
3270:                            END;                                                    P      489
3271:                         GENUJPXJP(57(*UJP*),CSSTART);                              P      490
3272:                         FSTPTR := NEXT; LMIN := LMIN + 1                           PASCP 3019
3273:                       END                                                          PASCP 3020
3274:                   UNTIL FSTPTR = NIL;                                              PASCP 3021
3275:                   PUTLABEL(LADDR)                                                  PASCP 3022
3276:                 END                                                              PASCP 3023
3277:               ELSE ERROR(157)                                                      PASCP 3024
3278:             END;                                                                   PASCP 3025
3279:             IF SY = ENDSY THEN INSYMBOL ELSE ERROR(13)                             PASCP 3026
3280:         END (*CASESTATEMENT*) ;                                                    PASCP 3027
3281:                                                                                    PASCP 3028
3282:         PROCEDURE REPEATSTATEMENT;                                                 PASCP 3029
3283:           VAR LADDR: INTEGER;                                                      PASCP 3030
3284:         BEGIN GENLABEL(LADDR); PUTLABEL(LADDR);                                    PASCP 3031
3285:           REPEAT STATEMENT(FSYS + [SEMICOLON,UNTILSY]);                            P      491
3286:             IF SY IN STATBEGSYS THEN ERROR(14)                                     P      492
3287:           UNTIL NOT(SY IN STATBEGSYS);                                             P      493
3288:           WHILE SY = SEMICOLON DO                                                  P      494
3289:             BEGIN INSYMBOL;                                                        P      495
3290:               REPEAT STATEMENT(FSYS + [SEMICOLON,UNTILSY]);                        P      496
3291:                 IF SY IN STATBEGSYS THEN ERROR(14)                                 P      497
3292:               UNTIL NOT (SY IN STATBEGSYS);                                        P      498
3293:             END;                                                                   P      499
3294:           IF SY = UNTILSY THEN                                                     PASCP 3038
3295:             BEGIN INSYMBOL; EXPRESSION(FSYS); GENFJP(LADDR)                        PASCP 3039
3296:             END                                                                    PASCP 3040
3297:           ELSE ERROR(53)                                                           PASCP 3041
3298:         END (*REPEATSTATEMENT*) ;                                                  PASCP 3042
3299:                                                                                    PASCP 3043
3300:         PROCEDURE WHILESTATEMENT;                                                  PASCP 3044
3301:           VAR LADDR, LCIX: INTEGER;                                                PASCP 3045
3302:         BEGIN GENLABEL(LADDR); PUTLABEL(LADDR);                                    PASCP 3046
3303:           EXPRESSION(FSYS + [DOSY]); GENLABEL(LCIX); GENFJP(LCIX);                 PASCP 3047
3304:           IF SY = DOSY THEN INSYMBOL ELSE ERROR(54);                               PASCP 3048
3305:           STATEMENT(FSYS); GENUJPXJP(57(*IJP*),LADDR); PUTLABEL(LCIX)              P      500
3306:         END (*WHILESTATEMENT*) ;                                                   PASCP 3050
3307:                                                                                    PASCP 3051
3308:         PROCEDURE FORSTATEMENT;                                                    PASCP 3052
3309:           VAR LATTR: ATTR; LSP: STP;  LSY: SYMBOL;                                 PASCP 3053
3310:               LCIX, LADDR: INTEGER;                                                PASCP 3054
3311:                     LLC: ADDRRANGE;                                                P      501
3312:         BEGIN LLC := LC;                                                           P      502
3313:           WITH LATTR DO                                                            P      503
3314:             BEGIN TYPTR := NIL; KIND := VARBL;                                     P      504
3315:               ACCESS := DRCT; VLEVEL := LEVEL; DPLMT := 0                          P      505
```

```
3316:             END;                                                                P      506
3317:           IF SY = IDENT THEN                                                    PASCP 3056
3318:             BEGIN SEARCHID([VARS],LCP);                                         PASCP 3057
3319:               WITH LCP^, LATTR DO                                               PASCP 3058
3320:                 BEGIN TYPTR := IDTYPE; KIND := VARBL;                           PASCP 3059
3321:                   IF VKIND = ACTUAL THEN                                        PASCP 3060
3322:                     BEGIN ACCESS := DRCT; VLEVEL := VLEV;                       PASCP 3061
3323:                       DPLMT := VADDR                                            PASCP 3062
3324:                     END                                                         PASCP 3063
3325:                   ELSE BEGIN ERROR(155); TYPTR := NIL END                       PASCP 3064
3326:                 END;                                                            PASCP 3065
3327:               IF LATTR.TYPTR <> NIL THEN                                        PASCP 3066
3328:                 IF (LATTR.TYPTR^.FORM > SUBRANGE)                               PASCP 3067
3329:                    OR COMPTYPES(REALPTR,LATTR.TYPTR) THEN                       PASCP 3068
3330:                   BEGIN ERROR(143); LATTR.TYPTR := NIL END;                     PASCP 3069
3331:               INSYMBOL                                                          PASCP 3070
3332:             END                                                                 PASCP 3071
3333:           ELSE                                                                  PASCP 3072
3334:             BEGIN ERROR(2); SKIP(FSYS + [BECOMES,TOSY,DOWNTOSY,DOSY]) END;      PASCP 3073
3335:           IF SY = BECOMES THEN                                                  PASCP 3074
3336:             BEGIN INSYMBOL; EXPRESSION(FSYS + [TOSY,DOWNTOSY,DOSY]);            PASCP 3075
3337:               IF GATTR.TYPTR <> NIL THEN                                        PASCP 3076
3338:                   IF GATTR.TYPTR^.FORM <> SCALAR THEN ERROR(144)                PASCP 3077
3339:                   ELSE                                                          PASCP 3078
3340:                     IF COMPTYPES(LATTR.TYPTR,GATTR.TYPTR) THEN                  PASCP 3079
3341:                       BEGIN LOAD; STORE(LATTR) END                              PASCP 3080
3342:                     ELSE ERROR(145)                                             PASCP 3081
3343:             END                                                                 PASCP 3082
3344:           ELSE                                                                  PASCP 3083
3345:             BEGIN ERROR(51); SKIP(FSYS + [TOSY,DOWNTOSY,DOSY]) END;             PASCP 3084
3346:           IF SY IN [TOSY,DOWNTOSY] THEN                                         PASCP 3085
3347:             BEGIN LSY := SY; INSYMBOL; EXPRESSION(FSYS + [DOSY]);               PASCP 3086
3348:               IF GATTR.TYPTR <> NIL THEN                                        PASCP 3087
3349:               IF GATTR.TYPTR^.FORM <> SCALAR THEN ERROR(144)                    PASCP 3088
3350:                 ELSE                                                            PASCP 3089
3351:                   IF COMPTYPES(LATTR.TYPTR,GATTR.TYPTR) THEN                    PASCP 3090
3352:                     BEGIN LOAD;                                                 P      507
3353:                       IF NOT COMPTYPES(LATTR.TYPTR,INTPTR) THEN                 P      508
3354:                         GEN0T(58(*ORD*),GATTR.TYPTR);                           P      509
3355:                       ALIGN(INTPTR,LC);                                         P      510
3356:                       GEN2T(56(*STR*),0,LC,INTPTR);                             P      511
3357:                       GENLABEL(LADDR); PUTLABEL(LADDR);                         PASCP 3092
3358:                       GATTR := LATTR; LOAD;                                     P      512
3359:                       IF NOT COMPTYPES(GATTR.TYPTR,INTPTR) THEN                 P      513
3360:                         GEN0T(58(*ORD*),GATTR.TYPTR);                           P      514
3361:                       GEN2T(54(*LOD*),0,LC,INTPTR);                             P      515
3362:                       LC := LC + INTSIZE;                                       PASCP 3094
3363:                       IF LC > LCMAX THEN LCMAX := LC;                           PASCP 3095
3364:                       IF LSY = TOSY THEN GEN2(52(*LEQ*),ORD('I'),1)             PASCP 3096
3365:                       ELSE GEN2(48(*GEQ*),ORD('I'),1);                          PASCP 3097
3366:                     END                                                         PASCP 3098
3367:                   ELSE ERROR(145)                                               PASCP 3099
3368:             END                                                                 PASCP 3100
3369:           ELSE BEGIN ERROR(55); SKIP(FSYS + [DOSY]) END;                        PASCP 3101
3370:           GENLABEL(LCIX); GENUJPXJP(33(*FJP*),LCIX);                            P      516
3371:           IF SY = DOSY THEN INSYMBOL ELSE ERROR(54);                            PASCP 3103
3372:           STATEMENT(FSYS);                                                      PASCP 3104
3373:           GATTR := LATTR; LOAD;                                                 PASCP 3105
3374:           IF LSY=TOSY THEN GEN1T(34(*INC*),1,GATTR.TYPTR)                       P      517
3375:           ELSE  GEN1T(31(*DEC*),1,GATTR.TYPTR);                                 P      518
3376:           STORE(LATTR); GENUJPXJP(57(*UJP*),LADDR); PUTLABEL(LCIX);             P      519
3377:           LC := LLC;                                                            P      520
3378:         END (*FORSTATEMENT*) ;                                                  PASCP 3109
3379:                                                                                 PASCP 3110
3380:                                                                                 PASCP 3111
```

```
3381:          PROCEDURE WITHSTATEMENT;                                                    PASCP 3112
3382:            VAR LCP: CTP; LCNT1: DISPRANGE; LLC: ADDRRANGE;                           P      521
3383:          BEGIN LCNT1 := 0; LLC := LC;                                                P      522
3384:            REPEAT                                                                    PASCP 3115
3385:              IF SY = IDENT THEN                                                      PASCP 3116
3386:                BEGIN SEARCHID([VARS,FIELD],LCP); INSYMBOL END                        PASCP 3117
3387:              ELSE BEGIN ERROR(2); LCP := UVARPTR END;                                PASCP 3118
3388:              SELECTOR(FSYS + [COMMA,DOSY],LCP);                                      PASCP 3119
3389:              IF GATTR.TYPTR <> NIL THEN                                              PASCP 3120
3390:                IF GATTR.TYPTR^.FORM = RECORDS THEN                                   PASCP 3121
3391:                  IF TOP < DISPLIMIT THEN                                             PASCP 3122
3392:                    BEGIN TOP := TOP + 1; LCNT1 := LCNT1 + 1;                         PASCP 3123
3393:                      WITH DISPLAY[TOP] DO                                            PASCP 3124
3394:                        BEGIN FNAME := GATTR.TYPTR^.FSTFLD;                           PASCP 3125
3395:                          FLABEL := NIL                                               PASCP 3126
3396:                        END;                                                          PASCP 3127
3397:                      IF GATTR.ACCESS = DRCT THEN                                     PASCP 3128
3398:                        WITH DISPLAY[TOP] DO                                          PASCP 3129
3399:                          BEGIN OCCUR := CREC; CLEV := GATTR.VLEVEL;                  PASCP 3130
3400:                            CDSPL := GATTR.DPLMT                                      PASCP 3131
3401:                          END                                                         PASCP 3132
3402:                      ELSE                                                            PASCP 3133
3403:                        BEGIN LOADADDRESS;                                            P      523
3404:                          ALIGN(NILPTR,LC);                                           P      524
3405:                          GEN2T(56(*STR*),0,LC,NILPTR);                               P      525
3406:                          WITH DISPLAY[TOP] DO                                        PASCP 3135
3407:                            BEGIN OCCUR := VREC; VDSPL := LC END;                     PASCP 3136
3408:                          LC := LC+PTRSIZE;                                           P      526
3409:                          IF LC > LCMAX THEN LCMAX := LC                              PASCP 3138
3410:                        END                                                           PASCP 3139
3411:                    END                                                               PASCP 3140
3412:                  ELSE ERROR(250)                                                     PASCP 3141
3413:                ELSE ERROR(140);                                                      PASCP 3142
3414:              TEST := SY <> COMMA;                                                    PASCP 3143
3415:              IF NOT TEST THEN INSYMBOL                                               PASCP 3144
3416:            UNTIL TEST;                                                               PASCP 3145
3417:            IF SY = DOSY THEN INSYMBOL ELSE ERROR(54);                                PASCP 3146
3418:            STATEMENT(FSYS);                                                          PASCP 3147
3419:            TOP := TOP-LCNT1; LC := LLC;                                              P      527
3420:          END (*WITHSTATEMENT*) ;                                                     PASCP 3149
3421:                                                                                      PASCP 3150
3422:        BEGIN (*STATEMENT*)                                                           PASCP 3151
3423:          IF SY = INTCONST THEN (*LABEL*)                                             PASCP 3152
3424:            BEGIN LLP := DISPLAY[TOP].FLABEL;                                         PASCP 3153
3425:              WHILE LLP <> NIL DO                                                     PASCP 3154
3426:                WITH LLP^ DO                                                          PASCP 3155
3427:                  IF LABVAL = VAL.IVAL THEN                                           PASCP 3156
3428:                    BEGIN IF DEFINED THEN ERROR(165);                                 PASCP 3157
3429:                      PUTLABEL(LABNAME); DEFINED := TRUE;                             PASCP 3158
3430:                      GOTO 1                                                          PASCP 3159
3431:                    END                                                               PASCP 3160
3432:                  ELSE LLP := NEXTLAB;                                                PASCP 3161
3433:              ERROR(167);                                                             PASCP 3162
3434:        1:    INSYMBOL;                                                               PASCP 3163
3435:              IF SY = COLON THEN INSYMBOL ELSE ERROR(5)                               PASCP 3164
3436:            END;                                                                      PASCP 3165
3437:          IF NOT (SY IN FSYS + [IDENT]) THEN                                          PASCP 3166
3438:            BEGIN ERROR(6); SKIP(FSYS) END;                                           PASCP 3167
3439:          IF SY IN STATBEGSYS + [IDENT] THEN                                          PASCP 3168
3440:            BEGIN                                                                     PASCP 3169
3441:              CASE SY OF                                                              PASCP 3170
3442:                IDENT:    BEGIN SEARCHID([VARS,FIELD,FUNC,PROC],LCP); INSYMBOL;       PASCP 3171
3443:                            IF LCP^.KLASS = PROC THEN CALL(FSYS,LCP)                  PASCP 3172
3444:                            ELSE ASSIGNMENT(LCP)                                      PASCP 3173
3445:                          END;                                                        PASCP 3174
```

```
3446:                BEGINSY:  BEGIN INSYMBOL; COMPOUNDSTATEMENT END;                  PASCP 3175
3447:                GOTOSY:   BEGIN INSYMBOL; GOTOSTATEMENT END;                     PASCP 3176
3448:                IFSY:     BEGIN INSYMBOL; IFSTATEMENT END;                       PASCP 3177
3449:                CASESY:   BEGIN INSYMBOL; CASESTATEMENT END;                     PASCP 3178
3450:                WHILESY:  BEGIN INSYMBOL; WHILESTATEMENT END;                    PASCP 3179
3451:                REPEATSY: BEGIN INSYMBOL; REPEATSTATEMENT END;                   PASCP 3180
3452:                FORSY:    BEGIN INSYMBOL; FORSTATEMENT END;                      PASCP 3181
3453:                WITHSY:   BEGIN INSYMBOL; WITHSTATEMENT END                      PASCP 3182
3454:              END;                                                               PASCP 3183
3455:              IF NOT (SY IN [SEMICOLON,ENDSY,ELSESY,UNTILSY]) THEN               PASCP 3184
3456:                BEGIN ERROR(6); SKIP(FSYS) END                                   PASCP 3185
3457:            END                                                                  PASCP 3186
3458:        END (*STATEMENT*) ;                                                      PASCP 3187
3459:                                                                                 PASCP 3188
3460:      BEGIN (*BODY*)                                                             PASCP 3189
3461:        IF FPROCP <> NIL THEN ENTNAME := FPROCP^.PFNAME                          PASCP 3190
3462:        ELSE GENLABEL(ENTNAME);                                                  PASCP 3191
3463:        CSTPTRIX := 0; TOPNEW := 5; TOPMAX := 5;                                 P      528
3464:        PUTLABEL(ENTNAME); GENLABEL(SEGSIZE); GENLABEL(STACKTOP);                P      529
3465:        GENCUPENT(32(*ENT1*),1,SEGSIZE); GENCUPENT(32(*ENT2*),2,STACKTOP);       P      530
3466:        IF FPROCP <> NIL THEN (*COPY MULTIPLE VALUES INTO LOACAL CELLS*)         PASCP 3195
3467:          BEGIN LLC1 := LCAFTERMARKSTACK;                                        PASCP 3196
3468:            LCP := FPROCP^.NEXT;                                                 PASCP 3197
3469:            WHILE LCP <> NIL DO                                                  PASCP 3198
3470:              WITH LCP^ DO                                                       PASCP 3199
3471:                BEGIN                                                            PASCP 3200
3472:                  IF KLASS = VARS THEN                                           PASCP 3201
3473:                    IF IDTYPE <> NIL THEN                                        PASCP 3202
3474:                      IF (VKIND=ACTUAL) AND (IDTYPE^.FORM>POWER) THEN            P      531
3475:                        BEGIN                                                    PASCP 3204
3476:                          GEN2(50(*LDA*),0,VADDR);                               PASCP 3205
3477:                          GEN2T(54(*LOD*),0,LLC1,NILPTR);                        P      532
3478:                          GEN1(40(*MOV*),IDTYPE^.SIZE);                          PASCP 3207
3479:                          LLC1 := LLC1 + PTRSIZE                                 PASCP 3208
3480:                        END                                                      PASCP 3209
3481:                      ELSE LLC1 := LLC1 + IDTYPE^.SIZE;                          PASCP 3210
3482:                  LCP := LCP^.NEXT;                                              PASCP 3211
3483:                END;                                                             PASCP 3212
3484:          END;                                                                   PASCP 3213
3485:        LCMAX := LC;                                                             PASCP 3214
3486:        REPEAT                                                                   PASCP 3215
3487:          REPEAT STATEMENT(FSYS + [SEMICOLON,ENDSY])                             PASCP 3216
3488:          UNTIL NOT (SY IN STATBEGSYS);                                          PASCP 3217
3489:          TEST := SY <> SEMICOLON;                                               PASCP 3218
3490:          IF NOT TEST THEN INSYMBOL                                              PASCP 3219
3491:        UNTIL TEST;                                                              PASCP 3220
3492:        IF SY = ENDSY THEN INSYMBOL ELSE ERROR(13);                              PASCP 3221
3493:        LLP := DISPLAY[TOP].FLABEL; (*TEST FOR UNDEFINED LABELS*)                PASCP 3222
3494:        WHILE LLP <> NIL DO                                                      PASCP 3223
3495:          WITH LLP^ DO                                                           PASCP 3224
3496:            BEGIN                                                                PASCP 3225
3497:              IF NOT DEFINED THEN                                                PASCP 3226
3498:                BEGIN ERROR(168);                                                PASCP 3227
3499:                  WRITELN(OUTPUT); WRITELN(OUTPUT,' LABEL ',LABVAL);             PASCP 3228
3500:                 WRITE(OUTPUT,' ':CHCNT+16)                                      PASCP 3229
3501:                END;                                                             PASCP 3230
3502:              LLP := NEXTLAB                                                     PASCP 3231
3503:            END;                                                                 PASCP 3232
3504:        IF FPROCP <> NIL THEN                                                    PASCP 3233
3505:          BEGIN                                                                  PASCP 3234
3506:            IF FPROCP^.IDTYPE = NIL THEN GEN1(42(*RET*),ORD('P'))                PASCP 3235
3507:                ELSE GEN0T(42(*RET*),FPROCP^.IDTYPE);                            J       11
3508:            ALIGN(PARMPTR,LCMAX);                                                P      533
3509:            IF PRCODE THEN                                                       P      534
3510:              BEGIN WRITELN(PRR,'L',SEGSIZE:4,'=',LCMAX);                        P      535
```

```
3511:               WRITELN(PRR,'L',STACKTOP:4,'=',TOPMAX)                          P      536
3512:             END                                                               P      537
3513:         END                                                                   PASCP 3248
3514:       ELSE                                                                    PASCP 3249
3515:         BEGIN GEN1(42(*RET*),ORD('P'));                                       J       12
3516:           ALIGN(PARMPTR,LCMAX);                                               P      538
3517:           IF PRCODE THEN                                                      P      539
3518:             BEGIN WRITELN(PRR,'L',SEGSIZE:4,'=',LCMAX);                       P      540
3519:               WRITELN(PRR,'L',STACKTOP:4,'=',TOPMAX);                         P      541
3520:             WRITELN(PRR,'Q')                                                  P      542
3521:             END;                                                              P      543
3522:           IC := 0;                                                            PASCP 3254
3523:           (*GENERATE CALL OF MAIN PROGRAM; NOTE THAT THIS CALL MUST BE LOADED PASCP 3255
3524:            AT ABSOLUTE ADDRESS ZERO*)                                         PASCP 3256
3525:           GEN1(41(*MST*),0); GENCUPENT(46(*CUP*),0,ENTNAME); GEN0(29(*STP*)); P      544
3526:           IF PRCODE THEN                                                      PASCP 3258
3527:             WRITELN(PRR,'Q');                                                 P      545
3528:           SAVEID := ID;                                                       PASCP 3260
3529:           WHILE FEXTFILEP <> NIL DO                                           PASCP 3261
3530:             BEGIN                                                             PASCP 3262
3531:               WITH FEXTFILEP^ DO                                              PASCP 3263
3532:                 IF NOT ((FILENAME = 'INPUT   ') OR (FILENAME = 'OUTPUT  ') OR PASCP 3264
3533:                         (FILENAME = 'PRD     ') OR (FILENAME = 'PRR     '))  PASCP 3265
3534:                 THEN BEGIN ID := FILENAME;                                    PASCP 3266
3535:                        SEARCHID([VARS],LLCP);                                 PASCP 3267
3536:                        IF LLCP^.IDTYPE<>NIL THEN                              P      546
3537:                          IF LLCP^.IDTYPE^.FORM<>FILES THEN                    P      547
3538:                          BEGIN WRITELN(OUTPUT);                               PASCP 3272
3539:                            WRITELN(OUTPUT,' ':8,'UNDECLARED ','EXTERNAL ',    J       13
3540:                                    'FILE',FEXTFILEP^.FILENAME:8);             J       14
3541:                            WRITE(OUTPUT,' ':CHCNT+16)                         PASCP 3275
3542:                          END                                                  PASCP 3276
3543:                      END;                                                     PASCP 3277
3544:                 FEXTFILEP := FEXTFILEP^.NEXTFILE                              PASCP 3278
3545:             END;                                                              PASCP 3279
3546:           ID := SAVEID;                                                       PASCP 3280
3547:           IF PRTABLES THEN                                                    P      548
3548:             BEGIN WRITELN(OUTPUT); PRINTTABLES(TRUE)                          P      549
3549:             END                                                               P      550
3550:         END;                                                                  PASCP 3284
3551:     END (*BODY*) ;                                                            PASCP 3285
3552:                                                                               PASCP 3286
3553:   BEGIN (*BLOCK*)                                                             PASCP 3287
3554:     DP := TRUE;                                                               PASCP 3288
3555:     REPEAT                                                                    PASCP 3289
3556:       IF SY = LABELSY THEN                                                    PASCP 3290
3557:         BEGIN INSYMBOL; LABELDECLARATION END;                                 PASCP 3291
3558:       IF SY = CONSTSY THEN                                                    PASCP 3292
3559:         BEGIN INSYMBOL; CONSTDECLARATION END;                                 PASCP 3293
3560:       IF SY = TYPESY THEN                                                     PASCP 3294
3561:         BEGIN INSYMBOL; TYPEDECLARATION END;                                  PASCP 3295
3562:       IF SY = VARSY THEN                                                      PASCP 3296
3563:         BEGIN INSYMBOL; VARDECLARATION END;                                   PASCP 3297
3564:       WHILE SY IN [PROCSY,FUNCSY] DO                                          PASCP 3298
3565:         BEGIN LSY := SY; INSYMBOL; PROCDECLARATION(LSY) END;                  PASCP 3299
3566:       IF SY <> BEGINSY THEN                                                   PASCP 3300
3567:         BEGIN ERROR(18); SKIP(FSYS) END                                       PASCP 3301
3568:     UNTIL (SY IN STATBEGSYS) OR EOF(INPUT);                                   P      551
3569:     DP := FALSE;                                                              PASCP 3303
3570:     IF SY = BEGINSY THEN INSYMBOL ELSE ERROR(17);                             PASCP 3304
3571:     REPEAT BODY(FSYS + [CASESY]);                                             PASCP 3305
3572:       IF SY <> FSY THEN                                                       PASCP 3306
3573:         BEGIN ERROR(6); SKIP(FSYS) END                                        P      552
3574:     UNTIL ((SY = FSY) OR (SY IN BLOCKBEGSYS)) OR EOF(INPUT);                  P      553
3575:   END (*BLOCK*) ;                                                             PASCP 3309
```

```
3576:                                                                                PASCP 3310
3577:    PROCEDURE PROGRAMME(FSYS:SETOFSYS);                                         PASCP 3311
3578:      VAR EXTFP:EXTFILEP;                                                       PASCP 3312
3579:    BEGIN                                                                       PASCP 3313
3580:      IF SY = PROGSY THEN                                                       PASCP 3314
3581:        BEGIN INSYMBOL; IF SY <> IDENT THEN ERROR(2); INSYMBOL;                 PASCP 3315
3582:          IF NOT (SY IN [LPARENT,SEMICOLON]) THEN ERROR(14);                    PASCP 3316
3583:          IF SY = LPARENT  THEN                                                 PASCP 3317
3584:            BEGIN                                                               PASCP 3318
3585:              REPEAT INSYMBOL;                                                  PASCP 3319
3586:                IF SY = IDENT THEN                                              PASCP 3320
3587:                  BEGIN NEW(EXTFP);                                             PASCP 3321
3588:                    WITH EXTFP^ DO                                              PASCP 3322
3589:                      BEGIN FILENAME := ID; NEXTFILE := FEXTFILEP END;          PASCP 3323
3590:                    FEXTFILEP := EXTFP;                                         PASCP 3324
3591:                    INSYMBOL;                                                   PASCP 3325
3592:                    IF NOT ( SY IN [COMMA,RPARENT] ) THEN ERROR(20)             PASCP 3326
3593:                  END                                                           PASCP 3327
3594:                ELSE ERROR(2)                                                   PASCP 3328
3595:              UNTIL SY <> COMMA;                                                PASCP 3329
3596:              IF SY <> RPARENT THEN ERROR(4);                                   PASCP 3330
3597:              INSYMBOL                                                          PASCP 3331
3598:            END;                                                                PASCP 3332
3599:          IF SY <> SEMICOLON THEN ERROR(14)                                     PASCP 3333
3600:          ELSE INSYMBOL;                                                        PASCP 3334
3601:        END;                                                                    PASCP 3335
3602:      REPEAT BLOCK(FSYS,PERIOD,NIL);                                            PASCP 3336
3603:        IF SY <> PERIOD THEN ERROR(21)                                          PASCP 3337
3604:      UNTIL (SY = PERIOD) OR EOF(INPUT);                                        P      554
3605:      IF ERRINX <> 0 THEN INSYMBOL                                              P      555
3606:    END (*PROGRAMME*) ;                                                         PASCP 3339
3607:                                                                                PASCP 3340
3608:                                                                                PASCP 3341
3609:    PROCEDURE STDNAMES;                                                         PASCP 3342
3610:    BEGIN                                                                       PASCP 3343
3611:      NA[ 1] := 'FALSE   '; NA[ 2] := 'TRUE    '; NA[ 3] := 'INPUT   ';        PASCP 3344
3612:      NA[ 4] := 'OUTPUT  '; NA[ 5] := 'GET     '; NA[ 6] := 'PUT     ';        PASCP 3345
3613:      NA[ 7] := 'RESET   '; NA[ 8] := 'REWRITE '; NA[ 9] := 'READ    ';        PASCP 3346
3614:      NA[10] := 'WRITE   '; NA[11] := 'PACK    '; NA[12] := 'UNPACK  ';        PASCP 3347
3615:      NA[13] := 'NEW     '; NA[14] := 'RELEASE '; NA[15] := 'READLN  ';        PASCP 3348
3616:      NA[16] := 'WRITELN ';                                                     PASCP 3349
3617:      NA[17] := 'ABS     '; NA[18] := 'SQR     '; NA[19] := 'TRUNC   ';        PASCP 3350
3618:      NA[20] := 'ODD     '; NA[21] := 'ORD     '; NA[22] := 'CHR     ';        PASCP 3351
3619:      NA[23] := 'PRED    '; NA[24] := 'SUCC    '; NA[25] := 'EOF     ';        PASCP 3352
3620:      NA[26] := 'EOLN    ';                                                     PASCP 3353
3621:      NA[27] := 'SIN     '; NA[28] := 'COS     '; NA[29] := 'EXP     ';        PASCP 3354
3622:      NA[30] := 'SQRT    '; NA[31] := 'LN      '; NA[32] := 'ARCTAN  ';        PASCP 3355
3623:      NA[33] := 'PRD     '; NA[34] := 'PRR     '; NA[35] := 'MARK    ';        PASCP 3356
3624:    END (*STDNAMES*) ;                                                          PASCP 3357
3625:                                                                                PASCP 3358
3626:    PROCEDURE ENTERSTDTYPES;                                                    PASCP 3359
3627:      VAR SP: STP;                                                              PASCP 3360
3628:    BEGIN                                                (*TYPE UNDERLIEING:*)  PASCP 3361
3629:                                                          (******************)  PASCP 3362
3630:                                                                                PASCP 3363
3631:      NEW(INTPTR,SCALAR,STANDARD);                            (*INTEGER*)       PASCP 3364
3632:      WITH INTPTR^ DO                                                           PASCP 3365
3633:        BEGIN SIZE := INTSIZE; FORM := SCALAR; SCALKIND := STANDARD END;        PASCP 3366
3634:      NEW(REALPTR,SCALAR,STANDARD);                           (*REAL*)          PASCP 3367
3635:      WITH REALPTR^ DO                                                          PASCP 3368
3636:        BEGIN SIZE := REALSIZE; FORM := SCALAR; SCALKIND := STANDARD END;      PASCP 3369
3637:      NEW(CHARPTR,SCALAR,STANDARD);                           (*CHAR*)          PASCP 3370
3638:      WITH CHARPTR^ DO                                                          PASCP 3371
3639:        BEGIN SIZE := CHARSIZE; FORM := SCALAR; SCALKIND := STANDARD END;      PASCP 3372
3640:      NEW(BOOLPTR,SCALAR,DECLARED);                           (*BOOLEAN*)       PASCP 3373
```

```
3641:      WITH BOOLPTR^ DO                                                          PASCP 3374
3642:        BEGIN SIZE := BOOLSIZE; FORM := SCALAR; SCALKIND := DECLARED END;       PASCP 3375
3643:      NEW(NILPTR,POINTER);                                    (*NIL*)           PASCP 3376
3644:      WITH NILPTR^ DO                                                           PASCP 3377
3645:        BEGIN ELTYPE := NIL; SIZE := PTRSIZE; FORM := POINTER END;              PASCP 3378
3646:      NEW(PARMPTR,SCALAR,STANDARD); (*FOR ALIGNMENT OF PARAMETERS*)             P      556
3647:      WITH PARMPTR^ DO                                                          P      557
3648:        BEGIN SIZE := PARMSIZE; FORM := SCALAR; SCALKIND := STANDARD END ;      P      558
3649:      NEW(TEXTPTR,FILES);                                     (*TEXT*)          PASCP 3379
3650:      WITH TEXTPTR^ DO                                                          PASCP 3380
3651:        BEGIN FILTYPE := CHARPTR; SIZE := CHARSIZE; FORM := FILES END           PASCP 3381
3652:    END (*ENTERSTDTYPES*) ;                                                     PASCP 3382
3653:                                                                                PASCP 3383
3654:    PROCEDURE ENTSTDNAMES;                                                      PASCP 3384
3655:      VAR CP,CP1: CTP; I: INTEGER;                                              PASCP 3385
3656:    BEGIN                                                     (*NAME:*)         PASCP 3386
3657:                                                              (*******)         PASCP 3387
3658:                                                                                PASCP 3388
3659:      NEW(CP,TYPES);                                          (*INTEGER*)       PASCP 3389
3660:      WITH CP^ DO                                                               PASCP 3390
3661:        BEGIN NAME := 'INTEGER '; IDTYPE := INTPTR; KLASS := TYPES END;         PASCP 3391
3662:      ENTERID(CP);                                                              PASCP 3392
3663:      NEW(CP,TYPES);                                          (*REAL*)          PASCP 3393
3664:      WITH CP^ DO                                                               PASCP 3394
3665:        BEGIN NAME := 'REAL    '; IDTYPE := REALPTR; KLASS := TYPES END;        PASCP 3395
3666:      ENTERID(CP);                                                              PASCP 3396
3667:      NEW(CP,TYPES);                                          (*CHAR*)          PASCP 3397
3668:      WITH CP^ DO                                                               PASCP 3398
3669:        BEGIN NAME := 'CHAR    '; IDTYPE := CHARPTR; KLASS := TYPES END;        PASCP 3399
3670:      ENTERID(CP);                                                              PASCP 3400
3671:      NEW(CP,TYPES);                                          (*BOOLEAN*)       PASCP 3401
3672:      WITH CP^ DO                                                               PASCP 3402
3673:        BEGIN NAME := 'BOOLEAN '; IDTYPE := BOOLPTR; KLASS := TYPES END;        PASCP 3403
3674:      ENTERID(CP);                                                              PASCP 3404
3675:      CP1 := NIL;                                                               PASCP 3405
3676:      FOR I := 1 TO 2 DO                                                        PASCP 3406
3677:        BEGIN NEW(CP,KONST);                                  (*FALSE,TRUE*)    PASCP 3407
3678:          WITH CP^ DO                                                           PASCP 3408
3679:            BEGIN NAME := NA[I]; IDTYPE := BOOLPTR;                             PASCP 3409
3680:              NEXT := CP1; VALUES.IVAL := I - 1; KLASS := KONST                 PASCP 3410
3681:            END;                                                                PASCP 3411
3682:          ENTERID(CP); CP1 := CP                                                PASCP 3412
3683:        END;                                                                    PASCP 3413
3684:      BOOLPTR^.FCONST := CP;                                                    PASCP 3414
3685:      NEW(CP,KONST);                                           (*NIL*)          PASCP 3415
3686:      WITH CP^ DO                                                               PASCP 3416
3687:        BEGIN NAME := 'NIL     '; IDTYPE := NILPTR;                           PASCP 3417
3688:          NEXT := NIL; VALUES.IVAL := 0; KLASS := KONST                         PASCP 3418
3689:        END;                                                                    PASCP 3419
3690:      ENTERID(CP);                                                              PASCP 3420
3691:      FOR I := 3 TO 4 DO                                                        PASCP 3421
3692:        BEGIN NEW(CP,VARS);                                   (*INPUT,OUTPUT*)  PASCP 3422
3693:          WITH CP^ DO                                                           PASCP 3423
3694:            BEGIN NAME := NA[I]; IDTYPE := TEXTPTR; KLASS := VARS;              PASCP 3424
3695:              VKIND := ACTUAL; NEXT := NIL; VLEV := 1;                          PASCP 3425
3696:              VADDR := LCAFTERMARKSTACK+(I-3)*CHARMAX;                          P      559
3697:            END;                                                                PASCP 3427
3698:          ENTERID(CP)                                                           PASCP 3428
3699:        END;                                                                    PASCP 3429
3700:  FOR I:=33 TO 34 DO                                                           PASCP 3430
3701:        BEGIN NEW(CP,VARS);                                   (*PRD,PRR FILES*) PASCP 3431
3702:           WITH CP^ DO                                                          PASCP 3432
3703:             BEGIN NAME := NA[I]; IDTYPE := TEXTPTR; KLASS := VARS;             PASCP 3433
3704:                VKIND := ACTUAL; NEXT := NIL; VLEV := 1;                        PASCP 3434
3705:                VADDR := LCAFTERMARKSTACK+(I-31)*CHARMAX;                       P      560
```

```
3706:            END;                                                                    PASCP 3436
3707:          ENTERID(CP)                                                               PASCP 3437
3708:        END;                                                                        PASCP 3438
3709:      FOR I := 5 TO 16 DO                                                           PASCP 3439
3710:        BEGIN NEW(CP,PROC,STANDARD);                          (*GET,PUT,RESET*)    PASCP 3440
3711:          WITH CP^ DO                                            (*REWRITE,READ*)  PASCP 3441
3712:            BEGIN NAME := NA[I]; IDTYPE := NIL;                  (*WRITE,PACK*)    PASCP 3442
3713:              NEXT := NIL; KEY := I - 4;                         (*UNPACK,PACK*)   PASCP 3443
3714:              KLASS := PROC; PFDECKIND := STANDARD                                  PASCP 3444
3715:            END;                                                                    PASCP 3445
3716:          ENTERID(CP)                                                               PASCP 3446
3717:        END;                                                                        PASCP 3447
3718:      NEW(CP,PROC,STANDARD);                                                        PASCP 3448
3719:      WITH CP^ DO                                                                   PASCP 3449
3720:          BEGIN NAME:=NA[35]; IDTYPE:=NIL;                                        PASCP 3450
3721:                NEXT:= NIL; KEY:=13;                                              PASCP 3451
3722:                KLASS:=PROC; PFDECKIND:= STANDARD                                 PASCP 3452
3723:          END; ENTERID(CP);                                                       PASCP 3453
3724:      FOR I := 17 TO 26 DO                                                          PASCP 3454
3725:        BEGIN NEW(CP,FUNC,STANDARD);                          (*ABS,SQR,TRUNC*)    PASCP 3455
3726:          WITH CP^ DO                                            (*ODD,ORD,CHR*)   PASCP 3456
3727:            BEGIN NAME := NA[I]; IDTYPE := NIL;                 (*PRED,SUCC,EOF*)  PASCP 3457
3728:              NEXT := NIL; KEY := I - 16;                                           PASCP 3458
3729:              KLASS := FUNC; PFDECKIND := STANDARD                                  PASCP 3459
3730:            END;                                                                    PASCP 3460
3731:          ENTERID(CP)                                                               PASCP 3461
3732:        END;                                                                        PASCP 3462
3733:      NEW(CP,VARS);                    (*PARAMETER OF PREDECLARED FUNCTIONS*)      PASCP 3463
3734:      WITH CP^ DO                                                                   PASCP 3464
3735:        BEGIN NAME := '        '; IDTYPE := REALPTR; KLASS := VARS;                 PASCP 3465
3736:          VKIND := ACTUAL; NEXT := NIL; VLEV := 1; VADDR := 0                       PASCP 3466
3737:        END;                                                                        PASCP 3467
3738:      FOR I := 27 TO 32 DO                                                          PASCP 3468
3739:        BEGIN NEW(CP1,FUNC,DECLARED,ACTUAL);                   (*SIN,COS,EXP*)    PASCP 3469
3740:          WITH CP1^ DO                                        (*SQRT,LN,ARCTAN*)    PASCP 3470
3741:            BEGIN NAME := NA[I]; IDTYPE := REALPTR; NEXT := CP;                   PASCP 3471
3742:              FORWDECL := FALSE; EXTERN := TRUE; PFLEV := 0; PFNAME := I - 12;    PASCP 3472
3743:              KLASS := FUNC; PFDECKIND := DECLARED; PFKIND := ACTUAL              PASCP 3473
3744:            END;                                                                    PASCP 3474
3745:          ENTERID(CP1)                                                            PASCP 3475
3746:        END                                                                       PASCP 3476
3747:    END (*ENTSTDNAMES*) ;                                                         PASCP 3477
3748:                                                                                  PASCP 3478
3749:    PROCEDURE ENTERUNDECL;                                                        PASCP 3479
3750:    BEGIN                                                                         PASCP 3480
3751:      NEW(UTYPPTR,TYPES);                                                         PASCP 3481
3752:      WITH UTYPPTR^ DO                                                            PASCP 3482
3753:        BEGIN NAME := '        '; IDTYPE := NIL; KLASS := TYPES END;              PASCP 3483
3754:      NEW(UCSTPTR,KONST);                                                         PASCP 3484
3755:      WITH UCSTPTR^ DO                                                            PASCP 3485
3756:        BEGIN NAME := '        '; IDTYPE := NIL; NEXT := NIL;                     PASCP 3486
3757:          VALUES.IVAL := 0; KLASS := KONST                                        PASCP 3487
3758:        END;                                                                      PASCP 3488
3759:      NEW(UVARPTR,VARS);                                                          PASCP 3489
3760:      WITH UVARPTR^ DO                                                            PASCP 3490
3761:        BEGIN NAME := '        '; IDTYPE := NIL; VKIND := ACTUAL;                 PASCP 3491
3762:          NEXT := NIL; VLEV := 0; VADDR := 0; KLASS := VARS                       PASCP 3492
3763:        END;                                                                      PASCP 3493
3764:      NEW(UFLDPTR,FIELD);                                                         PASCP 3494
3765:      WITH UFLDPTR^ DO                                                            PASCP 3495
3766:        BEGIN NAME := '        '; IDTYPE := NIL; NEXT := NIL; FLDADDR := 0;       PASCP 3496
3767:          KLASS := FIELD                                                          PASCP 3497
3768:        END;                                                                      PASCP 3498
3769:      NEW(UPRCPTR,PROC,DECLARED,ACTUAL);                                          PASCP 3499
3770:      WITH UPRCPTR^ DO                                                            PASCP 3500
```

```
3771:       BEGIN NAME := '        '; IDTYPE := NIL; FORWDECL := FALSE;                PASCP 3501
3772:         NEXT := NIL; EXTERN := FALSE; PFLEV := 0; GENLABEL(PFNAME);              PASCP 3502
3773:         KLASS := PROC; PFDECKIND := DECLARED; PFKIND := ACTUAL                   PASCP 3503
3774:       END;                                                                       PASCP 3504
3775:     NEW(UFCTPTR,FUNC,DECLARED,ACTUAL);                                           PASCP 3505
3776:     WITH UFCTPTR^ DO                                                             PASCP 3506
3777:       BEGIN NAME := '        '; IDTYPE := NIL; NEXT := NIL;                      PASCP 3507
3778:         FORWDECL := FALSE; EXTERN := FALSE; PFLEV := 0; GENLABEL(PFNAME);        PASCP 3508
3779:         KLASS := FUNC; PFDECKIND := DECLARED; PFKIND := ACTUAL                   PASCP 3509
3780:       END                                                                        PASCP 3510
3781:   END (*ENTERUNDECL*) ;                                                          PASCP 3511
3782:                                                                                  PASCP 3512
3783:   PROCEDURE INITSCALARS;                                                         PASCP 3513
3784:   BEGIN FWPTR := NIL;                                                            PASCP 3514
3785:     PRTABLES := FALSE; LIST := TRUE; PRCODE := FALSE;                            PASCP 3515
3786:     DEBUG := TRUE;                                                               P      561
3787:     DP := TRUE; PRTERR := TRUE; ERRINX := 0;                                     PASCP 3516
3788:     INTLABEL := 0; KK := 8; FEXTFILEP := NIL;                                    PASCP 3517
3789:     LC := LCAFTERMARKSTACK+FILEBUFFER*CHARMAX;                                   P      562
3790:     (* NOTE IN THE ABOVE RESERVATION OF BUFFER STORE FOR 2 TEXT FILES *)         PASCP 3519
3791:     IC := 3; EOL := TRUE; LINECOUNT := 0;                                        PASCP 3520
3792:     CH := ' '; CHCNT := 0;                                                       PASCP 3521
3793:     GLOBTESTP := NIL;                                                            PASCP 3522
3794:     MXINT10 := MAXINT DIV 10; DIGMAX := STRGLGTH - 1;                            PASCP 3523
3795:   END (*INITSCALARS*) ;                                                          PASCP 3525
3796:                                                                                  PASCP 3526
3797:   PROCEDURE INITSETS;                                                            PASCP 3527
3798:   BEGIN                                                                          PASCP 3528
3799:     CONSTBEGSYS := [ADDOP,INTCONST,REALCONST,STRINGCONST,IDENT];                 PASCP 3529
3800:     SIMPTYPEBEGSYS := [LPARENT] + CONSTBEGSYS;                                   PASCP 3530
3801:     TYPEBEGSYS:=[ARROW,PACKEDSY,ARRAYSY,RECORDSY,SETSY,FILESY]+SIMPTYPEBEGSYS;   PASCP 3531
3802:     TYPEDELS := [ARRAYSY,RECORDSY,SETSY,FILESY];                                 PASCP 3532
3803:     BLOCKBEGSYS := [LABELSY,CONSTSY,TYPESY,VARSY,PROCSY,FUNCSY,                  PASCP 3533
3804:                     BEGINSY];                                                    PASCP 3534
3805:     SELECTSYS := [ARROW,PERIOD,LBRACK];                                          PASCP 3535
3806:     FACBEGSYS := [INTCONST,REALCONST,STRINGCONST,IDENT,LPARENT,LBRACK,NOTSY];    PASCP 3536
3807:     STATBEGSYS := [BEGINSY,GOTOSY,IFSY,WHILESY,REPEATSY,FORSY,WITHSY,            PASCP 3537
3808:                    CASESY];                                                      PASCP 3538
3809:   END (*INITSETS*) ;                                                             PASCP 3539
3810:                                                                                  PASCP 3540
3811:   PROCEDURE INITTABLES;                                                          PASCP 3541
3812:     PROCEDURE RESWORDS;                                                          PASCP 3542
3813:     BEGIN                                                                        PASCP 3543
3814:       RW[ 1] := 'IF      '; RW[ 2] := 'DO      '; RW[ 3] := 'OF      ';         PASCP 3544
3815:       RW[ 4] := 'TO      '; RW[ 5] := 'IN      '; RW[ 6] := 'OR      ';         PASCP 3545
3816:       RW[ 7] := 'END     '; RW[ 8] := 'FOR     '; RW[ 9] := 'VAR     ';         PASCP 3546
3817:       RW[10] := 'DIV     '; RW[11] := 'MOD     '; RW[12] := 'SET     ';         PASCP 3547
3818:       RW[13] := 'AND     '; RW[14] := 'NOT     '; RW[15] := 'THEN    ';         PASCP 3548
3819:       RW[16] := 'ELSE    '; RW[17] := 'WITH    '; RW[18] := 'GOTO    ';         PASCP 3549
3820:       RW[19] := 'CASE    '; RW[20] := 'TYPE    ';                               PASCP 3550
3821:       RW[21] := 'FILE    '; RW[22] := 'BEGIN   ';                               PASCP 3551
3822:       RW[23] := 'UNTIL   '; RW[24] := 'WHILE   '; RW[25] := 'ARRAY   ';         PASCP 3552
3823:       RW[26] := 'CONST   '; RW[27] := 'LABEL   ';                               PASCP 3553
3824:       RW[28] := 'REPEAT  '; RW[29] := 'RECORD  '; RW[30] := 'DOWNTO  ';         PASCP 3554
3825:       RW[31] := 'PACKED  '; RW[32] := 'FORWARD '; RW[33] := 'PROGRAM ';         PASCP 3555
3826:       RW[34] := 'FUNCTION'; RW[35] := 'PROCEDUR';                               PASCP 3556
3827:       FRW[1] :=  1; FRW[2] :=  1; FRW[3] :=  7; FRW[4] := 15; FRW[5] := 22;     PASCP 3557
3828:       FRW[6] := 28; FRW[7] := 32; FRW[8] := 34; FRW[9] := 36;                   PASCP 3558
3829:     END (*RESWORDS*) ;                                                           PASCP 3559
3830:                                                                                  PASCP 3560
3831:     PROCEDURE SYMBOLS;                                                           PASCP 3561
3832:     BEGIN                                                                        PASCP 3562
3833:       RSY[1] := IFSY; RSY[2] := DOSY; RSY[3] := OFSY; RSY[4] := TOSY;            PASCP 3563
3834:       RSY[5] := RELOP; RSY[6] := ADDOP; RSY[7] := ENDSY; RSY[8] := FORSY;        PASCP 3564
3835:       RSY[9] := VARSY; RSY[10] := MULOP; RSY[11] := MULOP; RSY[12] := SETSY;     PASCP 3565
```

```
3836:        RSY[13] := MULOP; RSY[14] := NOTSY; RSY[15] := THENSY;                    PASCP 3566
3837:        RSY[16] := ELSESY; RSY[17] := WITHSY; RSY[18] := GOTOSY;                  PASCP 3567
3838:        RSY[19] := CASESY; RSY[20] := TYPESY;                                     PASCP 3568
3839:        RSY[21] := FILESY; RSY[22] := BEGINSY;                                    PASCP 3569
3840:        RSY[23] := UNTILSY; RSY[24] := WHILESY; RSY[25] := ARRAYSY;               PASCP 3570
3841:        RSY[26] := CONSTSY; RSY[27] := LABELSY;                                   PASCP 3571
3842:        RSY[28] := REPEATSY; RSY[29] := RECORDSY; RSY[30] := DOWNTOSY;            PASCP 3572
3843:        RSY[31] := PACKEDSY; RSY[32] := FORWARDSY; RSY[33] := PROGSY;             PASCP 3573
3844:        RSY[34] := FUNCSY; RSY[35] := PROCSY;                                     PASCP 3574
3845:        SSY['+'] := ADDOP; SSY['-'] := ADDOP; SSY['*'] := MULOP;                  PASCP 3575
3846:        SSY['/'] := MULOP; SSY['('] := LPARENT; SSY[')'] := RPARENT;              PASCP 3576
3847:        SSY['$'] := OTHERSY; SSY['='] := RELOP; SSY[' '] := OTHERSY;              PASCP 3577
3848:        SSY[','] := COMMA; SSY['.'] := PERIOD; SSY[''''] := OTHERSY;              PASCP 3578
3849:        SSY['['] := LBRACK; SSY[']'] := RBRACK; SSY[':'] := COLON;                PASCP 3579
3850:        SSY['^'] := ARROW;                                                        PASCP 3580
3851:        SSY['<'] := RELOP; SSY['>'] := RELOP;                                     PASCP 3581
3852:        SSY[';'] := SEMICOLON;                                                    PASCP 3582
3853:      END (*SYMBOLS*) ;                                                           PASCP 3583
3854:                                                                                  PASCP 3584
3855:      PROCEDURE RATORS;                                                           PASCP 3585
3856:        VAR I: INTEGER; CH: CHAR;                                                 PASCP 3586
3857:      BEGIN                                                                       PASCP 3587
3858:        FOR I := 1 TO 35 (*NR OF RES WORDS*) DO ROP[I] := NOOP;                   PASCP 3588
3859:        ROP[5] := INOP; ROP[10] := IDIV; ROP[11] := IMOD;                         PASCP 3589
3860:        ROP[6] := OROP; ROP[13] := ANDOP;                                         PASCP 3590
3861:      FOR CH := CHR(ORDMINCHAR) TO CHR(ORDMAXCHAR) DO SOP[CH] := NOOP;            BOOT     4
3862:        SOP['+'] := PLUS; SOP['-'] := MINUS; SOP['*'] := MUL; SOP['/'] := RDIV;  PASCP 3592
3863:        SOP['='] := EQOP;                                                         PASCP 3593
3864:        SOP['<'] := LTOP; SOP['>'] := GTOP;                                       PASCP 3594
3865:      END (*RATORS*) ;                                                            PASCP 3595
3866:                                                                                  PASCP 3596
3867:      PROCEDURE PROCMNEMONICS;                                                    PASCP 3597
3868:      BEGIN                                                                       PASCP 3598
3869:        SNA[ 1] :=' GET'; SNA[ 2] :=' PUT'; SNA[ 3] :=' RDI'; SNA[ 4] :=' RDR';  PASCP 3599
3870:        SNA[ 5] :=' RDC'; SNA[ 6] :=' WRI'; SNA[ 7] :=' WRO'; SNA[ 8] :=' WRR';  PASCP 3600
3871:        SNA[ 9] :=' WRC'; SNA[10] :=' WRS'; SNA[11] :=' PAK'; SNA[12] :=' NEW';  PASCP 3601
3872:        SNA[13] :=' RST'; SNA[14] :=' ELN'; SNA[15] :=' SIN'; SNA[16] :=' COS';  PASCP 3602
3873:        SNA[17] :=' EXP'; SNA[18] :=' SQT'; SNA[19] :=' LOG'; SNA[20] :=' ATN';  PASCP 3603
3874:        SNA[21] :=' RLN'; SNA[22] :=' WLN'; SNA[23] :=' SAV';                    PASCP 3604
3875:      END (*PROCMNEMONICS*) ;                                                     PASCP 3605
3876:                                                                                  PASCP 3606
3877:      PROCEDURE INSTRMNEMONICS;                                                   PASCP 3607
3878:      BEGIN                                                                       PASCP 3608
3879:        MN[0] :=' ABI'; MN[1] :=' ABR'; MN[2] :=' ADI'; MN[3] :=' ADR';           PASCP 3609
3880:        MN[4] :=' AND'; MN[5] :=' DIF'; MN[6] :=' DVI'; MN[7] :=' DVR';           PASCP 3610
3881:        MN[8] :=' EOF'; MN[9] :=' FLO'; MN[10] :=' FLT'; MN[11] :=' INN';         PASCP 3611
3882:        MN[12] :=' INT'; MN[13] :=' IOR'; MN[14] :=' MOD'; MN[15] :=' MPI';       PASCP 3612
3883:        MN[16] :=' MPR'; MN[17] :=' NGI'; MN[18] :=' NGR'; MN[19] :=' NOT';       PASCP 3613
3884:        MN[20] :=' ODD'; MN[21] :=' SBI'; MN[22] :=' SBR'; MN[23] :=' SGS';       PASCP 3614
3885:        MN[24] :=' SQI'; MN[25] :=' SQR'; MN[26] :=' STO'; MN[27] :=' TRC';       PASCP 3615
3886:        MN[28] :=' UNI'; MN[29] :=' STP'; MN[30] :=' CSP'; MN[31] :=' DEC';       PASCP 3616
3887:        MN[32] :=' ENT'; MN[33] :=' FJP'; MN[34] :=' INC'; MN[35] :=' IND';       PASCP 3617
3888:        MN[36] :=' IXA'; MN[37] :=' LAO'; MN[38] :=' LCA'; MN[39] :=' LDO';       PASCP 3618
3889:        MN[40] :=' MOV'; MN[41] :=' MST'; MN[42] :=' RET'; MN[43] :=' SRO';       PASCP 3619
3890:        MN[44] :=' XJP'; MN[45] :=' CHK'; MN[46] :=' CUP'; MN[47] :=' EQU';       PASCP 3620
3891:        MN[48] :=' GEQ'; MN[49] :=' GRT'; MN[50] :=' LDA'; MN[51] :=' LDC';       PASCP 3621
3892:        MN[52] :=' LEQ'; MN[53] :=' LES'; MN[54] :=' LOD'; MN[55] :=' NEQ';       PASCP 3622
3893:        MN[56] :=' STR'; MN[57] :=' UJP'; MN[58] :=' ORD'; MN[59] :=' CHR';       P      563
3894:     MN[60] :=' UJC';                                                             P      564
3895:      END (*INSTRMNEMONICS*) ;                                                    PASCP 3624
3896:                                                                                  P      565
3897:                                                                                  P      566
3898:       PROCEDURE CHARTYPES;                                                       P      567
3899:       VAR I : INTEGER;                                                           P      568
3900:       BEGIN                                                                      P      569
```

```
3901:         FOR I := ORDMINCHAR TO ORDMAXCHAR DO CHARTP[CHR(I)] := ILLEGAL;          P      570
3902:         CHARTP['A'] := LETTER ;                                                  P      571
3903:         CHARTP['B'] := LETTER ; CHARTP['C'] := LETTER ;                          P      572
3904:         CHARTP['D'] := LETTER ; CHARTP['E'] := LETTER ;                          P      573
3905:         CHARTP['F'] := LETTER ; CHARTP['G'] := LETTER ;                          P      574
3906:         CHARTP['H'] := LETTER ; CHARTP['I'] := LETTER ;                          P      575
3907:         CHARTP['J'] := LETTER ; CHARTP['K'] := LETTER ;                          P      576
3908:         CHARTP['L'] := LETTER ; CHARTP['M'] := LETTER ;                          P      577
3909:         CHARTP['N'] := LETTER ; CHARTP['O'] := LETTER ;                          P      578
3910:         CHARTP['P'] := LETTER ; CHARTP['Q'] := LETTER ;                          P      579
3911:         CHARTP['R'] := LETTER ; CHARTP['S'] := LETTER ;                          P      580
3912:         CHARTP['T'] := LETTER ; CHARTP['U'] := LETTER ;                          P      581
3913:         CHARTP['V'] := LETTER ; CHARTP['W'] := LETTER ;                          P      582
3914:         CHARTP['X'] := LETTER ; CHARTP['Y'] := LETTER ;                          P      583
3915:         CHARTP['Z'] := LETTER ; CHARTP['0'] := NUMBER ;                          P      584
3916:         CHARTP['1'] := NUMBER ; CHARTP['2'] := NUMBER ;                          P      585
3917:         CHARTP['3'] := NUMBER ; CHARTP['4'] := NUMBER ;                          P      586
3918:         CHARTP['5'] := NUMBER ; CHARTP['6'] := NUMBER ;                          P      587
3919:         CHARTP['7'] := NUMBER ; CHARTP['8'] := NUMBER ;                          P      588
3920:         CHARTP['9'] := NUMBER ; CHARTP['+'] := SPECIAL;                          P      589
3921:         CHARTP['-'] := SPECIAL; CHARTP['*'] := SPECIAL;                          P      590
3922:         CHARTP['/'] := SPECIAL; CHARTP['('] := SPECIAL;                          P      591
3923:         CHARTP[')'] := SPECIAL; CHARTP['$'] := SPECIAL;                          P      592
3924:         CHARTP['='] := SPECIAL; CHARTP[' '] := SPECIAL;                          P      593
3925:         CHARTP[','] := SPECIAL; CHARTP['.'] := SPECIAL;                          P      594
3926:         CHARTP[''''] := SPECIAL; CHARTP['['] := SPECIAL;                         P      595
3927:         CHARTP[']'] := SPECIAL; CHARTP[':'] := SPECIAL;                          P      596
3928:         CHARTP['^'] := SPECIAL; CHARTP[';'] := SPECIAL;                          P      597
3929:         CHARTP['<'] := SPECIAL; CHARTP['>'] := SPECIAL;                          P      598
3930:         ORDINT['0'] := 0; ORDINT['1'] := 1; ORDINT['2'] := 2;                    CH       4
3931:         ORDINT['3'] := 3;                                                        CH       5
3932:         ORDINT['4'] := 4; ORDINT['5'] := 5; ORDINT['6'] := 6;                    CH       6
3933:       ORDINT['7'] := 7; ORDINT['8'] := 8; ORDINT['9'] := 9;                      CH       7
3934:      END;                                                                        P      599
3935:                                                                                  PASCP 3625
3936:      PROCEDURE INITDX;                                                           P      600
3937:      BEGIN                                                                       P      601
3938:        CDX[ 0] :=  0; CDX[ 1] :=  0; CDX[ 2] := -1; CDX[ 3] := -1;               P      602
3939:        CDX[ 4] := -1; CDX[ 5] := -1; CDX[ 6] := -1; CDX[ 7] := -1;               P      603
3940:        CDX[ 8] :=  0; CDX[ 9] :=  0; CDX[10] :=  0; CDX[11] := -1;               P      604
3941:        CDX[12] := -1; CDX[13] := -1; CDX[14] := -1; CDX[15] := -1;               P      605
3942:        CDX[16] := -1; CDX[17] :=  0; CDX[18] :=  0; CDX[19] :=  0;               P      606
3943:        CDX[20] :=  0; CDX[21] := -1; CDX[22] := -1; CDX[23] :=  0;               P      607
3944:        CDX[24] :=  0; CDX[25] :=  0; CDX[26] := -2; CDX[27] :=  0;               P      608
3945:        CDX[28] := -1; CDX[29] :=  0; CDX[30] :=  0; CDX[31] :=  0;               P      609
3946:        CDX[32] :=  0; CDX[33] := -1; CDX[34] :=  0; CDX[35] :=  0;               P      610
3947:        CDX[36] := -1; CDX[37] := +1; CDX[38] := +1; CDX[39] := +1;               P      611
3948:        CDX[40] := -2; CDX[41] :=  0; CDX[42] :=  0; CDX[43] := -1;               P      612
3949:        CDX[44] := -1; CDX[45] :=  0; CDX[46] :=  0; CDX[47] := -1;               P      613
3950:        CDX[48] := -1; CDX[49] := -1; CDX[50] := +1; CDX[51] := +1;               P      614
3951:        CDX[52] := -1; CDX[53] := -1; CDX[54] := +1; CDX[55] := -1;               P      615
3952:        CDX[56] := -1; CDX[57] :=  0; CDX[58] :=  0; CDX[59] :=  0;               P      616
3953:        CDX[60] := 0;                                                             P      617
3954:        PDX[ 1] := -1; PDX[ 2] := -1; PDX[ 3] := -2; PDX[ 4] := -2;               P      618
3955:        PDX[ 5] := -2; PDX[ 6] := -3; PDX[ 7] := -3; PDX[ 8] := -3;               P      619
3956:        PDX[ 9] := -3; PDX[10] := -4; PDX[11] :=  0; PDX[12] := -2;               P      620
3957:        PDX[13] := -1; PDX[14] :=  0; PDX[15] :=  0; PDX[16] :=  0;               P      621
3958:        PDX[17] :=  0; PDX[18] :=  0; PDX[19] :=  0; PDX[20] :=  0;               P      622
3959:        PDX[21] := -1; PDX[22] := -1; PDX[23] := -1;                              P      623
3960:      END;                                                                        P      624
3961:                                                                                  P      625
3962:    BEGIN (*INITTABLES*)                                                          PASCP 3626
3963:      RESWORDS; SYMBOLS; RATORS;                                                  PASCP 3627
3964:      INSTRMNEMONICS; PROCMNEMONICS;                                              PASCP 3628
3965:      CHARTYPES; INITDX;                                                          P      626
```

```
3966:    END (*INITTABLES*) ;                                                    PASCP 3629
3967:                                                                            PASCP 3630
3968:  BEGIN                                                                     PASCP 3631
3969:    (*INITIALIZE*)                                                          PASCP 3632
3970:    (************)                                                          PASCP 3633
3971:    INITSCALARS; INITSETS; INITTABLES;                                      PASCP 3634
3972:                                                                            PASCP 3635
3973:                                                                            PASCP 3636
3974:    (*ENTER STANDARD NAMES AND STANDARD TYPES:*)                            PASCP 3637
3975:    (****************************************)                              PASCP 3638
3976:                                                                            PASCP 3639
3977:    LEVEL := 0; TOP := 0;                                                   PASCP 3640
3978:    WITH DISPLAY[0] DO                                                      PASCP 3641
3979:      BEGIN FNAME := NIL; FLABEL := NIL; OCCUR := BLCK END;                 PASCP 3642
3980:    ENTERSTDTYPES;   STDNAMES; ENTSTDNAMES;   ENTERUNDECL;                  PASCP 3643
3981:    TOP := 1; LEVEL := 1;                                                   PASCP 3644
3982:    WITH DISPLAY[1] DO                                                      PASCP 3645
3983:      BEGIN FNAME := NIL; FLABEL := NIL; OCCUR := BLCK END;                 PASCP 3646
3984:                                                                            PASCP 3647
3985:                                                                            PASCP 3648
3986:    (*COMPILE:*)                                                            PASCP 3649
3987:    (**********)                                                            PASCP 3650
3988:                                                                            PASCP 3651
3989:    INSYMBOL;                                                               PASCP 3652
3990:    PROGRAMME(BLOCKBEGSYS+STATBEGSYS-[CASESY]);                             PASCP 3653
3991:                                                                            PASCP 3654
3992:  END.                                                                      PASCP 3655
3993:                                                                            PASCP 3656

THAT'S ALL FOLKS!      LINES: 3993   CHARACTERS: 359314
```