



```

MAXEXTFILES=20#,%MAX NUMBER OF EXTERNAL FILES. 00054
MAXFILES=20#,%MAX NUMBER OF FILES DECLARED AT ONE TIME. 00055
MAXPNTRS=50#,%MAX NUMBER OF UNDECLARED POINTERS. 00056
% 00057
%***-NAME TABLES-*** 00058
ARRAY NAMETAB1,NAMETAB2,NAMETAB3(0:MAXTABLES,0:MAXNAMES);% 00059
DEFINE NAMELENGTH=[41:6]#,% 00060
TYPE=[9:10]#,% 00061
IDCLASS=[12:3]#,% 00062
VAR=0#,% 00063
CONST=1#,% 00064
FUNC=2#,% 00065
PROC=3#,% 00066
TYPES=4#,% 00067
INFO=[23:11]#,% 00068
FORMAL=[24:1]#,% 00069
FORWARDDEF=[25:1]#,% 00070
EXTERNALFILE=[26:1]#;% 00071
% 00072
%***-DISPLAY VECTOR-*** 00073
ARRAY DISPLAY(0:MAXLEVEL);% 00074
DEFINE RECTYPE=[9:10]#,% 00075
FIRSTWITHSYM=[19:10]#,% 00076
LASTWITHSYM=[29:10]#,% 00077
NUMPNTRSINWITH=[35:6]#,% 00078
BRACKETSINWITH=[36:1]#,% 00079
NAMETAB=[46:7]#;% 00080
% 00081
%***-TYPE TABLES-*** 00082
ARRAY TYPETAB1,TYPETAB2,TYPETAB3(0:MAXTYPES);% 00083
DEFINE FORM=[3:4]#,% 00084
NUMERIC=0#,% 00085
SYMBOLIC=1#,% 00086
SUBTYPE=2#,% 00087
MAINTYPE=[33:10]#,% 00088
CHAR=3#,% 00089
FLOATING=4#,% 00090
ALFA=5#,% 00091
SET=6#,% 00092
SETTYPE=[33:10]#,% 00093
POINTERS=7#,% 00094
POINTTYPE=[33:10]#,% 00095
ARRAYS=8#,% 00096
INXTYPE=[33:10]#,% 00097
ARRTYPE=[43:10]#,% 00098
RECORD=9#,% 00099
RECTAB=[33:10]#,% 00100
FILES=10#,% 00101
FILETYPE=[33:10]#,% 00102
TEXTFILE=11#,% 00103
SIZE=[15:12]#,% 00104
STRUCT=[23:8]#;% 00105
INTEGER-NUMTYPES;% 00106
% 00107
%*** PARAMETER TABLE *** 00108
ARRAY PARAMTAB(0:MAXPARAMS);% 00109
DEFINE PARAMNAME=[9:10]#,% 00110

```

```

PARAMKIND = [13:4]#,%
PARAMLEVEL = [23:10]#,%
PARAMTYPE = [33:10]#,%
PARAMFILE = [34:1]#;%
INTEGER NUMPARAMS;%
%
%*** CONSTANT TABLE ***
ARRAY CONSTTAB[0:MAXCONSTS];%
INTEGER NUMCONSTS;%
%
%*** LABEL TABLE ***
ARRAY LABTAB[0:MAXLABS];%
DEFINE LABVAL = [14:15]#,%
LABDEF = [15:1]#;%
INTEGER NUMLABS, FIRSTLAB;%
%
%*** TABLES FOR I/O AND CHARACTER HANDLING ***
ARRAY CHIO[0], TEXTIO[1], STRINGIO[11];%
POINTER CHARPNT, TEXTPNT, TEXTPNT0, STRINGPNT;%
ARRAY ICARD[0:9], LINE[0:16], XLINE[0:10], ALGOLCARD[0:9];%
POINTER CARDPNT, LINEPNT, XLINEPNT, ALGOLPNT;%
INTEGER CHARCNT, ALGOLCNT, MARGINCNT;%
ARRAY HEADTEXT[0:10], ERRLINE[0:16];%
INTEGER LINECNT, PAGECNT, ERRINX;%
%
%*** XREF FILE AND TABLE ***
FILE XREFFILE DISK SERIAL [20:3000] (2,3,150);%
ARRAY BLOCKTAB[0:MAXTABLES], XREFLINE[0:16];%
INTEGER NUMXREF, NUMBLOCKS; POINTER XREFPNT;%
%
%*** OTHER TABLES ***
INTEGER ARRAY VARLIST[0:LISTLENGTH]; % TEMPORARY LIST OF VARIABLES.
INTEGER VARINDEX, FIRSTVAR;%
ARRAY SYMTAB[0:MAXSYMS]; % USED BY "EXPRESSION".
INTEGER NUMSYMS;%
ARRAY WITHTAB[0:MAXWITHSYMS]; % USED BY "WITHSTAT".
INTEGER NWITHSYMS;%
INTEGER ARRAY SYMBOL[0:64]; % USED BY "INSYMBOL".
INTEGER ARRAY SYMKIND[0:61]; % USED IN ERROR RECOVERY.
ARRAY PNTRTAB1, PNTRTAB2, PNTRTAB3[0:MAXPNTRS]; % USED FOR FORWARD POINTERS
INTEGER NUMPNTRS;%
ARRAY EXTFILETAB[0:MAXEXTFILES]; % EXTERNAL FILES.
INTEGER NUMEXTFILES;%
ARRAY FILETAB[0:MAXFILES]; % FILES IN USE.
INTEGER NUMFILES;%
BOOLEAN ARRAY ERR[0:119]; % RECORDS ERROR MESSAGES.
%
%*** COMPILER TIME OPTIONS ***
BOOLEAN LISTOPTION, RESWORDOPTION, CHECKOPTION, DUMPOPTION, XREFOPTION;%
INTEGER CARDLLENGTH;%
%
%*** INTRINSIC TYPES ***
INTEGER INTTYPE, REALTYPE, ALFATYPE, CHARTYPE, BOOLOPTYPE, NILTYPE, TEXTTYPE,%
INPUTFILE, OUTPUTFILE, EMPTYSET;%
BOOLEAN INPUTDECL, OUTPUTDECL;%
%
%*** TEMPORARY VARIABLES ***

```

```

00111
00112
00113
00114
00115
00116
00117
00118
00119
00120
00121
00122
00123
00124
00125
00126
00127
00128
00129
00130
00131
00132
00133
00134
00135
00136
00137
00138
00139
00140
00141
00142
00143
00144
00145
00146
00147
00148
00149
00150
00151
00152
00153
00154
00155
00156
00157
00158
00159
00160
00161
00162
00163
00164
00165
00166
00167

```

INTEGER T1,T2,T3,T4,T5;%

\*\*\* OTHER VARIABLES \*\*\*

ALPHA USER; % THE USER NUMBER FOUND ON THE USER CARD.

INTEGER CURLEVEL, % CURRENT PROCEDURE LEVEL.
TOPLEVEL, % TOP LEVEL IN DISPLAY VECTOR.
NUMBEGINS, % NUMBER OF "BEGIN" S IN THE PROGRAM.
NUMCASES, % NUMBER OF CASE-STATEMENTS IN PROGRAM.
NUMREPS, % NUMBER OF REPEAT-STATEMENTS IN PROGRAM.
NUMTEMPS, % NUMBER OF TEMPORARY VARIABLES IN USE.
CURFUNC, % INDEX OF FUNCTION CURRENTLY COMPILED.
CURSY, % LAST SYMBOL READ BY SCANNER.
CURTYPE, % TYPE OF ENTITY LAST COMPILED.
CURMODE, % CURRENT EXPRESSION MODE.
LASTREC; % LAST RECORD TABLE DEFINED.

LABEL ENDOFINPUT;%

FORMAT-NOERRORS ("NO-ERRORS DETECTED."),%
ERRORS (IS," ERRORS DETECTED"/),%
ALIST (" \$ SET LIST SINGLE"),%
NOALIST (" \$ RESET LIST"),%
LASTLINE (" ; TERMINATE: END OF PASCAL PROGRAM."),%
TERMMESS ("\*\*\* END-OF-INPUT. COMPILATION TERMINATED.");%

MONITOR-EXPOVR:=REALOVERFLOW;%

\*\*\* SCANNER SYMBOLS \*\*\*

DEFINE IDENTIFIER=1#, INTCONST=2#, REALCONST=3#, ALFACONST=4#,%
CHARCONST=5#, NOTSY=6#, ASTERISK=7#, SLASH=8#,%
ANDSY=9#, DIVSY=10#, MUDSY=11#, PLUS=12#,%
MINUS=13#, ORSY=14#, LSSSY=15#, LEQSY=16#,%
GEQSY=17#, GTRSY=18#, NEQSY=19#, EQLSY=20#,%
INSY=21#, LPAR=22#, RPAR=23#, LBRACKET=24#,%
RBRACKET=25#, DOUBLEDOT=26#, CUMMA=27#, SEMICOLON=28#,%
DOT=29#, ARROW=30#, COLON=31#, ASSIGNSY=32#,%
BEGINSY=33#, ENDSY=34#, IFSY=35#, THENSY=36#,%
ELSESY=37#, CASESY=38#, OFSY=39#, REPEATSY=40#,%
UNTILSY=41#, WHILESY=42#, DOSY=43#, FORSY=44#,%
TOSY=45#, DOWNTOSY=46#, GOTOSY=47#, NILSY=48#,%
TYPESY=49#, ARRAYSY=50#, RECORDSY=51#, FILESY=52#,%
SETSY=53#, CONSTSY=54#, VARSY=55#, LABELSY=56#,%
FUNCSY=57#, PROCYSY=58#, WITHSY=59#, PROGRAMSY=60#,%
PACKEDSY=61#;%

DEFINE INITIAL=0#, MIDDLE=1#, TERMINAL=2#;%

DEFINE NUMBER=0#,... BITPATTERN=1#;%

\$ PAGE%

%%%%%%%%%

PART 2: ... COMPILER UTILITY ROUTINES.

00168
00169
00170
00171
00172
00173
00174
00175
00176
00177
00178
00179
00180
00181
00182
00183
00184
00185
00186
00187
00188
00189
00190
00191
00192
00193
00194
00195
00196
00197
00198
00199
00200
00201
00202
00203
00204
00205
00206
00207
00208
00209
00210
00211
00212
00213
00214
00215
00216
00217
00218
00219
00220
00221
00222
00223
00224

14127
Moore Business Forms, Inc. sv

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% 00225
% 00226
% 00227
PROCEDURE INSYMBOL; FORWARD;% 00228
PROCEDURE WRITEALGOL; FORWARD;% 00229
PROCEDURE NEWXREF(NAME1,NAME2,TABLE,DECL);% 00230
VALUE NAME1, NAME2, TABLE, DECL;% 00231
REAL NAME1, NAME2;% 00232
INTEGER TABLE;% 00233
BOOLEAN DECL;% 00234
FORWARD;% 00235
% 00236
DEFINE NDIGITS(N)=% 00237
IF N< 9 THEN 1 ELSE% 00238
IF N<99 THEN 2 ELSE 3 DIGITS#;% 00239
% 00240
DEFINE HEADING=% 00241
BEGIN COMMENT *** PRINTS A HEADING ON TOP OF A NEW PAGE. ;% 00242
PAGECNT:=PAGECNT+1;% 00243
REPLACE POINTER(HEADTEXT[*])+85 BY PAGECNT FOR NDIGITS(PAGECNT);% 00244
WRITE(LINES[PAGE]);% 00245
WRITE(LINES[DBL],11,HEADTEXT[*]);% 00246
LINECNT:=2;% 00247
END OF HEADING#;% 00248
% 00249
% 00250
DEFINE PRINTLINE= %*** PRINTS A SOURCE CODE LINE. 00251
BEGIN% 00252
REPLACE LINEPNT-8 BY CARDCNT FOR 5 DIGITS;% 00253
IF LINECNT>=LINESPERPAGE THEN HEADING;% 00254
IF RESWORDOPTION THEN% 00255
BEGIN% 00256
WRITE(LINES[NO],11,XLINE[*]);% 00257
WRITE(LINES[NO],11,XLINE[*]);% 00258
END;% 00259
WRITE(LINES,17,LINE[*]);% 00260
LINECNT:=LINECNT+1;% 00261
END OF PRINTLINE#;% 00262
% 00263
% 00264
DEFINE NEWCARD= %*** READS A NEW SOURCE CODE CARD. 00265
BEGIN% 00266
IF LISTOPTION THEN PRINTLINE;% 00267
IF ERRINX>0 THEN PRINTERRORS;% 00268
READ(CARD,10,ICARD[*]) [ENDOFINPUT];% 00269
CARDPNT:=POINTER(ICARD[*]);% 00270
REPLACE LINEPNT BY CARDPNT FOR 10 WORDS, " " FOR 6 WORDS;% 00271
REPLACE XLINEPNT BY " " FOR 10 WORDS;% 00272
CHARCNT:=CARDLENGTH;% 00273
MARGINCNT:=85;% 00274
CARDCNT:=CARDCNT+1;% 00275
END#;% 00276
% 00277
% 00278
DEFINE GEN(T,N,START)= %*** GENERATE A TEXT "T", CONSISTING OF 00279
BEGIN %*** "N" LETTERS, STARTING AT "START". 00280
IF ALGOLCNT<N THEN WRITEALGOL;% 00281

```

```

TEXT[0]:=T;%
REPLACE ALGOLPNT:ALGOLPNT BY TEXTPNT0+START FOR N;%
ALGOLCNT:=ALGOLCNT+N;%
END#;%
%
%
DEFINE GENID(L,NUM,NDIG)= %*** GENERATE AN XALGOL IDENTIFIER.
BEGIN%
IF ALGOLCNT<=NDIG THEN WRITEALGOL;%
CH[0]:=L;%
REPLACE ALGOLPNT:ALGOLPNT BY CHARPNT FOR 1, NUM FOR NDIG DIGITS;%
ALGOLCNT:=ALGOLCNT-NDIG-1;%
END#;%
%
%
INTEGER NABS,NSIZE;%
%
DEFINE GENINT(N)=%
BEGIN%
NABS:=ABS(N); IF N<0 THEN GEN("",-1,7);%
NSIZE:=IF NABS< 9 THEN 1 ELSE%
IF NABS< 99 THEN 3 ELSE%
IF NABS< 999 THEN 5 ELSE%
IF NABS<99999 THEN 8 ELSE 12;%
IF ALGOLCNT<NSIZE THEN WRITEALGOL;%
IF NSIZE=12 THEN%
REPLACE ALGOLPNT:ALGOLPNT BY (NABS DIV 1000000) FOR 6 DIGITS,%
ENTIER(NABS MOD 1000000) FOR 6 DIGITS ELSE%
REPLACE ALGOLPNT:ALGOLPNT BY NABS FOR NSIZE DIGITS;%
ALGOLCNT:=ALGOLCNT-NSIZE;%
END OF GENINT#;%
%
%
PROCEDURE GENREAL(X);%
VALUE X; REAL X;%
BEGIN%
REAL ABSX;%
INTEGER POWER,V1,V2;%
%
IF X.[46:5]=0 THEN%
BEGIN%
IF ALGOLCNT<9 THEN WRITEALGOL;%
TEXT[0]:=X;%
REPLACE ALGOLPNT:ALGOLPNT BY "-", TEXTPNT FOR 7, "";%
ALGOLCNT:=ALGOLCNT-9;%
END ELSE%
BEGIN%
IF ALGOLCNT<22 THEN WRITEALGOL;%
IF X<0 THEN GEN("(-",2,6);%
ABSX:=ABS(X);%
IF ABSX>0 THEN%
BEGIN%
WHILE ABSX<=1@7 DO BEGIN ABSX:=ABSX/10; POWER:=POWER+1 END;%
WHILE ABSX<1@6 DO BEGIN ABSX:=ABSX*10; POWER:=POWER-1 END;%
V1:=ENTIER(ABSX);%
V2:=ENTIER((ABSX-V1)*1000000);%
REPLACE ALGOLPNT:ALGOLPNT BY V1 FOR 7 DIGITS, ",",%

```

```

00282
00283
00284
00285
00286
00287
00288
00289
00290
00291
00292
00293
00294
00295
00296
00297
00298
00299
00300
00301
00302
00303
00304
00305
00306
00307
00308
00309
00310
00311
00312
00313
00314
00315
00316
00317
00318
00319
00320
00321
00322
00323
00324
00325
00326
00327
00328
00329
00330
00331
00332
00333
00334
00335
00336
00337
00338

```

```

V2 FOR 6 DIGITS, "@";%
ALGOLCNT:=ALGOLCNT-15;%
IF POWER<0 THEN GEN("-",1,7);%
POWER:=ABS(POWER);%
REPLACE ALGOLPNT:ALGOLPNT-BY-POWER FOR 2 DIGITS;%
ALGOLCNT:=ALGOLCNT-2;%
END ELSE GEN("0",1,7);%
IF X<0 THEN GEN(")",1,7);%
END;%
END OF GENREAL;%
%
%
INTEGER TYPEINDEX;%
%
DEFINE NEWTYPE=%
BEGIN%
IF NUMTYPES>MAXTYPES THEN BEGIN ERROR(45);NUMTYPES:=MAXTYPES-20 END;
TYPEINDEX:=NUMTYPES:=NUMTYPES+1;%
END #;%
%
%
PROCEDURE WRITEALGOL;          %*** WRITES A COMPLETED XALGOL CARD TO
BEGIN                          %*** THE FILE.
REPLACE POINTER(ALGOLCARD[9]) BY CARDNT FOR 8 DIGITS;%
WRITE(PASCALGOL,10,ALGOLCARD[*]);%
IF DUMPOPTION THEN WRITE(LINES,10,ALGOLCARD[*]);%
ALGOLPNT:=POINTER(ALGOLCARD[*]); ALGOLCNT:=71;%
REPLACE ALGOLPNT BY " " FOR 9 WORDS;%
END OF WRITEALGOL;%
%
%
DEFINE MARGIN(LETTER,NUM)=%
BEGIN COMMENT *** PLACES INFORMATION IN THE MARGIN. ;%
IF MARGINCNT<=118 THEN%
BEGIN TEXT[0]:=LETTER;%
REPLACE LINEPNT+MARGINCNT BY TEXTPNT+5 FOR 2,%
NUM FOR NDIGITS(NUM);%
MARGINCNT:=MARGINCNT+6;%
END;%
END OF MARGIN#;%
%
%
PROCEDURE SKIP(SYMBOL);        %*** SKIP SYMBOLS TO RECOVER FROM ERROR
VALUE SYMBOL; INTEGER SYMBOL; %*** CONDITION.
BEGIN%
WHILE CURSY<SYMBOL AND SYMKIND[CURSY]=MIDDLE DO%
IF CURSY=RECORDSY THEN%
BEGIN DO BEGIN INSYMBOL;%
SKIP(99);%
END UNTIL CURSY<SEMICOLON AND CURSY<CASESY;%
END ELSE INSYMBOL;%
END OF SKIP;%
%
%
PROCEDURE ERROR(ERRNUM);%
VALUE ERRNUM; INTEGER ERRNUM;%
BEGIN COMMENT *** ARRANGE ERROR INDICATOR. ;%

```

```

00339
00340
00341
00342
00343
00344
00345
00346
00347
00348
00349
00350
00351
00352
00353
00354
00355
00356
00357
00358
00359
00360
00361
00362
00363
00364
00365
00366
00367
00368
00369
00370
00371
00372
00373
00374
00375
00376
00377
00378
00379
00380
00381
00382
00383
00384
00385
00386
00387
00388
00389
00390
00391
00392
00393
00394
00395

```

```

NUMERRS:=NUMERRS+1;%
ERR[ERRNUM]:=TRUE;%
ERRINX:=MAX(ERRINX,CARDLENGTH-2-CHARCNT);%
IF ERRINX<=115 THEN%
BEGIN REPLACE POINTER(ERRLINE[1])+ERRINX BY "X",%
ERRNUM FOR NDIGITS(ERRNUM);%
ERRINX:=ERRINX+(IF ERRNUM<= 9 THEN 2 ELSE%
IF ERRNUM<=99 THEN 3 ELSE 4);%
END END OF ERROR;%
%
%
PROCEDURE PRINTERRORS;%
BEGIN COMMENT *** PRINT ERROR INDICATORS. ;%
IF NOT LISTOPTION THEN PRINTLINE;%
WRITE(LINES,17,ERRLINE[*]);%
LINECNT:=LINECNT+1;%
REPLACE POINTER(ERRLINE[1]) BY " " FOR 16 WORDS;%
ERRINX:=0;%
END OF PRINTERRORS;%
%
%
DEFINE HASH(N) = (N).[35:36] MOD MAXNAMES#;%
%
INTEGER THISLEVEL,THISTAB,THISINDEX;%
ALPHA THISID,TNAME;%
BOOLEAN FOUND;%
%
DEFINE SEARCHTAB(TAB)= %*** SEARCH NAME TABLE "TAB" FOR THE
BEGIN %*** IDENTIFIER JUST READ,
THISINDEX:=HASH(CURNAME1);%
TNAME:=NAMETAB1[TAB,THISINDEX];%
WHILE (IF TNAME=CURNAME1 THEN NAMETAB2[TAB,THISINDEX]≠CURNAME2%
ELSE TNAME≠0) DO%
BEGIN%
THISINDEX:=IF THISINDEX=0 THEN MAXNAMES ELSE THISINDEX-1;%
TNAME:=NAMETAB1[TAB,THISINDEX];%
END;%
FOUND:=TNAME≠0;%
IF XREFOPTION THEN%
IF FOUND THEN NEWXREF(CURNAME1,CURNAME2,TAB,FALSE); %
END OF SEARCHTAB#;%
%
DEFINE SEARCH= %*** SEARCH ALL TABLES CURRENTLY IN USE.
BEGIN%
THISLEVEL:=TOPLEVEL+1;%
DO BEGIN%
THISLEVEL:=THISLEVEL-1;%
THISTAB:=IF THISLEVEL<=CURLEVEL THEN THISLEVEL%
ELSE DISPLAY[THISLEVEL].NAMETAB;%
SEARCHTAB(THISTAB);%
END UNTIL FOUND OR THISLEVEL=0;%
THISID:=NAMETAB3[THISTAB,THISINDEX];%
END OF SEARCH #;%
%
%
DEFINE NEWNAME(NAME1,NAME2,TAB) =%
BEGIN %*** ENTER A NEW NAME INTO THE NAME TABLE "TAB".

```

```

00396
00397
00398
00399
00400
00401
00402
00403
00404
00405
00406
00407
00408
00409
00410
00411
00412
00413
00414
00415
00416
00417
00418
00419
00420
00421
00422
00423
00424
00425
00426
00427
00428
00429
00430
00431
00432
00433
00434
00435
00436
00437
00438
00439
00440
00441
00442
00443
00444
00445
00446
00447
00448
00449
00450
00451
00452

```



```

THISINDEX:=HASH(NAME1);%
TNAME:=NAMETAB1[TAB,THISINDEX];%
WHILE(IF TNAME=NAME1 THEN NAME1AB2[TAB,THISINDEX]#NAME2%
      ELSE TNAME#0) DO%
BEGIN%
  THISINDEX:=IF THISINDEX=0 THEN MAXNAMES ELSE THISINDEX-1;%
  TNAME:=NAMETAB1[TAB,THISINDEX];%
END;%
IF TNAME#0 THEN ERROR(2);%
NAMETAB1[TAB,THISINDEX]:=NAME1;%
NAMETAB2[TAB,THISINDEX]:=NAME2;%
IF XREFOPTION THEN NEWXREF(NAME1,NAME2,TAB,TRUE);%
END OF NEWNAME #;%
%
%
PROCEDURE INITIALIZE;          %*** INITIALIZATION ***
BEGIN                          %*****
  INTEGER T1,T3;%
  ALPHA A;%
  FILL SYMKIND[*] WITH 28(MIDDLE),TERMINAL,4(MIDDLE),INITIAL,TERMINAL,
  INITIAL,MIDDLE,TERMINAL,INITIAL,MIDDLE,INITIAL,TERMINAL,INITIAL,%
  MIDDLE,INITIAL,2(MIDDLE),INITIAL,MIDDLE,INITIAL,4(MIDDLE),%
  7(INITIAL),MIDDLE;%
%
  FILL SYMBOL[*] WITH 10(0),0,ARROW,0,COLON,GTRSY,GEQSY,PLUS,9(0),%
  DOT,LBRACKET,ANDSY,LPAR,LSSSY,ARROW,0,9(0),0,ASTERISK,MINUS,%
  RPAR,SEMICOLON,LEQSY,0,SLASH,8(0),COMMA,0,NEQSY,EQLSY,RBRACKET,%
  0,DOUBLEDOT;%
%
  LINEPNT :=POINTER(LINE[1]);%
  XLINEPNT:=POINTER(XLINE[1]);%
  REPLACE LINEPNT-8 BY " " => " " FOR 16 WORDS;%
  REPLACE XLINEPNT-8 BY " " FOR 11 WORDS;%
  REPLACE POINTER(ERRLINE[*]) BY "**** " FOR 16 WORDS;%
  ALGOLPNT:=POINTER(ALGOLCARD[*]); ALGOLCNT:=71;%
  REPLACE ALGOLPNT BY " " FOR 9 WORDS;%
  CHARPNT:=POINTER(CH[*])+7;%
  TEXTPNT:=POINTER(TEXT[*])+1; TEXTPNT0:=TEXTPNT-1;%
  REPLACE TEXTPNT BY " " FOR 15;%
  STRINGPNT:=POINTER(STRING[*]);%
  REPLACE POINTER(HEADTEXT[*]) BY " " FOR 10 WORDS, "PAGE ";%
  REPLACE POINTER(HEADTEXT[*]) BY "PASCAL(", EDITION, ")/8-5700";%
  TEXT[0]:=TIME(5);%
  REPLACE POINTER(HEADTEXT[*])+45 BY TEXTPNT+3 FOR 2, "/",%
  TEXTPNT+1 FOR 2, "/", TEXTPNT+5 FOR 2;%
  T1:=TIME(1)/3600;%
  REPLACE POINTER(HEADTEXT[*])+57 BY (T1 DIV 60) FOR 2 DIGITS, ":",%
  ENTIER(T1 MOD 60) FOR 2 DIGITS;%
  HEADING;%
%
%
%*** INITIALIZE INTRINSIC TYPES, CONSTANTS ETC. ***
%
INTIYPE:=T3:=1;          %*** "INTEGER" ***
T1:=NUMERIC; T1.SIZE:=1; T1.STRUCT:=0;%
TYPETAB1[1]:=T1; TYPETAB2[1]:=-MAXINT; TYPETAB3[1]:=MAXINT;%
NEWNAME("7INTEGE","R",0); T3.IDCLASS:=TYPES;%
NAMETAB3[0,THISINDEX]:=T3;%

```

```

00453
00454
00455
00456
00457
00458
00459
00460
00461
00462
00463
00464
00465
00466
00467
00468
00469
00470
00471
00472
00473
00474
00475
00476
00477
00478
00479
00480
00481
00482
00483
00484
00485
00486
00487
00488
00489
00490
00491
00492
00493
00494
00495
00496
00497
00498
00499
00500
00501
00502
00503
00504
00505
00506
00507
00508
00509

```

14127 Moore Business Forms, Inc. 5V

```

REALTYPE:=T3:=2; %*** "REAL" *** 00510
T1.FORM:=FLOATING; TYPETAB1[2]:=T1;% 00511
NEWNAME("400REAL",0,0); T3.IDCLASS:=TYPES;% 00512
NAMETAB3[0,THISINDEX]:=T3;% 00513
ALFA-TYPE:=T3:=3; %*** "ALFA" *** 00514
T1.FORM:=ALFA; TYPETAB1[3]:=T1;% 00515
NEWNAME("400ALFA",0,0); T3.IDCLASS:=TYPES;% 00516
NAMETAB3[0,THISINDEX]:=T3;% 00517
BOOLTYPE:=T3:=4; %*** "BOOLEAN" *** 00518
T1.FORM:=SYMBOLIC; TYPETAB1[4]:=T1; TYPETAB3[4]:=1;% 00519
NEWNAME("7BOOLEAN","N",0); T3.IDCLASS:=TYPES;% 00520
NAMETAB3[0,THISINDEX]:=T3;% 00521
CHARTYPE:=T3:=5; %*** "CHAR" *** 00522
T1.FORM:=CHAR; TYPETAB1[5]:=T1; TYPETAB3[5]:=63;% 00523
NEWNAME("400CHAR",0,0); T3.IDCLASS:=TYPES;% 00524
NAMETAB3[0,THISINDEX]:=T3;% 00525
T3:=BOOLTYPE; T3.IDCLASS:=CONST; %*** "FALSE" *** 00526
NEWNAME("50FALSE",0,0); NAMETAB3[0,THISINDEX]:=T3;% 00527
T3.INFO:=1; %*** "TRUE" *** 00528
NEWNAME("400TRUE",0,0); NAMETAB3[0,THISINDEX]:=T3;% 00529
NUMTYPES:=5;% 00530
NILTYPE:=-1; %*** TYPE OF "NIL" *** 00531
EMPTYSET:=-2; %*** TYPE OF [] *** 00532
NEWNAME("6MAXINT",0,0); T3:=INITYPE; %*** "MAXINT" *** 00533
T3.IDCLASS:=CONST; T3.INFO:=1024;% 00534
NAMETAB3[0,THISINDEX]:=T3;% 00535
NUMCONSTS:=1; CONSTTAB[1]:=MAXINT;% 00536
% 00537
T3:=0; T3.IDCLASS:=PROC; %*** PROCEDURES *** 00538
FOR A:="3000GET", "3000NEW", "400PACK", "400PAGE", "3000PUT",% 00539
"400READ", "6READLN", "50RESET", "6UNPACK", "50WRITE" DO% 00540
BEGIN% 00541
NEWNAME(A,0,0); NAMETAB3[0,THISINDEX]:=T3;% 00542
END;% 00543
NEWNAME("7DISPOS","E",0); NAMETAB3[0,THISINDEX]:=T3;% 00544
NEWNAME("7REWRITE","E",0); NAMETAB3[0,THISINDEX]:=T3;% 00545
NEWNAME("7WRITEL","N",0); NAMETAB3[0,THISINDEX]:=T3;% 00546
% 00547
T3.IDCLASS:=FUNC; %*** FUNCTIONS *** 00548
FOR A:="3000ABS", "6ARCTAN", "3000CHR", "3000COS", "3000EOF",% 00549
"400EOLN", "3000EXP", "20000LN", "30000DD", "400PRED",% 00550
"400SUCC", "50ROUND", "3000SIN", "3000SQR", "400SQRT",% 00551
"50TRUNC", "6CONCAT", "400TIME", "400DATE", "6IOTIME",% 00552
"400USER", "3000ORD"% 00553
DO BEGIN% 00554
NEWNAME(A,0,0); NAMETAB3[0,THISINDEX]:=T3;% 00555
END;% 00556
NEWNAME("7ELAPSE","D",0); NAMETAB3[0,THISINDEX]:=T3;% 00557
NEWNAME("7WEEKDA","Y",0); NAMETAB3[0,THISINDEX]:=T3;% 00558
% 00559
TEXTTYPE:=T3:=NUMTYPES:=NUMTYPES+1; %*** "TEXT" *** 00560
T1:=TEXTFILE; T1.STRUCT:=1; TYPETAB1[TEXTTYPE]:=T1;% 00561
T3.IDCLASS:=TYPES;% 00562
NEWNAME("400TEXT",0,0); NAMETAB3[0,THISINDEX]:=T3;% 00563
T3:=TEXTTYPE; T3.IDCLASS:=VAR; %*** "INPUT" *** 00564
T3.EXTERNALFILE:=1;% 00565
NEWNAME("50INPUT",0,0); INPUTFILE:=THISINDEX;% 00566

```

```

NAMETAB3[0,THISINDEX]:=T3;%
NEWNAME("6OUTPUT",0,0); %*** "OUTPUT" ***
NAMETAB3[0,THISINDEX]:=T3; OUTPUTFILE:=THISINDEX;%
END OF INITIALIZE;%
%
%
%
%*** XREF ROUTINES ***
%*****
%
DEFINE XREFCARD=[16:17]#,%
XREFBLOCK=[26:10]#;%
REAL A0,B0,A1,B1,LASTA0,LASTA1;%
INTEGER NL, LASTBLOCK, A2, AX;%
%
PROCEDURE NEWXREF(NAME1,NAME2,TABLE,DECL);%
VALUE NAME1,NAME2,TABLE,DECL;%
REAL NAME1,NAME2;%
INTEGER TABLE;%
BOOLEAN DECL;%
BEGIN%
NL:=NAME1.NAMELENGTH;%
IF NL<7 THEN NAME1:=0&NAME1[41:41:6]&NAME1[35:6xNL-1:6xNL]%
ELSE NAME2:=0&NAME2[35:6x(NL-6)-1:6x(NL-6)];%
AX:=CARDCNT; AX.XREFBLOCK:=BLOCKTAB[TABLE];%
IF DECL THEN AX:=AX-1000000000000;%
WRITE(XREFFILE,*,NAME1,NAME2,AX);%
END OF NEWXREF;%
%
PROCEDURE XREFMAX(A);%
ARRAY A[0];%
BEGIN%
A[0]:="AZZZZZ"; A[1]:="ZZZZZZ"; A[2]:=9999999999;%
END OF XREFMAX;%
%
%
BOOLEAN PROCEDURE XREFCOMPARE(A,B);%
ARRAY A,B[0];%
BEGIN%
A0:=A[0]; B0:=B[0]; A1:=A[1]; B1:=B[1];%
XREFCOMPARE:=%
IF A0.[35:36]#B0.[35:36] THEN A0.[35:36]<B0.[35:36] ELSE%
IF A1#B1 THEN A1<B1 ELSE%
IF A0#B0 THEN A0.NAMELENGTH<B0.NAMELENGTH ELSE%
A[2] LEQ B[2];%
END OF XREFCOMPARE;%
%
%
PROCEDURE PRINTXREF(FINIS,A);%
VALUE FINIS; BOOLEAN FINIS;%
ARRAY A[0];%
BEGIN%
IF FINIS THEN%
BEGIN%
WRITE(LINES,17,XREFLINE[*]);%
CLOSE(LINES);%
CLOSE(XREFFILE);%

```

```

00567
00568
00569
00570
00571
00572
00573
00574
00575
00576
00577
00578
00579
00580
00581
00582
00583
00584
00585
00586
00587
00588
00589
00590
00591
00592
00593
00594
00595
00596
00597
00598
00599
00600
00601
00602
00603
00604
00605
00606
00607
00608
00609
00610
00611
00612
00613
00614
00615
00616
00617
00618
00619
00620
00621
00622
00623

```

```

END%                                00624
ELSE%                                00625
BEGIN%                                00626
  A0:=A[0];  A1:=A[1];  A2:=A[2];%    00627
  IF A0=LASTA0 AND A1=LASTA1 AND A2.XREFBLOCK=LASTBLUCK THEN% 00628
  BEGIN%                                00629
    IF NUMXREF=15 THEN%                00630
    BEGIN%                                00631
      WRITE(LINES,17,XREFLINE[*]); LINECNT:=LINECNT+1;%    00632
      IF LINECNT>LINESPERPAGE THEN HEADING;%    00633
      XREFPNT:=POINTER(XREFLINE[*]); NUMXREF:=0;%    00634
      REPLACE XREFPNT BY " " FOR 17 WORDS; XREFPNT:=XREFPNT+24;% 00635
    END;%                                00636
    REPLACE XREFPNT BY A2.XREFCARD FOR 5 DIGITS;%    00637
    XREFPNT:=XREFPNT+7; NUMXREF:=NUMXREF+1;%    00638
  END ELSE%                                00639
  IF A2<0 THEN%                        00640
  BEGIN%                                00641
    A2:=A2+1000000000000;%            00642
    WRITE(LINES,17,XREFLINE[*]); LINECNT:=LINECNT+1;%    00643
    IF LINECNT>LINESPERPAGE THEN HEADING;%    00644
    XREFPNT:=POINTER(XREFLINE[*]); NUMXREF:=0;%    00645
    REPLACE XREFPNT BY " " FOR 17 WORDS;%    00646
    TEXT[0]:=A0.[35:36]; LASTA0:=A0;%    00647
    REPLACE XREFPNT BY TEXTPNT+1 FOR A0.NAMELENGTH;%    00648
    TEXT[0]:=LASTA1:=A1;%              00649
    IF A0.NAMELENGTH>6 THEN%           00650
    REPLACE XREFPNT+6 BY TEXTPNT+1 FOR A0.NAMELENGTH-6;%    00651
    REPLACE XREFPNT+17 BY A2.XREFCARD FOR 5 DIGITS;%    00652
    XREFPNT:=XREFPNT+24; LASTBLUCK:=A2.XREFBLOCK;%    00653
  END;%                                00654
END;%                                00655
END OF PRINTXREF;%                    00656
%                                     00657
%                                     00658
%                                     00659
INTEGER TT1,TT2,F1,F2,L1,RT;%         00660
%                                     00661
DEFINE CHECKTYPES(LEFTTYPE,RIGHTTYPE)=% 00662
BEGIN%                                00663
  IF LEFTTYPE>0 AND RIGHTTYPE>0 THEN% 00664
  IF LEFTTYPE≠RIGHTTYPE THEN%         00665
  BEGIN%                                00666
    LT:=LEFTTYPE; RT:=RIGHTTYPE;%     00667
    TT1:=TYPETAB1[LT]; TT2:=TYPETAB1[RT];% 00668
    F1:=TT1.FORM; F2:=TT2.FORM;%      00669
    IF L1≠REALTYPE OR F2≠NUMERIC THEN% 00670
    IF (F1≠SET AND LT≠EMPTYSET)OR(F2≠SET AND RT≠EMPTYSET)THEN% 00671
    IF (F1≠POINTERS AND LT≠NILTYPE)OR(F2≠POINTERS AND RT≠NILTYPE)THEN% 00672
    BEGIN%                                00673
      IF F1=SET AND F2=SET THEN%       00674
      BEGIN%                                00675
        LT:=TT1.SETTYPE; RT:=TT2.SETTYPE;% 00676
        TT1:=TYPETAB1[LT]; TT2:=TYPETAB1[RT];% 00677
        F1:=TT1.FORM; F2:=TT2.FORM;%    00678
      END;%                                00679
    IF F1=POINTERS AND F2=POINTERS THEN% 00680

```

```

BEGIN%
  LT:=TT1.POINTTYPE; RT:=TT2.POINTTYPE;%
  TT1:=TYPETAB1[LT]; TT2:=TYPETAB1[RT];%
  F1:=TT1.FORM; F2:=TT2.FORM;%
END;%
WHILE F1=SUBTYPE DO%
  BEGIN LT:=TT1.MAINTYPE; TT1:=TYPETAB1[LT]; F1:=TT1.FORM END;%
WHILE F2=SUBTYPE DO%
  BEGIN RT:=TT2.MAINTYPE; TT2:=TYPETAB1[RT]; F2:=TT2.FORM END;%
IF LT>0 AND RT>0 THEN%
  IF LT<RT THEN%
  IF F1<NUMERIC OR F2<NUMERIC THEN%
  IF F1<CHAR OR F2<CHAR THEN ERROR(17);%
END;%
END;%
END OF CHECKTYPES#;%
%
%
INTEGER FILENAME;%
BOOLEAN LPARFOUND;%
%
DEFINE FILEPARAM(DEFAULTFILE)=%*** CHECKS THE FIRST PARAMETER TO SEE
BEGIN% *** IF IT IS A FILE.
  INSYMBOL; FILENAME:=CURTYPE:=0;%
  LPARFOUND:=CURSY=LPAR;%
  IF LPARFOUND THEN%
  BEGIN%
  INSYMBOL;%
  IF CURSY=IDENTIFIER THEN%
  BEGIN%
  SEARCH;%
  IF FOUND THEN%
  BEGIN%
  IF THISID.IDCLASS=VAR THEN%
  BEGIN%
  CURTYPE:=THISID.TYPE;%
  IF TYPETAB1[CURTYPE].FORM<FILES THEN%
  BEGIN%
  FILENAME:=1000*THISLEVEL+THISINDEX;%
  INSYMBOL;%
  END END END END;%
  IF SYMKIND[CURSY]=TERMINAL THEN ERROR(46);%
  END;%
  IF FILENAME=0 THEN FILENAME:=DEFAULTFILE;%
  IF (FILENAME=INPUTFILE AND NOT INPUTDECL) OR%
  (FILENAME=OUTPUTFILE AND NOT OUTPUTDECL) THEN ERROR(96);%
END OF FILEPARAM#;%
%
%
INTEGER IFORM;%
BOOLEAN SIGNED,NEGATIVE;%
%
DEFINE CONSTANT(CVAL,CTYPE)=%*** <CONSTANT> ***
BEGIN%
  IF CURSY=MINUS OR CURSY=PLUS THEN%
  BEGIN SIGNED:=TRUE; NEGATIVE:=CURSY=MINUS;%
  INSYMBOL;%

```

```

00681
00682
00683
00684
00685
00686
00687
00688
00689
00690
00691
00692
00693
00694
00695
00696
00697
00698
00699
00700
00701
00702
00703
00704
00705
00706
00707
00708
00709
00710
00711
00712
00713
00714
00715
00716
00717
00718
00719
00720
00721
00722
00723
00724
00725
00726
00727
00728
00729
00730
00731
00732
00733
00734
00735
00736
00737

```

McGraw-Hill Business Forms, Inc. 3v 14127

```

END ELSE SIGNED:=NEGATIVE:=FALSE;%
IF CURSY=INTCONST THEN%
BEGIN CTYPE:=INTTYPE;%
CVAL:=IF NEGATIVE THEN -CURVAL ELSE CURVAL;%
END ELSE%
IF CURSY=CHARCONST THEN%
BEGIN IF SIGNED THEN ERROR(29);%
CTYPE:=CHARTYPE; CVAL:=CURVAL;%
END ELSE%
IF CURSY=REALCONST THEN%
BEGIN CTYPE:=REALTYPE;%
CVAL:=IF NEGATIVE THEN -CURVAL ELSE CURVAL;%
END ELSE%
IF CURSY=ALFACONST THEN%
BEGIN IF SIGNED THEN ERROR(29);%
IF CURLength>7 THEN ERROR(41);%
CTYPE:=ALFATYPE; CVAL:=CURVAL;%
END ELSE%
IF CURSY=IDENTIFIER THEN%
BEGIN%
SEARCH;%
IF FOUND THEN%
BEGIN%
IF THISID.IDCLASS=CONST AND NOT BOOLEAN(THISID.FORMAL) THEN%
BEGIN%
IF TYPETAB1[THISID.TYPE].FORMSALFA THEN%
BEGIN%
CVAL:=THISID.INFO;%
IF CVAL>1023 THEN CVAL:=CONSTTAB[CVAL-1023];%
CTYPE:=THISID.TYPE;%
IF SIGNED THEN%
BEGIN%
TFORM:=TYPETAB1[THISID.TYPE].FORM;%
IF TFORM#NUMERIC AND TFORM#FLOATING THEN ERROR(29) ELSE%
IF NEGATIVE THEN CVAL:=-CVAL;%
END;%
END ELSE BEGIN ERROR(48); CVAL:=CTYPE:=0 END;%
END ELSE BEGIN ERROR(32); CVAL:=CTYPE:=0 END;%
END ELSE BEGIN ERROR(32); CVAL:=CTYPE:=0 END;%
INSYMBOL;%
END OF CONSTANT#;%
$ PAGE%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%
%
PART 3: THE SCANNER.
-----
%
%
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%
INTERNAL INTERNAL SYMBOL
SYMBOL NUMBER NAME KIND
%

```

```

00738
00739
00740
00741
00742
00743
00744
00745
00746
00747
00748
00749
00750
00751
00752
00753
00754
00755
00756
00757
00758
00759
00760
00761
00762
00763
00764
00765
00766
00767
00768
00769
00770
00771
00772
00773
00774
00775
00776
00777
00778
00779
00780
00781
00782
00783
00784
00785
00786
00787
00788
00789
00790
00791
00792
00793
00794

```

%	IDENTIFIER	1	IDENTIFIER	MIDDLE	00795
%	122	2	INTCONST	MIDDLE	00796
%	2.5	3	REALCONST	MIDDLE	00797
%	"ABCD"	4	ALFACONST	MIDDLE	00798
%	"C"	5	CHARCONST	MIDDLE	00799
%	NOT	6	NOTSY	MIDDLE	00800
%	*	7	ASTERISK	MIDDLE	00801
%	/	8	SLASH	MIDDLE	00802
%	& AND	9	ANDSY	MIDDLE	00803
%	DIV	10	DIVSY	MIDDLE	00804
%	MOD	11	MODSY	MIDDLE	00805
%	+	12	PLUS	MIDDLE	00806
%	-	13	MINUS	MIDDLE	00807
%	OR	14	ORSY	MIDDLE	00808
%	< LSS	15	LSSSY	MIDDLE	00809
%	<= LEQ ≤	16	LEQSY	MIDDLE	00810
%	>= GEQ ≥	17	GEQSY	MIDDLE	00811
%	> GTR	18	GTRSY	MIDDLE	00812
%	<> NEQ ≠	19	NEQSY	MIDDLE	00813
%	= EQL	20	EQLSY	MIDDLE	00814
%	IN	21	INSY	MIDDLE	00815
%	(	22	LPAR	MIDDLE	00816
%	)	23	RPAR	MIDDLE	00817
%	[	24	LBRACKET	MIDDLE	00818
%	]	25	RBRACKET	MIDDLE	00819
%	.	26	DOUBLEDOT	MIDDLE	00820
%	,	27	COMMA	MIDDLE	00821
%	;	28	SEMICOLON	TERMINAL	00822
%	.	29	DOT	MIDDLE	00823
%	← @	30	ARROW	MIDDLE	00824
%	:	31	COLON	MIDDLE	00825
%	:=	32	ASSIGNSY	MIDDLE	00826
%	BEGIN	33	BEGINSY	INITIAL	00827
%	END	34	ENDSY	TERMINAL	00828
%	IF	35	IFSY	INITIAL	00829
%	THEN	36	THENSY	MIDDLE	00830
%	ELSE	37	ELSESY	TERMINAL	00831
%	CASE	38	CASESY	INITIAL	00832
%	OF	39	OFSY	MIDDLE	00833
%	REPEAT	40	REPEATSY	INITIAL	00834
%	UNTIL	41	UNTILSY	TERMINAL	00835
%	WHILE	42	WHILESY	INITIAL	00836
%	DO	43	DOSY	MIDDLE	00837
%	FOR	44	FORSY	INITIAL	00838
%	TO	45	TOSY	MIDDLE	00839
%	DOWNTD	46	DOWNTOSY	MIDDLE	00840
%	GOTO	47	GOTOSY	INITIAL	00841
%	NIL	48	NILSY	MIDDLE	00842
%	TYPE	49	TYPESY	INITIAL	00843
%	ARRAY	50	ARRAYSY	MIDDLE	00844
%	RECORD	51	RECORDSY	MIDDLE	00845
%	FILE	52	FILESY	MIDDLE	00846
%	SET	53	SETSY	MIDDLE	00847
%	CONST	54	CONSTSY	INITIAL	00848
%	VAR	55	VARSY	INITIAL	00849
%	LABEL	56	LABELSY	INITIAL	00850
%	FUNCTION	57	FUNCSY	INITIAL	00851

%	PROCEDURE	58	PROCSY	INITIAL	00852
%	WITH	59	WITHSY	INITIAL	00853
%	PROGRAM	60	PROGRAMSY	INITIAL	00854
%	PACKED	61	PACKEDSY	MIDDLE	00855
%					00856
%					00857
%	DEFINE BLANK=48#, EQUAL=61#, QUOTES=63#, DOLLAR=42#,%				00858
%	LETTER(C)=(17<C AND C<25)OR(33<C AND C<41)OR(50<C AND C<57)#,%				00859
%	ALFANUM(C)=(LETTER(C) OR C<9)#;%				00860
%					00861
%	REAL-CURVAL;%				00862
%	ALPHA CURNAME1,CURNAME2,C,CX;%				00863
%	INTEGER CURLLENGTH, LASTCHARPOS;%				00864
%	BOOLEAN-FINIS;%				00865
%					00866
%	DEFINE NEXTCHAR=%				00867
%	BEGIN COMMENT *** READ NEXT CHARACTER. *** ;%				00868
%	IF CHARCNT=0 THEN C:=BLANK ELSE%				00869
%	BEGIN%				00870
%	REPLACE CHARPNT BY CARDPNT: CARDPNT FOR 1;%				00871
%	C:=CH[0]; CHARCNT:=CHARCNT-1;%				00872
%	END END #;%				00873
%					00874
%					00875
%					00876
%	PROCEDURE INSYMBOL;%				00877
%	BEGIN COMMENT *** READS THE NEXT SYMBOL. ***;%				00878
%	INTEGER SCALE, EXP;%				00879
%	BOOLEAN NEGEXP;%				00880
%	LABEL START, OVERFLOW;%				00881
%					00882
%	START;%				00883
%	IF C=BLANK THEN%				00884
%	BEGIN SCAN CARDPNT: CARDPNT FOR CHARCNT: CHARCNT WHILE =" ";%				00885
%	IF CHARCNT=0 THEN BEGIN NEWCARD; GO TO START END;%				00886
%	NEXTCHAR;%				00887
%	END;%				00888
%	IF LETTER(C) THEN%				00889
%	BEGIN%				00890
%	CURLLENGTH:=1; CURNAME1:=C; CURNAME2:=0;%				00891
%	NEXTCHAR;%				00892
%	WHILE ALFANUM(C) AND CURLLENGTH<6 DO%				00893
%	BEGIN CURNAME1:=C&CURNAME1[35:29:30];%				00894
%	CURLLENGTH:=CURLLENGTH+1; NEXTCHAR;%				00895
%	END;%				00896
%	IF CURLLENGTH=6 THEN%				00897
%	BEGIN%				00898
%	WHILE ALFANUM(C) AND CURLLENGTH<12 DO%				00899
%	BEGIN CURNAME2:=C&CURNAME2[35:29:30];%				00900
%	CURLLENGTH:=CURLLENGTH+1; NEXTCHAR;%				00901
%	END;%				00902
%	WHILE ALFANUM(C) DO NEXTCHAR;%				00903
%	END;%				00904
%	CURNAME1.NAMELENGTH:=CURLLENGTH;%				00905
%	CASE CURLLENGTH OF%				00906
%	BEGIN ;%				00907
%	CURSY:=IDENTIFIER;%				00908

Moore Business Forms, Inc. sv 14121



```

CURSY:=IF CURNAME1="200001F" THEN IFSY ELSE% 00909
IF CURNAME1="200000D" THEN DOSY ELSE% 00910
IF CURNAME1="200001U" THEN TOSY ELSE% 00911
IF CURNAME1="200000R" THEN ORSY ELSE% 00912
IF CURNAME1="200000F" THEN OFSY ELSE% 00913
IF CURNAME1="200001N" THEN INSY ELSE IDENTIFIER;% 00914
CURSY:=IF CURNAME1="3000END" THEN ENDSY ELSE% 00915
IF CURNAME1="3000FUR" THEN FORSY ELSE% 00916
IF CURNAME1="3000DIV" THEN DIVSY ELSE% 00917
IF CURNAME1="3000MUD" THEN MODSY ELSE% 00918
IF CURNAME1="3000NIL" THEN NILSY ELSE% 00919
IF CURNAME1="3000AND" THEN ANDSY ELSE% 00920
IF CURNAME1="3000NUT" THEN NOTSY ELSE% 00921
IF CURNAME1="3000VAR" THEN VARSY ELSE% 00922
IF CURNAME1="3000SET" THEN SETSY ELSE% 00923
IF CURNAME1="3000LSS" THEN LSSSY ELSE% 00924
IF CURNAME1="3000LEQ" THEN LEQSY ELSE% 00925
IF CURNAME1="3000GEQ" THEN GEQSY ELSE% 00926
IF CURNAME1="3000GTR" THEN GTRSY ELSE% 00927
IF CURNAME1="3000NEQ" THEN NEQSY ELSE% 00928
IF CURNAME1="3000EQL" THEN EQLSY ELSE IDENTIFIER;% 00929
CURSY:=IF CURNAME1="400THEN" THEN THENSY ELSE% 00930
IF CURNAME1="400ELSE" THEN ELSESY ELSE% 00931
IF CURNAME1="400WITH" THEN WITHSY ELSE% 00932
IF CURNAME1="400CASE" THEN CASESY ELSE% 00933
IF CURNAME1="400GOTO" THEN GOTOSY ELSE% 00934
IF CURNAME1="400TYPE" THEN TYPESY ELSE% 00935
IF CURNAME1="400FILE" THEN FILESY ELSE IDENTIFIER;% 00936
CURSY:=IF CURNAME1="50BEGIN" THEN BEGINSY ELSE% 00937
IF CURNAME1="50WHILE" THEN WHILESY ELSE% 00938
IF CURNAME1="50UNTIL" THEN UNTILSY ELSE% 00939
IF CURNAME1="50ARRAY" THEN ARRAYSY ELSE% 00940
IF CURNAME1="50CONST" THEN CONSTSY ELSE% 00941
IF CURNAME1="50LABEL" THEN LABELSY ELSE IDENTIFIER;% 00942
CURSY:=IF CURNAME1="6REPEAT" THEN REPEATSY ELSE% 00943
IF CURNAME1="6DOWNTU" THEN DOWNTOSY ELSE% 00944
IF CURNAME1="6RECORD" THEN RECORDSY ELSE% 00945
IF CURNAME1="6PACKED" THEN PACKEDSY ELSE IDENTIFIER;% 00946
CURSY:=IF CURNAME1="7PROGRA" AND CURNAME2="M" THEN PROGRAMSY% 00947
ELSE IDENTIFIER;% 00948
CURSY:=IF CURNAME1="8FUNCTI" AND CURNAME2="ON" THEN FUNCSY% 00949
ELSE IDENTIFIER;% 00950
CURSY:=IF CURNAME1="9PROCED" AND CURNAME2="URE" THEN PROCSY% 00951
ELSE IDENTIFIER;% 00952
CURSY:=IDENTIFIER; % 10 CHARACTERS. 00953
CURSY:=IDENTIFIER; % 11 CHARACTERS. 00954
CURSY:=IDENTIFIER; % 12 CHARACTERS. 00955
END OF CASE;% 00956
IF RESWORDPTION AND CURSY#IDENTIFIER THEN% 00957
BEGIN T1:=CARDLENGTH-CHARCNT-CURLNGTH;% 00958
IF CHARCNT=0 THEN CARDPNT:=CARDPNT+1 ELSE T1:=T1-1;% 00959
REPLACE X(LINEPNT+T1 BY CARDPNT-(CURLNGTH+1)% 00960
FOR CURLNGTH;% 00961
END;% 00962
END OF LETTER ELSE% 00963
IF C<=9 THEN% 00964
BEGIN% 00965

```

```
CURVAL:=C; CURSY:=[INTCONST;% 00966
NEXTCHAR;% 00967
WHILE C<9 DO BEGIN CURVAL:=10xCURVAL+C; NEXTCHAR END;% 00968
IF C="." THEN% 00969
BEGIN% 00970
NEXTCHAR;% 00971
IF C<9 THEN% 00972
BEGIN--CURSY:=REALCONST;% 00973
DO BEGIN CURVAL:=10xCURVAL+C;% 00974
SCALE:=SCALE-1; NEXTCHAR;% 00975
END--UNTIL C>9;% 00976
END ELSE IF C="." THEN C:=64 % SPECIAL MARK FOR "." 00977
ELSE ERROR(4);% 00978
END;% 00979
IF C="E" THEN% 00980
BEGIN% 00981
CURSY:=REALCONST;--NEXTCHAR;% 00982
IF C="+" OR C="-" THEN BEGIN NEGEXP:=C="-"; NEXTCHAR END;% 00983
IF C<9 THEN% 00984
BEGIN--EXP:=C;--NEXTCHAR;% 00985
WHILE C<9 DO BEGIN EXP:=10*EXP+C; NEXTCHAR END;% 00986
IF NEGEXP THEN EXP:="-EXP;% 00987
END--ELSE--ERROR(4);% 00988
SCALE:=SCALE+EXP;% 00989
END;% 00990
IF--CURSY=REALCONST--THEN% 00991
BEGIN% 00992
REALOVERFLOW:=OVERFLOW;% 00993
CURVAL:=CURVAL*10*SCALE;% 00994
REALOVERFLOW:=0;% 00995
END ELSE% 00996
IF--CURVAL>MAXINT--THEN% 00997
BEGIN% 00998
OVERFLOW: ERROR(14); CURVAL:=0; REALOVERFLOW:=0;% 00999
END;% 01000
END OF DIGIT ELSE% 01001
IF C=QUOTES THEN% 01002
BEGIN% 01003
CURSY:=ALFACONST; CURLLENGTH:=0; NEXTCHAR;% 01004
FINIS:=FALSE;% 01005
DO--BEGIN% 01006
IF C=QUOTES THEN BEGIN NEXTCHAR; FINIS:=C<QUOTES END ELSE% 01007
IF CHARCNT=0 THEN BEGIN ERROR(6); FINIS:=TRUE END;% 01008
IF NOT--FINIS--THEN% 01009
BEGIN% 01010
REPLACE STRINGPNT+CURLLENGTH BY CHARPNT FOR 1;% 01011
CURLLENGTH:=CURLLENGTH+1;% 01012
NEXTCHAR;% 01013
END END UNTIL FINIS;% 01014
IF--CURLLENGTH=0--THEN--ERROR(4)--ELSE% 01015
IF CURLLENGTH=1 THEN% 01016
BEGIN CURSY:=CHARCONST;% 01017
REPLACE--CHARPNT--BY STRINGPNT FOR 1; CURVAL:=CH[0];% 01018
END ELSE% 01019
IF CURLLENGTH<=7 THEN% 01020
BEGIN--TEXT[0]:="";% 01021
REPLACE TEXTPNT BY STRINGPNT FOR CURLLENGTH;% 01022
```

Moore Business Forms, Inc. sy 1412T

```

CURVAL:=TEXTLOI;%
END;%
END OF STRINGS ELSE%
BEGIN%
CURSY:=SYMBOLIC; NEXTCHAR;%
IF CURSY=COLON AND C=EQUAL THEN%
BEGIN CURSY:=ASSIGNSY; NEXTCHAR END ELSE%
IF CURSY=DOT AND C="." THEN%
BEGIN CURSY:=DOUBLEDOT; NEXTCHAR END ELSE%
IF CURSY=LSSSY AND C=EQUAL THEN%
BEGIN CURSY:=LEQSY; NEXTCHAR END ELSE%
IF CURSY=LSSSY AND C=">" THEN%
BEGIN CURSY:=NEQSY; NEXTCHAR END ELSE%
IF CURSY=GTRSY AND C=EQUAL THEN%
BEGIN CURSY:=GEQSY; NEXTCHAR END ELSE%
IF CURSY=LPAR AND C="*" THEN%
BEGIN
% *** COMMENT ***
NEXTCHAR;%
IF C=DOLLAR THEN % DOLLAR INDICATES COMPILER OPTIONS.
DO BEGIN%
NEXTCHAR; CX:=C; NEXTCHAR;%
IF CX="L" THEN IF C=1 THEN HEADING%
ELSE LISTOPTION:=C"+" ELSE%
IF CX="R" THEN RESWORDUPTION:=C"+" ELSE%
IF CX="C" THEN CHECKOPTION:=C"+" ELSE%
IF CX="D" THEN DUMPOPTION:=C"+" ELSE%
IF CX="X" THEN XREFOPTION:=C"+" ELSE%
IF CX="A" THEN%
IF C="+" THEN WRITE(PASCALGOL,ALIST)%
ELSE WRITE(PASCALGOL,NOALIST) ELSE%
IF CX="T" THEN%
BEGIN LASTCHARPOS:=CHARCNT - CARDLENGTH;%
CARDLENGTH:=10xC;%
NEXTCHAR; CARDLENGTH:=CARDLENGTH+C;%
IF CARDLENGTH<=9 OR CARDLENGTH>80 THEN%
BEGIN ERROR(14); CARDLENGTH:=72 END;%
CHARCNT:=MAX(0, LASTCHARPOS+CARDLENGTH-1);%
END;%
NEXTCHAR;%
END UNTIL CX=",";%
FINIS:=FALSE;%
DO BEGIN%
IF CX="*" THEN%
SCAN CARDPNT: CARDPNT FOR CHARCNT: CHARCNT UNTIL ="*";%
IF CHARCNT=0 THEN NEWCARD ELSE%
BEGIN NEXTCHAR;%
WHILE C="*" DO NEXTCHAR;%
FINIS:=C=")";%
END END UNTIL FINIS;%
NEXTCHAR;%
GO TO START;%
END OF COMMENT;%
END;%
END OF INSYMBOL;%
$ PAGE%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
```

```

01023
01024
01025
01026
01027
01028
01029
01030
01031
01032
01033
01034
01035
01036
01037
01038
01039
01040
01041
01042
01043
01044
01045
01046
01047
01048
01049
01050
01051
01052
01053
01054
01055
01056
01057
01058
01059
01060
01061
01062
01063
01064
01065
01066
01067
01068
01069
01070
01071
01072
01073
01074
01075
01076
01077
01078
01079
```



```

IF TX=1 THEN GEN(SX,1,7) ELSE%
BEGIN%
    T1:=T1+1; SX:=SYMTAB[T1];%
    IF SX.[44:6]=0 THEN GENINT(SX) ELSE GENREAL(SX);%
END END;%
NUMSYMS:=0;%
END OF WRITEEXPR#;%
%
%
DEFINE CHECKEXPR(LLIM,ULIM)=%
BEGIN%
    PUTTEXT("CHECK(");%
    EXPRESSION;%
    PUTSYM(","); PUTCONST(LLIM);%
    PUTSYM(","); PUTCONST(ULIM);%
    PUTSYM(","); PUTCONST(CARDCNT);%
    PUTSYM(")");%
END OF CHECKEXPR#;%
%
%
BOOLEAN SIMPLEVARIABLE,INSIDEBRACKETS;%
INTEGER NUMPOINTERS;%
%
PROCEDURE VARIABLE;%
BEGIN%
    INTEGER STARTSYM,LLIM,ULIM;%
    REAL T;%
    BOOLEAN INBRACKET,INRECORD;%
    LABEL ADDADDR;%
%
    STARTSYM:=NUMSYMS+1;%
    IF THISLEVEL>CURLEVEL THEN
        BEGIN
            T:=DISPLAY[THISLEVEL];%
            T4:=T.FIRSTWITHSYM; T5:=T.LASTWITHSYM;%
            FOR T3:=T4 STEP 1 UNTIL T5 DO PUTTEXT(WITHTAB[T3]);%
            INRECORD:=TRUE;%
            INBRACKET:=BOOLEAN(T.BRACKETSINWITH);%
            NUMPOINTERS:=NUMPOINTERS+T.NUMPNTRSINWITH;%
            SIMPLEVARIABLE:=FALSE;%
            CURTYPE:=T.RECTYPE; T:=TYPETAB1[CURTYPE];%
            GO TO ADDADDR;%
        END;%
    IF THISLEVEL>1 AND THISLEVEL<CURLEVEL THEN ERROR(5);%
    CURTYPE:=THISID.TYPE; SIMPLEVARIABLE:=TRUE;%
    PUTID("V",1000*THISLEVEL+THISINDEX,5);%
    INSYMBOL;%
    IF CURSY=LBRACKET OR CURSY=DOT OR CURSY=ARROW THEN%
    BEGIN%
        SIMPLEVARIABLE:=FALSE;%
        DO BEGIN%
            IF CURSY=LBRACKET THEN%
            BEGIN%
                IF NOT(INBRACKET OR INRECORD) THEN%
                BEGIN PUTSYM("["); INBRACKET:=TRUE END;%
            DO BEGIN%
                T:=TYPETAB1[CURTYPE];%

```

```

01137
01138
01139
01140
01141
01142
01143
01144
01145
01146
01147
01148
01149
01150
01151
01152
01153
01154
01155
01156
01157
01158
01159
01160
01161
01162
01163
01164
01165
01166
01167
01168
01169
01170
01171
01172
01173
01174
01175
01176
01177
01178
01179
01180
01181
01182
01183
01184
01185
01186
01187
01188
01189
01190
01191
01192
01193

```

```

LLIM:=TYPETAB2[CURTYPE]; ULM:=TYPETAB3[CURTYPE];% 01194
IF T.FORM≠ARRAYS THEN ERROR(12);% 01195
IF INRECORD THEN PUTTEXT("  +(");% 01196
INSYMBOL;% 01197
IF CHECKOPTION THEN CHECKEXPR(LLIM,ULM) ELSE EXPRESSION;% 01198
CHECKTYPES(T.INXTYPE,CURTYPE);% 01199
CURTYPE:=T.ARRTYPE;% 01200
IF INRECORD THEN% 01201
BEGIN% 01202
  IF LLIM<0 THEN BEGIN PUTSYM("+"); PUTCONST(-LLIM) END ELSE 01203
  IF LLIM>0 THEN BEGIN PUTSYM("-"); PUTCONST(LLIM) END;% 01204
  PUTSYM(")");% 01205
  IF TYPETAB1[CURTYPE].SIZE>1 THEN% 01206
  BEGIN PUTSYM("x"); PUTCONST(TYPETAB1[CURTYPE].SIZE) END;% 01207
  END ELSE IF TYPETAB1[CURTYPE].STRUCT>0 THEN PUTSYM(",");% 01208
END UNTIL CURSY≠COMMA;% 01209
IF CURSY≠RBRACKET THEN% 01210
BEGIN ERROR(59); SKIP(RBRACKET);% 01211
  IF CURSY=RBRACKET THEN INSYMBOL;% 01212
END ELSE INSYMBOL;% 01213
END OF BRACKETS ELSE% 01214
IF CURSY=DOT THEN% 01215
BEGIN% 01216
  IF NOT(INBRACKET OR INRECORD) THEN% 01217
  BEGIN PUTSYM("L"); INBRACKET:=TRUE END;% 01218
T:=TYPETAB1[CURTYPE];% 01219
IF T.FORM≠RECORD THEN ERROR(12);% 01220
INSYMBOL;% 01221
IF CURSY=IDENTIFIER THEN% 01222
BEGIN% 01223
  SEARCHTAB(T,RECTAB);% 01224
  IF FOUND THEN% 01225
  BEGIN% 01226
    THISID:=NAMETAB3(T,RECTAB,THISINDEX);% 01227
    PUTSYM("+");% 01228
    PUTCONST(THISID.INFO); CURTYPE:=THISID.TYPE;% 01229
  END ELSE BEGIN ERROR(1); CURTYPE:=0 END;% 01230
  END ELSE BEGIN ERROR(9); CURTYPE:=0 END;% 01231
  INRECORD:=TRUE;% 01232
  INSYMBOL;% 01233
END OF DOT ELSE% 01234
BEGIN % CURSY=ARROW 01235
  T:=TYPETAB1[CURTYPE];% 01236
  IF T.FORM=FILES THEN% 01237
  BEGIN% 01238
    CURTYPE:=T.FILETYPE;% 01239
    IF TYPETAB1[CURTYPE].STRUCT=0 THEN PUTTEXT(" [0]");% 01240
  END ELSE% 01241
  IF T.FORM=TEXTFILE THEN% 01242
  BEGIN% 01243
    SYMTAB[NUMSYMS]:=SYMTAB[NUMSYMS] & "1" [35:5:6];% 01244
    PUTSYM("."); PUTTEXT("LASTCH");% 01245
    CURTYPE:=CHARTYPE;% 01246
  END ELSE% 01247
  IF T.FORM=POINTERS THEN% 01248
  BEGIN% 01249
    IF INBRACKET THEN PUTSYM("]");% 01250

```

```

INBRACKET:=FALSE;%
IF NUMSYMS+2<=MAXSYMS THEN%
BEGIN%
FOR T1:=NUMSYMS STEP -1 UNTIL STARTSYM DO%
SYMTAB[T1+2]:=SYMTAB[T1];%
SYMTAB[STARTSYM]:= " MEM[";%
SYMTAB[STARTSYM+1]:= " ("T=";%
NUMSYMS:=NUMSYMS+2; NUMPOINTERS:=NUMPOINTERS+1;%
INRECORD:=TRUE;%
END ELSE ERROR(63);%
CURTYPE:=T.POINTTYPE;%
END ELSE BEGIN ERROR(12); CURTYPE:=0 END;%
INSYMBOL;%
END OF ARROW;%
END UNTIL CURSY<LBRACKET AND CURSY<DOT AND CURSY<ARROW;%
IF TYPETAB[CURTYPE].STRUCT=0 THEN%
BEGIN%
IF INBRACKET THEN PUTSYM(")");%
WHILE NUMPOINTERS>0 DO%
BEGIN PUTTEXT("-1)DIV"); PUTTEXT(" 1022,");%
PUTTEXT(" T MOD"); PUTTEXT(" 1022");%
NUMPOINTERS:=NUMPOINTERS-1;%
END;%
END;%
INSIDEBRACKETS:=INBRACKET;%
CURMODE:=NUMBER;%
END OF VARIABLE;%
%
%
PROCEDURE PASSPARAMS;%
BEGIN%
INTEGER NPARS,PARAM,P,PTYPE,P,FIRSTSYM;%
BOOLEAN FORMALPROC,CHECK;%
LABEL EXIT;%
%
PUTID("V",1000*THISLEVEL+THISINDEX,5);%
P:=THISID.INFO;%
FORMALPROC:=BOOLEAN(THISID.FORMAL);%
NPARS:=PARAMTAB[P]; P:=P+1;%
IF FORMALPROC THEN NPARS:=9999;%
INSYMBOL;%
IF CURSY=LPAR THEN%
BEGIN%
PUTSYM("(");%
DO BEGIN%
INSYMBOL;%
IF NPARS=0 THEN BEGIN ERROR(3); SKIP(RPAR); GO TO EXIT END;%
PARAM:=PARAMTAB[P]; P:=P+1;%
PTYPE:=PARAM.PARAMTYPE;%
IF PARAM.PARAMKIND=CONST THEN%
BEGIN%
CHECK:=CHECKOPTION AND TYPETAB[PTYPE].FORM LEQ CHAR;%
IF CHECK THEN PUTTEXT("CHECK(");%
PUTDUMMY; FIRSTSYM:=NUMSYMS;%
EXPRLEVEL:=EXPRLEVEL+1;%
EXPRESSION; EXPRLEVEL:=EXPRLEVEL-1;%

```

```

01251
01252
01253
01254
01255
01256
01257
01258
01259
01260
01261
01262
01263
01264
01265
01266
01267
01268
01269
01270
01271
01272
01273
01274
01275
01276
01277
01278
01279
01280
01281
01282
01283
01284
01285
01286
01287
01288
01289
01290
01291
01292
01293
01294
01295
01296
01297
01298
01299
01300
01301
01302
01303
01304
01305
01306
01307

```

```

IF CURMUDE=BITPATTERN THEN%                                01308
BEGIN SYMTAB[FIRSTSYM]:=" REAL("; PUTSYM(")"); END;%        01309
IF CHECK THEN%                                             01310
BEGIN%                                                     01311
PUTSYM(","); PUTCONST(TYPETAB2[PTYPE]);%                   01312
PUTSYM(","); PUTCONST(TYPETAB3[PTYPE]);%                   01313
PUTSYM(","); PUTCONST(CARDCNT); PUTSYM(")");%             01314
END;%                                                       01315
END ELSE%                                                  01316
IF PARAM.PARAMKIND=VAR THEN%                               01317
BEGIN%                                                     01318
IF CURSY=IDENTIFIER THEN%                                 01319
BEGIN%                                                     01320
SEARCH;%                                                    01321
IF FOUND THEN%                                           01322
BEGIN%                                                     01323
IF THISID.IDCLASS=VAR OR%                                  01324
THISID.IDCLASS=CONST AND BOOLEAN(THISID.FORMAL) THEN%   01325
BEGIN%                                                     01326
IF PARAM.PARAMFILE=1 THEN%                               01327
BEGIN%                                                     01328
CURTYPE:=THISID.TYPE;%                                    01329
PUTID("V",1000*THISLEVEL+THISINDEX,5); PUTSYM(",");%     01330
PUTID("F",1000*THISLEVEL+THISINDEX,5); PUTSYM(",");%     01331
PUTID("I",1000*THISLEVEL+THISINDEX,5);%                  01332
INSYMBOL;%                                                01333
END ELSE%                                                  01334
BEGIN%                                                     01335
VARIABLE;%                                                01336
IF TYPETAB1[CURTYPE].STRUCT>0 THEN%                       01337
IF NOT SIMPLEVARIABLE THEN ERROR(92);%                   01338
END;%                                                       01339
END ELSE BEGIN ERROR(8); CURTYPE:=0 END;%                 01340
END ELSE BEGIN ERROR(1); CURTYPE:=0 END;%                 01341
END ELSE BEGIN ERROR(9); CURTYPE:=0 END;%                 01342
END ELSE%                                                  01343
BEGIN%                                                     01344
IF CURSY=IDENTIFIER THEN%                                 01345
BEGIN%                                                     01346
SEARCH;%                                                    01347
IF FOUND THEN%                                           01348
BEGIN%                                                     01349
IF THISID.IDCLASS≠PARAM.PARAMKIND THEN ERROR(91);%       01350
PUTID("V",1000*THISLEVEL+THISINDEX,5);%                  01351
CURTYPE:=IF THISID.IDCLASS=FUNC THEN THISID.TYPE ELSE 0;% 01352
INSYMBOL;%                                                01353
END ELSE BEGIN ERROR(1); CURTYPE:=0 END;%                 01354
END ELSE BEGIN ERROR(9); CURTYPE:=0 END;%                 01355
END;%                                                       01356
CHECKTYPES(CTYPE,CURTYPE);%                               01357
NPARS:=NPARS-1;%                                           01358
IF CURSY=COMMA THEN PUTSYM(",");%                          01359
END UNTIL CURSY≠COMMA;%                                    01360
IF CURSY≠RPAR THEN BEGIN ERROR(89); SKIP(RPAR) END;%      01361
EXIT: PUTSYM(")");%                                        01362
IF CURSY=RPAR THEN INSYMBOL;%                              01363
END;%                                                       01364

```



```

IF NPARS>0 AND NOT FORMALPROC THEN ERROR(3);% 01365
CURMODE:=NUMBER;% 01366
END OF PASSPARAMS;% 01367
% 01368
% 01369
PROCEDURE FACTOR; %*** FACTOR *** 01370
BEGIN %***** 01371
INTEGER STARTSYM,STYPE,T;% 01372
BOOLEAN FIRST;% 01373
REAL VAL;% 01374
% 01375
DEFINE PARAMETER= %*** CHECK THAT THE FUNCTION HAS 1 PARAM. 01376
BEGIN% 01377
INSYMBOL;% 01378
IF CURSY=LPAR THEN% 01379
BEGIN% 01380
PUTSYM("("); INSYMBOL; EXPRESSION;% 01381
IF TYPETAB1[CURTYPE].FORM=NUMERIC THEN CURTYPE:=INTTYPE;% 01382
IF CURSY≠RPAR THEN BEGIN ERROR(3); SKIP(RPAR) END;% 01383
PUTSYM(")"); IF CURSY=RPAR THEN INSYMBOL;% 01384
END ELSE ERROR(3);% 01385
END OF PARAMETER#;% 01386
% 01387
CURMODE:=NUMBER;% 01388
IF CURSY=IDENTIFIER THEN% 01389
BEGIN% 01390
SEARCH;% 01391
IF FOUND THEN% 01392
BEGIN% 01393
IF THISID.IDCLASS=VAR OR% 01394
THISID.IDCLASS=CONST AND BOOLEAN(THISID.FORMAL)% 01395
THEN VARIABLE ELSE% 01396
IF THISID.IDCLASS=CONST THEN% 01397
BEGIN% 01398
IF THISID.INFO≤1023 THEN PUTCONST(THISID.INFO)% 01399
ELSE PUTCONST(CONSTTAB[THISID.INFO-1023]); 01400
CURTYPE:=THISID.TYPE; CURMODE:=NUMBER;% 01401
INSYMBOL;% 01402
END ELSE% 01403
IF THISID.IDCLASS=FUNC THEN% 01404
BEGIN% 01405
IF THISTAB=0 THEN %*** INTRINSIC FUNCTION *** 01406
BEGIN% 01407
INTEGER DUMMY;% 01408
IF CURNAME1="3000ABS" THEN % "ABS" 01409
BEGIN% 01410
PUTTEXT(" ABS"); PARAMETER;% 01411
IF CURTYPE≠REALTYPE AND CURTYPE≠INTTYPE THEN ERROR(67);% 01412
END ELSE% 01413
IF CURNAME1="3000CHR" THEN % "CHR" 01414
BEGIN% 01415
INSYMBOL;% 01416
IF CURSY=LPAR THEN% 01417
BEGIN INSYMBOL; CHECKEXPR(0,63);% 01418
IF TYPETAB1[CURTYPE].FORM≠NUMERIC THEN ERROR(67);% 01419
IF CURSY≠RPAR THEN BEGIN ERROR(46); SKIP(RPAR) END;% 01420
IF CURSY=RPAR THEN INSYMBOL;% 01421

```

```

END ELSE ERROR(58);%                                01422
CURTYPE:=CHARTYPE;%                                01423
END ELSE%                                           01424
IF CURNAME1="3000EOF" OR                            % "EOF"/"EOLN" 01425
CURNAME1="400EOLN" THEN%                             01426
BEGIN%                                              01427
FIRST:=CURNAME1="3000EOF";%                          01428
FILEPARAM(INPUTFILE);%                              01429
PUTID("I",FILENAME,5);%                             01430
PUTTEXT(IF FIRST THEN " .EOF" ELSE " .EOLN");%      01431
IF LPARFOUND THEN%                                  01432
BEGIN%                                              01433
IF CURSY≠RPAR THEN BEGIN ERROR(46); SKIP(RPAR) END;% 01434
IF CURSY=RPAR THEN INSYMBOL;%                      01435
END;%                                               01436
CURTYPE:=BOOLTYPE;%                                 01437
END ELSE%                                           01438
IF CURNAME1="3000ODD" THEN                          % "ODD" 01439
BEGIN%                                              01440
PUTTEXT(" .ODD"); PARAMETER;%                     01441
IF CURTYPE≠INTTYPE THEN ERROR(67);%                01442
CURTYPE:=BOOLTYPE; CURMODE:=BITPATTERN;%          01443
END ELSE%                                           01444
IF CURNAME1="3000ORD" THEN                          % "ORD" 01445
BEGIN%                                              01446
PUTSYM("("); INSYMBOL;%                             01447
IF CURSY=LPAR THEN%                                01448
BEGIN%                                              01449
INSYMBOL; EXPRESSION;%                             01450
IF TYPETAB1[CURTYPE].FORM>CHAR THEN ERROR(67);%   01451
IF CURSY≠RPAR THEN BEGIN ERROR(46); SKIP(RPAR) END;% 01452
INSYMBOL;%                                         01453
END ELSE ERROR(58);%                                01454
CURTYPE:=INTTYPE; PUTSYM(")");%                    01455
END ELSE%                                           01456
IF CURNAME1="400PRED" OR                            % "PRED"/"SUCC" 01457
CURNAME1="400SUCC" THEN%                             01458
BEGIN%                                              01459
FIRST:=CURNAME1="400PRED";%                         01460
PUTTEXT("CHECK("); INSYMBOL;%                     01461
IF CURSY=LPAR THEN%                                01462
BEGIN%                                              01463
INSYMBOL; EXPRESSION;%                             01464
PUTSYM(IF FIRST THEN "-" ELSE "+"); PUTSYM("1");%  01465
IF TYPETAB1[CURTYPE].FORM>CHAR THEN ERROR(67);%   01466
PUTSYM(","); PUTCONST(TYPETAB2[CURTYPE]);%         01467
PUTSYM(","); PUTCONST(TYPETAB3[CURTYPE]);%         01468
PUTSYM(","); PUTCONST(CARDCNT);%                  01469
PUTSYM(")");%                                       01470
IF CURSY≠RPAR THEN BEGIN ERROR(46); SKIP(RPAR) END;% 01471
IF CURSY=RPAR THEN INSYMBOL;%                      01472
END ELSE BEGIN ERROR(58); CURTYPE:=0 END;%         01473
END ELSE%                                           01474
IF CURNAME1="50ROUND" THEN                          % "ROUND" 01475
BEGIN%                                              01476
PUTTEXT(" .ROUND"); PARAMETER;%                   01477
IF CURTYPE≠REALTYPE THEN ERROR(67);%              01478

```

Moore Business Forms, Inc. sv 1412T

```

NUMSYMS:=NUMSYMS-1; PUTSYM(",");% 01479
PUTCONST(CARDCNT); PUTSYM(")");% 01480
CURTYPE:=INTTYPE;% 01481
END ELSE% 01482
IF CURNAME1="3000SQR" THEN % "SQR" 01483
BEGIN% 01484
PUTTEXT(" SQR"); PARAMETER;% 01485
NUMSYMS:=NUMSYMS-1; PUTSYM(",");% 01486
PUTCONST(CARDCNT); PUTSYM(")");% 01487
IF CURTYPE≠REALTYPE AND CURTYPE≠INTTYPE THEN ERROR(67);% 01488
END ELSE% 01489
IF CURNAME1="50TRUNC" THEN % "TRUNC" 01490
BEGIN% 01491
PUTTEXT(" TRUNC"); PARAMETER;% 01492
NUMSYMS:=NUMSYMS-1; PUTSYM(",");% 01493
PUTCONST(CARDCNT); PUTSYM(")");% 01494
IF CURTYPE≠REALTYPE THEN ERROR(67);% 01495
CURTYPE:=INTTYPE;% 01496
END ELSE% 01497
IF CURNAME1="6CONCAT" THEN % "CONCAT" 01498
CONCAT ELSE% 01499
IF CURNAME1="400TIME" THEN % "TIME" 01500
BEGIN% 01501
PUTTEXT("(TIME("); PUTTEXT("1)/60)");% 01502
CURTYPE:=REALTYPE; INSYMBOL;% 01503
END ELSE% 01504
IF CURNAME1="400DATE" THEN % "DATE" 01505
BEGIN% 01506
PUTTEXT("CURDAT");% 01507
CURTYPE:=ALFATYPE; INSYMBOL;% 01508
END ELSE% 01509
IF CURNAME1="7ELAPSE" AND CURNAME2="D" THEN % "ELAPSED" 01510
BEGIN% 01511
PUTTEXT("(TIME("); PUTTEXT("2)/60)");% 01512
CURTYPE:=REALTYPE; INSYMBOL;% 01513
END ELSE% 01514
IF CURNAME1="610TIME" THEN % "IOTIME" 01515
BEGIN% 01516
PUTTEXT("(TIME("); PUTTEXT("3)/60)");% 01517
CURTYPE:=REALTYPE; INSYMBOL;% 01518
END ELSE% 01519
IF CURNAME1="7WEEKDA" AND CURNAME2="Y" THEN % "WEEKDAY" 01520
BEGIN% 01521
PUTTEXT("WEEKDA");% 01522
CURTYPE:=ALFATYPE; INSYMBOL;% 01523
END ELSE IF CURNAME1="400USER" THEN % "USER" 01524
BEGIN% 01525
PUTTEXT(" TIME"); PUTTEXT(" (-1)");% 01526
CURTYPE:=ALFATYPE; INSYMBOL;% 01527
END ELSE % "SIN","COS" ETC. 01528
BEGIN% 01529
PUTTEXT(IF CURNAME1="3000SIN" THEN " SIN" ELSE% 01530
IF CURNAME1="3000CUS" THEN " COS" ELSE% 01531
IF CURNAME1="6ARCTAN" THEN "ARCTAN" ELSE% 01532
IF CURNAME1="400SQRT" THEN " SQRT" ELSE% 01533
IF CURNAME1="3000EXP" THEN " EXP" ELSE% 01534
" LN");% 01535

```

```

PARAMETER;% 01536
IF CURTYPE≠REALTYPE AND CURTYPE≠INITYPE THEN ERROR(67);% 01537
CURTYPE:=REALTYPE;% 01538
END;% 01539
END OF INTRINSIC FUNCTIONS ELSE% 01540
BEGIN% 01541
T:=THISID.TYPE;% 01542
PASSPARAMS;% 01543
CURTYPE:=T;% 01544
END;% 01545
END OF FUNCTIONS ---ELSE% 01546
IF THISID.IDCLASS=PROC THEN% 01547
BEGIN% 01548
ERROR(68); PASSPARAMS;% 01549
CURTYPE:=0;% 01550
END ELSE BEGIN ERROR(69); CURTYPE:=0; INSYMBOL END;% 01551
END ELSE BEGIN ERROR(1); CURTYPE:=0; INSYMBOL END;% 01552
END OF IDENTIFIER ELSE% 01553
IF CURSY≤CHARCONST THEN% 01554
BEGIN% 01555
CONSTANT(VAL,CURTYPE); PUTCONST(VAL);% 01556
END ELSE% 01557
IF CURSY=NOTSY THEN% 01558
BEGIN% 01559
PUTTEXT(" NOT "); PUTDUMMY; STARTSYM:=NUMSYMS;% 01560
INSYMBOL; FACTOR;% 01561
IF CURTYPE>0 THEN% 01562
IF CURTYPE≠BOOLTYPE THEN BEGIN ERROR(17); CURTYPE:=0 END;% 01563
IF CURMODE=NUMBER THEN% 01564
BEGIN SYMTAB[STARTSYM]:=" B("; PUTSYM(")");% 01565
CURMODE:=BITPATTERN;% 01566
END;% 01567
END ELSE% 01568
IF CURSY=NILSY THEN% 01569
BEGIN% 01570
PUTCONST(0); CURTYPE:=NILTYPE;% 01571
INSYMBOL;% 01572
END ELSE% 01573
IF CURSY=LPAR THEN% 01574
BEGIN% 01575
PUTSYM("(");% 01576
INSYMBOL; EXPRESSION;% 01577
IF CURSY≠RPAR THEN BEGIN ERROR(46); SKIP(RPAR) END;% 01578
PUTSYM(")");% 01579
INSYMBOL;% 01580
END ELSE% 01581
IF CURSY=LBRACKET THEN %*** SET CONSTANT *** 01582
BEGIN% 01583
INSYMBOL;% 01584
IF CURSY=RBRACKET THEN% 01585
BEGIN% 01586
PUTCONST(0); CURTYPE:=EMPTYSET; CURMODE:=NUMBER;% 01587
INSYMBOL;% 01588
END ELSE% 01589
BEGIN% 01590
FIRST:=TRUE;% 01591
DO BEGIN% 01592

```

Moore Business Forms, Inc. sv 14121

```

IF FIRST THEN FIRST:=FALSE ELSE INSYMBOL;% 01593
PUTTEXT("---BIT("); STARTSYM:=NUMSYMS;% 01594
EXPRESSION;% 01595
IF STYPE=0 THEN% 01596
BEGIN STYPE:=CURTYPE;% 01597
IF TYPETAB1[CURTYPE].FORM>CHAR THEN ERROR(72);% 01598
END ELSE CHECKTYPES(STYPE,CURTYPE);% 01599
IF CURSY=DOUBLEDOT THEN% 01600
BEGIN% 01601
PUTSYM(","); SYMTAB[STARTSYM]:=" BITS(";% 01602
INSYMBOL; EXPRESSION;% 01603
IF STYPE=0 THEN% 01604
BEGIN STYPE:=CURTYPE;% 01605
IF TYPETAB1[CURTYPE].FORM>CHAR THEN ERROR(72);% 01606
END ELSE CHECKTYPES(STYPE,CURTYPE);% 01607
END;% 01608
PUTSYM(","); PUTCONST(CARDCNT); PUTSYM(")");% 01609
IF CURSY=COMMA THEN PUTTEXT(" OR");% 01610
END UNTIL CURSY<COMMA;% 01611
IF CURSY<RBRACKET THEN% 01612
BEGIN ERROR(59); SKIP(RBRACKET);% 01613
IF CURSY=RBRACKET THEN INSYMBOL;% 01614
END ELSE INSYMBOL;% 01615
NEWTYPER; T1:=SET; T1.SIZE:=1; T1.STRUCT:=0;% 01616
T1.SETTYPE:=STYPE; TYPETAB1[TYPEINDEX]:=T1;% 01617
CURTYPE:=TYPEINDEX;% 01618
CURMODE:=BITPATTERN;% 01619
END;% 01620
END OF SET CONSTANT ELSE BEGIN ERROR(99); INSYMBOL END;% 01621
END OF FACTOR;% 01622
% 01623
% 01624
PROCEDURE TERM; %*** TERM *** 01625
BEGIN %***** 01626
INTEGER STARTSYM,MODE,TYPE1,MULOPTR,F;% 01627
PUTDUMMY; STARTSYM:=NUMSYMS;% 01628
FACTOR;% 01629
MODE:=CURMODE;% 01630
WHILE CURSY>ASTERISK AND CURSY<MODSY DO % "*" , "/" , "DIV" , "MOD" , "AND" 01631
BEGIN% 01632
TYPE1:=CURTYPE; MULOPTR:=CURSY;% 01633
F:=TYPETAB1[TYPE1].FORM;% 01634
IF F=NUMERIC OR F=FLOATING THEN% 01635
BEGIN% 01636
MODE:=NUMBER;% 01637
IF CURSY=ASTERISK THEN PUTSYM("x") ELSE% 01638
IF CURSY=SLASH THEN PUTSYM("/") ELSE% 01639
IF CURSY=ANDSY THEN ERROR(64) ELSE% 01640
BEGIN% 01641
IF F=FLOATING THEN ERROR(64);% 01642
IF CURSY=DIVSY THEN PUTTEXT(" DIV") ELSE PUTTEXT(" MOD");% 01643
END END ELSE% 01644
IF CURTYPE=BOOLTYPE OR F=SET THEN% 01645
BEGIN% 01646
MODE:=BITPATTERN;% 01647
IF CURMODE<MODE THEN% 01648
BEGIN SYMTAB[STARTSYM]:=" B("; PUTSYM(")"); END;% 01649

```

```

PUTTEXT(" AND ");%
IF CURSY≠(IF F=SET THEN ASTERISK ELSE ANDSY) THEN ERROR(64);%
END ELSE ERROR(64);%
PUTDUMMY; STARTSYM:=NUMSYMS;%
INSYMBOL; FACTOR;%
IF CURTYPE>0 AND TYPE1>0 THEN%
BEGIN%
IF CURTYPE≠TYPE1 THEN%
BEGIN%
IF TYPETAB1[TYPE1].FORM≠NUMERIC OR CURTYPE≠REALTYPE THEN%
CHECKTYPES(TYPE1,CURTYPE);%
IF TYPE1=REALTYPE THEN CURTYPE:=REALTYPE;%
END;%
IF CURTYPE=REALTYPE AND MULOPTR≥DIVSY THEN ERROR(65);%
END;%
IF MULOPTR=SLASH THEN CURTYPE:=REALTYPE;%
IF CURTYPE=0 THEN CURTYPE:=TYPE1;%
END OF WHILE LOOP;%
IF MODE=BITPATTERN AND CURMODE≠MODE THEN%
BEGIN SYMTAB[STARTSYM]:=" " B("; PUTSYM(")") END;%
CURMODE:=MODE;%
END OF TERM;%
%
%
PROCEDURE SIMPLEEXPRESSION; %*** SIMPLE EXPRESSION ***
BEGIN %*****
INTEGER STARTSYM,MODE,TYPE1,F;%
BOOLEAN SIGNED;%
%
PUTDUMMY; STARTSYM:=NUMSYMS;%
IF CURSY=PLUS OR CURSY=MINUS THEN%
BEGIN SIGNED:=TRUE;%
PUTSYM(IF CURSY=PLUS THEN"+" ELSE "-");%
INSYMBOL;%
END;%
TERM;%
MODE:=CURMODE;%
IF SIGNED THEN%
BEGIN F:=TYPETAB1[CURTYPE].FORM;%
IF F≠NUMERIC AND F≠FLOATING THEN ERROR(29);%
END;%
WHILE CURSY≥PLUS AND CURSY≤ORSY DO % "+","-","OR"
BEGIN%
TYPE1:=CURTYPE; F:=TYPETAB1[TYPE1].FORM;%
IF F=NUMERIC OR F=FLOATING THEN%
BEGIN MODE:=NUMBER;%
IF CURSY=PLUS THEN PUTSYM("+") ELSE%
IF CURSY=MINUS THEN PUTSYM("-") ELSE ERROR(64);%
END ELSE%
IF CURTYPE=BOOLTYPE THEN%
BEGIN%
MODE:=BITPATTERN;%
IF CURMODE≠MODE THEN%
BEGIN SYMTAB[STARTSYM]:=" " B("; PUTSYM(")") END;%
IF CURSY=ORSY THEN PUTTEXT(" OR") ELSE ERROR(64);%
END ELSE%
IF F=SET THEN%

```

```

01650
01651
01652
01653
01654
01655
01656
01657
01658
01659
01660
01661
01662
01663
01664
01665
01666
01667
01668
01669
01670
01671
01672
01673
01674
01675
01676
01677
01678
01679
01680
01681
01682
01683
01684
01685
01686
01687
01688
01689
01690
01691
01692
01693
01694
01695
01696
01697
01698
01699
01700
01701
01702
01703
01704
01705
01706

```

```

BEGIN%
MODE:=BITPATTERN;%
IF CURMODE≠MODE THEN%
BEGIN SYMTAB[STARTSYM]:=" B("; PUTSYM(")") END;%
IF CURSY=PLUS THEN PUTTEXT(" OR") ELSE%
IF CURSY=MINUS THEN BEGIN PUTTEXT(" AND"); PUTTEXT(" NOT ")END
ELSE ERROR(64);%
END ELSE ERROR(64);%
INSYMBOL;%
PUTDUMMY; STARTSYM:=NUMSYMS;%
TERM;%
IF CURTYPE>0 AND TYPE1>0 THEN%
BEGIN%
IF CURTYPE≠TYPE1 THEN%
BEGIN%
IF TYPETAB1[TYPE1].FORM≠NUMERIC OR CURTYPE≠REALTYPE THEN%
CHECKTYPES(TYPE1,CURTYPE);%
IF TYPE1=REALTYPE THEN CURTYPE:=REALTYPE;%
END END;%
IF CURTYPE=0 THEN CURTYPE:=TYPE1;%
END OF WHILE LOOP;%
IF MODE=BITPATTERN AND CURMODE≠BITPATTERN THEN%
BEGIN SYMTAB[STARTSYM]:=" B("; PUTSYM(")") END;%
CURMODE:=MODE;%
END OF SIMPLEEXPRESSION;%
%
%
PROCEDURE EXPRESSION; %*** EXPRESSION ***
BEGIN %*****
INTEGER STARTSYM,FIRSTSYM,TYPE1,RELOPTR,F;%
BOOLEAN CALLGEN;%
%
EXPRLEVEL:=EXPRLEVEL+1;%
IF EXPRLEVEL = 1 THEN%
BEGIN%
PUTDUMMY;%
FIRSTSYM := NUMSYMS;%
END;%
PUTDUMMY; STARTSYM:=NUMSYMS;%
PUTDUMMY;%
SIMPLEEXPRESSION;%
IF CURSY≥LSSSY AND CURSY≤INSY THEN % "<","≤","≥",">","=","≠","IN"
BEGIN%
TYPE1:=CURTYPE; F:=TYPETAB1[TYPE1].FORM;%
RELOPTR:=CURSY;%
IF F≠ALFA THEN%
BEGIN%
IF CURMODE=BITPATTERN THEN%
BEGIN SYMTAB[STARTSYM]:=" REAL("; PUTSYM(")") END;%
IF CURSY=LSSSY THEN PUTSYM("<") ELSE%
IF CURSY=LEQSY THEN PUTSYM("≤") ELSE%
IF CURSY=GEQSY THEN PUTSYM("≥") ELSE%
IF CURSY=GTRSY THEN PUTSYM(">") ELSE%
IF CURSY=EQLSY THEN PUTSYM("=") ELSE%
IF CURSY=NEQSY THEN PUTSYM("≠") ELSE%
BEGIN%
IF F≠FLOATING THEN ERROR(64);%

```

```

01707
01708
01709
01710
01711
01712
01713
01714
01715
01716
01717
01718
01719
01720
01721
01722
01723
01724
01725
01726
01727
01728
01729
01730
01731
01732
01733
01734
01735
01736
01737
01738
01739
01740
01741
01742
01743
01744
01745
01746
01747
01748
01749
01750
01751
01752
01753
01754
01755
01756
01757
01758
01759
01760
01761
01762
01763

```

```

SYMTAB[STARTSYM]:= "INTST("; PUTSYM(","); CALLGEN:=TRUE;% 01764
END;% 01765
END ELSE% 01766
IF F=SET THEN% 01767
BEGIN% 01766
IF CURMODE=BITPATTERN THEN% 01769
BEGIN SYMTAB[STARTSYM+1]:= " REAL("; PUTSYM(")") END;% 01770
IF CURSY=EQLSY OR CURSY=NEQSY THEN% 01771
BEGIN PUTSYM(IF CURSY=EQLSY THEN "=" ELSE "!=");% 01772
END ELSE% 01773
BEGIN% 01774
IF CURSY=LEQSY THEN SYMTAB[STARTSYM]:= "INCL1(" ELSE% 01775
IF CURSY=GEQSY THEN SYMTAB[STARTSYM]:= "INCL2(" ELSE ERROR(64); 01776
PUTSYM(","); CALLGEN:=TRUE;% 01777
END END ELSE% 01778
IF F=POINTERS THEN% 01779
BEGIN% 01780
IF CURSY=EQLSY THEN PUTSYM("=") ELSE% 01781
IF CURSY=NEQSY THEN PUTSYM("!=") ELSE ERROR(64);% 01782
END ELSE--ERROR(64);% 01783
INSYMBOL;% 01784
PUTDUMMY; STARTSYM:=NUMSYMS;% 01785
SIMPLEEXPRESSION;% 01786
IF CURTYPE>0 AND TYPE1>0 THEN% 01787
IF CURTYPE≠TYPE1 THEN% 01788
IF RELOPTR≠INSY THEN% 01789
BEGIN% 01790
IF TYPETAB1[TYPE1].FORM≠NUMERIC OR CURTYPE≠REALTYPE THEN% 01791
CHECKTYPES(TYPE1,CURTYPE);% 01792
END ELSE% 01793
IF TYPETAB1[CURTYPE].FORM≠SET THEN ERROR(66)% 01794
ELSE CHECKTYPES(TYPE1,TYPETAB1[CURTYPE].SETTYPE);% 01795
IF CURMODE=BITPATTERN THEN% 01796
BEGIN SYMTAB[STARTSYM]:= " REAL("; PUTSYM(")") END;% 01797
IF CALLGEN THEN PUTSYM(")");% 01798
CURTYPE:=BOOLTYPE; CURMODE:=BITPATTERN;% 01799
END;% 01800
EXPRLEVEL:=EXPRLEVEL-1;% 01801
IF EXPRLEVEL=0 THEN% 01802
BEGIN% 01803
IF CURMODE=BITPATTERN THEN% 01804
BEGIN% 01805
SYMTAB[FIRSTSYM] := " REAL("% 01806
PUTSYM(")");% 01807
END;% 01808
WRITEEXPR;% 01809
END;% 01810
END OF EXPRESSION;% 01811
% 01812
% 01813
DEFINE BOOLEXPRESS=% 01814
BEGIN% 01815
PUTDUMMY; EXPRLEVEL:=1; EXPRESSION;% 01816
IF CURTYPE>0 THEN IF CURTYPE≠BOOLTYPE THEN ERROR(17);% 01817
IF CURMODE≠BITPATTERN THEN% 01818
BEGIN SYMTAB[1]:= " B("; PUTSYM(")") END;% 01819
EXPRLEVEL:=0; WRITEEXPR;% 01820

```



END OF BOOLEXPRESS#;%

01821

\$ PAGE%

01822

%%%

01823

%

01824

%

01825

%

01826

%

PART 5: INTRINSIC ROUTINES.

01827

%

01828

%

01829

%

01830

%

01831

%%%

01832

%

01833

%

01834

PROCEDURE CONCAT;

\*\*\* "CONCAT" \*\*\*

01835

BEGIN

\*\*\*\*\*

01836

DEFINE INEXPR=%

01837

BEGIN INSYMBOL; EXPRESSION;%

01838

IF CURTYPE>0 THEN%

01839

IF TYPETAB1[CURTYPE].FORM#NUMERIC THEN ERROR(17);%

01840

END #;%

01841

%

01842

PUTTEXT("CONCAT"); PUTSYM("(");%

01843

INSYMBOL;%

01844

IF CURSY=LPAR THEN%

01845

BEGIN%

01846

INSYMBOL; EXPRESSION;%

01847

IF CURTYPE>0 THEN%

01848

IF TYPETAB1[CURTYPE].FORM>ALFA THEN ERROR(17);%

01849

IF CURSY=COMMA THEN%

01850

BEGIN%

01851

PUTSYM(","); INSYMBOL; EXPRESSION;%

01852

IF CURTYPE>0 THEN%

01853

IF TYPETAB1[CURTYPE].FORM>ALFA THEN ERROR(17);%

01854

IF CURSY=COMMA THEN%

01855

BEGIN%

01856

PUTSYM(","); INEXPR;%

01857

IF CURSY=COMMA THEN%

01858

BEGIN%

01859

PUTSYM(","); INEXPR;%

01860

IF CURSY=COMMA THEN%

01861

BEGIN%

01862

PUTSYM(","); INEXPR;%

01863

PUTSYM(","); PUTCONST(CARDCNT);%

01864

PUTSYM(")");%

01865

IF CURSY#RPAR THEN BEGIN ERROR(3); SKIP(RPAR) END;%

01866

END ELSE BEGIN ERROR(3); SKIP(RPAR) END;%

01867

END ELSE BEGIN ERROR(3); SKIP(RPAR) END;%

01868

END ELSE BEGIN ERROR(3); SKIP(RPAR) END;%

01869

END ELSE BEGIN ERROR(3); SKIP(RPAR) END;%

01870

END ELSE BEGIN ERROR(3); SKIP(RPAR) END;%

01871

CURTYPE:=REALTYPE;%

01872

IF CURSY=RPAR THEN INSYMBOL;%

01873

END OF CONCAT;%

01874

%

01875

%

01876

PROCEDURE PREAD(CHANGELINE);%

01877

14121 Moore Business Forms, Inc. sv

VALUE CHANGELINE; BOOLEAN CHANGELINE;%	01878
BEGIN%	01879
INTEGER FILEID,F;%	01880
BOOLEAN CHECK;%	01881
GEN("BEGIN",7,2);%	01882
FILEPARAM(INPUTFILE); FILEID:=FILENAME;%	01883
IF TYPETAB1[CURTYPE].FORM=FILES THEN ERROR(85);%	01884
IF SYMKIND[CURSY]≠TERMINAL THEN%	01885
BEGIN%	01886
IF CURSY NEQ RPAR THEN%	01887
DO-BEGIN%	01888
WHILE CURSY=COMMA DO INSYMBOL;%	01889
IF CURSY=IDENTIFIER THEN%	01890
BEGIN%	01891
SEARCH;%	01892
IF FOUND THEN%	01893
BEGIN%	01894
IF THISID.IDCLASS=VAR OR%	01895
THISID.IDCLASS=CONST AND BOOLEAN(THISID,FORMAL) THEN%	01896
BEGIN%	01897
VARIABLE; F:=TYPETAB1[CURTYPE].FORM;%	01898
IF F=NUMERIC OR F=FLOATING OR F=CHAR THEN%	01899
BEGIN%	01900
CHECK:=CHECKOPTION AND F≠FLOATING;%	01901
WRITEEXPR; GEN("=",2,6);%	01902
IF CHECK THEN GEN("CHECK",6,2);%	01903
GEN("PREAD",6,2); GENID("F",FILEID,5); GEN(",",1,7);%	01904
GENID("V",FILEID,5); GEN(",",1,7);%	01905
GENID("I",FILEID,5); GEN(",",1,7);%	01906
IF F=NUMERIC THEN GENINT(2) ELSE%	01907
IF F=FLOATING THEN GENINT(3) ELSE GENINT(1);%	01908
GEN(",",1,7); GENINT(CARDCNT); GEN(",",1,7);%	01909
IF CHECK THEN%	01910
BEGIN%	01911
GEN(",",1,7); GENINT(TYPETAB2[CURTYPE]); GEN(",",1,7);%	01912
GENINT(TYPETAB3[CURTYPE]); GEN(",",1,7);%	01913
GENINT(CARDCNT); GEN(",",1,7);%	01914
END;%	01915
END ELSE BEGIN ERROR(82); INSYMBOL END;%	01916
END ELSE BEGIN ERROR(8); INSYMBOL END;%	01917
END ELSE BEGIN ERROR(1); INSYMBOL END;%	01918
END ELSE ERROR(9);%	01919
GEN(",",1,7);%	01920
END UNTIL CURSY≠COMMA;%	01921
IF CURSY≠RPAR THEN BEGIN ERROR(46); SKIP(RPAR) END;%	01922
IF CURSY=RPAR THEN INSYMBOL;%	01923
END;%	01924
IF CHANGELINE THEN%	01925
BEGIN%	01926
GEN("RLINE",6,2); GENID("F",FILEID,5); GEN(",",1,7);%	01927
GENID("V",FILEID,5); GEN(",",1,7);%	01928
GENID("I",FILEID,5); GEN(",",1,7);%	01929
END;%	01930
GEN("END",4,5);%	01931
END OF PREAD;%	01932
%	01933
%	01934

```

PROCEDURE PWRITE(LINEFEED);%
VALUE LINEFEED; BOOLEAN LINEFEED;%
BEGIN%
  INTEGER FILEID,F,I, LASTSY;%
  POINTER P;%
  GEN(" BEGIN",7,2);%
  FILEPARAM(OUTPUTFILE); FILEID:=FILENAME;%
  IF TYPETAB1[CURTYPE].FORM=FILES THEN ERROR(85);%
  IF SYMKIND[CURSY]#TERMINAL THEN%
  BEGIN%
    IF CURSY NEQ RPAR THEN%
    DO BEGIN%
      WHILE CURSY=COMMA DO INSYMBOL;%
      IF CURSY=ALFACONST AND CURLLENGTH>7 THEN%
      BEGIN%
        GEN("WALFA(",6,2); GENID("F",FILEID,5); GEN(", ",1,7);%
        GENID("V",FILEID,5); GEN(", ",1,7);%
        GENID("I",FILEID,5); GEN(", ",1,7);%
        P:=STRINGPNT;%
        FOR I:=1 STEP 7 UNTIL 80 DO%
        IF I<CURLLENGTH THEN%
        BEGIN%
          IF ALGOLCNT<10 THEN WRITEALGOL;%
          REPLACE ALGOLPNT:ALGOLPNT BY " ", P:P FOR 7, " ", ", ";%
          ALGOLCNT:=ALGOLCNT+10;%
        END ELSE GEN("0",2,6);%
        GENINT(CURLLENGTH); GEN(", ",1,7);%
        GENINT(CARDCNT); GEN(", ",1,7);%
        INSYMBOL;%
      END OF ALFACONST ELSE%
      BEGIN%
        GEN("PWRITE(",7,1); GENID("F",FILEID,5); GEN(", ",1,7);%
        GENID("V",FILEID,5); GEN(", ",1,7);%
        GENID("I",FILEID,5); GEN(", ",1,7);%
        LASTSY:=CURSY;%
        EXPRESSION; F:=TYPETAB1[CURTYPE].FORM;%
        GEN(", ",1,7);%
        IF F=NUMERIC OR F=FLOATING OR F=CHAR OR F=ALFA OR%
        CURTYPE=BOOLTYPE THEN%
        BEGIN%
          IF F=NUMERIC THEN GENINT(1) ELSE%
          IF F=FLOATING THEN GENINT(2) ELSE%
          IF F=ALFA THEN GENINT(5) ELSE%
          IF F=CHAR THEN GENINT(4) ELSE GENINT(3);%
          GEN(", ",1,7);%
          IF CURSY=COLON THEN%
          BEGIN%
            INSYMBOL; EXPRESSION;%
            IF TYPETAB1[CURTYPE].FORM NEQ NUMERIC THEN ERROR(17);%
            GEN(", ",1,7);%
            IF CURSY=COLON THEN%
            BEGIN%
              IF F#FLOATING THEN ERROR(4);%
              INSYMBOL; EXPRESSION;%
              IF TYPETAB1[CURTYPE].FORM NEQ NUMERIC THEN ERROR(17);%
              GEN(", ",1,7);%
            END ELSE GEN("-1",3,5);%
          END ELSE GEN("-1",3,5);%
        END ELSE GEN("-1",3,5);%
      END ELSE GEN("-1",3,5);%
    END ELSE GEN("-1",3,5);%
  END ELSE GEN("-1",3,5);%

```

01935  
01936  
01937  
01938  
01939  
01940  
01941  
01942  
01943  
01944  
01945  
01946  
01947  
01948  
01949  
01950  
01951  
01952  
01953  
01954  
01955  
01956  
01957  
01958  
01959  
01960  
01961  
01962  
01963  
01964  
01965  
01966  
01967  
01968  
01969  
01970  
01971  
01972  
01973  
01974  
01975  
01976  
01977  
01978  
01979  
01980  
01981  
01982  
01983  
01984  
01985  
01986  
01987  
01988  
01989  
01990  
01991

```

END ELSE%
BEGIN%
  IF F=FLOATING THEN GENINT(16) ELSE%
  IF F=ALFA AND LASTSY=ALFACUNST THEN GENINT(CURLENGTH) ELSE
  IF F=ALFA THEN GENINT(7) ELSE%
  IF F=CHAR THEN GENINT(1) ELSE GENINT(10);%
  GEN(",1",4,4);%
END;%
END ELSE ERROR(17);%
GENINT(CARDCNT); GEN(",1,7");%
END OF EXPRESSION;%
GEN(";",1,7);%
END UNTIL CURSY=COMMA;%
IF CURSY=RPAR THEN BEGIN ERROR(46); SKIP(RPAR) END;%
IF CURSY=RPAR THEN INSYMBOL;%
END;%
FILENAME:=FILEID;%
IF LINEFEED THEN%
BEGIN%
  INTEGER DUMMY;%
  GEN("WLINE(",6,2); GENID("F",FILENAME,5); GEN(",1,7");%
  GENID("V",FILENAME,5); GEN(",1,7");%
  GENID("I",FILENAME,5); GEN(",1,7");%
END;%
GEN("END",4,5);%
END OF PWRITE;%
%
%
PROCEDURE FILEHANDLING(PROCNUM);      %*** FILE HANDLING PROCEDURES:
VALUE PROCNUM; INTEGER PROCNUM;      %***
BEGIN                                  %***
  INTEGER F;                            %***
  CASE PROCNUM OF                       %***
  BEGIN ;                                %***
  GEN("PUT",3,5);                         %***
  GEN("GET",3,5); %                       %***
  GEN("RESET",5,3); %                     %***
  GEN("REWRITE",7,1); %                  %***
  GEN("PAGE",4,4); %                     %***
  END; %                                  %***
  GEN(",1,7"); FILEPARAM(0); %           %***
  IF FILENAME=0 THEN ERROR(78); %       %***
  F:=TYPETAB1[CURTYPE].FORM;%          %***
  IF F=FILES AND PROCNUM=5 THEN ERROR(80);%
  GENID("F",FILENAME,5); GEN(",1,7");%
  GENID("V",FILENAME,5); GEN(",1,7");%
  GENID("I",FILENAME,5); GEN(",1,7");%
  GENINT(CARDCNT); GEN(",1,7");%
  IF CURSY=RPAR THEN BEGIN ERROR(46); SKIP(RPAR) END;%
  IF CURSY=RPAR THEN INSYMBOL;%
END OF FILEHANDLING;%
%
%
PROCEDURE PACK;%
BEGIN%
  INTEGER IT,T;%
  GEN("PACK(",5,3);%

```

```

01992
01993
01994
01995
01996
01997
01998
01999
02000
02001
02002
02003
02004
02005
02006
02007
02008
02009
02010
02011
02012
02013
02014
02015
02016
02017
02018
02019
02020
02021
02022
02023
02024
02025
02026
02027
02028
02029
02030
02031
02032
02033
02034
02035
02036
02037
02038
02039
02040
02041
02042
02043
02044
02045
02046
02047
02048

```

Moore Business Forms, Inc. sy 14121

```

INSYMBOL;%
IF CURSY=LPAR THEN%
BEGIN%
INSYMBOL;%
IF CURSY=IDENTIFIER THEN%
BEGIN%
SEARCH;%
IF FOUND THEN%
BEGIN%
IF THISID.IDCLASS=VAR THEN%
BEGIN%
T:=TYPETAB1[THISID.TYPE];%
IF T.FORM=ARRAYS THEN%
BEGIN%
IT:=T.INXTYPE;%
IF TYPETAB1[T.ARRTYPE].FORM#CHAR THEN ERROR(88);%
GENID("V",1000*THISLEVEL+THISINDEX,5);%
IF THISLEVEL>1 AND THISLEVEL#CURLEVEL THEN ERROR(5);%
GEN(", ",1,7); GENINT(TYPETAB2[THISID.TYPE]);%
GEN(", ",1,7); GENINT(TYPETAB3[THISID.TYPE]);%
END ELSE ERROR(88);%
END ELSE ERROR(88);%
END ELSE ERROR(1);%
END ELSE ERROR(9);%
INSYMBOL;%
IF CURSY=COMMA THEN%
BEGIN%
GEN(", ",1,7);%
INSYMBOL; EXPRESSION; CHECKTYPES(IT,CURTYPE);%
IF CURSY=COMMA THEN%
BEGIN%
GEN(", ",1,7);%
INSYMBOL;%
IF CURSY=IDENTIFIER THEN%
BEGIN%
SEARCH;%
IF FOUND THEN%
BEGIN%
IF THISID.IDCLASS=VAR OR%
THISID.IDCLASS=CONST AND BOOLEAN(THISID.FORMAL) THEN%
BEGIN%
VARIABLE; WRITEEXPR;%
IF CURTYPE>0 THEN%
IF TYPETAB1[CURTYPE].FORM#ALFA THEN ERROR(12);%
END ELSE ERROR(8);%
END ELSE ERROR(1);%
END ELSE ERROR(9);%
END ELSE BEGIN ERROR(89); SKIP(RPAR) END;%
END ELSE BEGIN ERROR(89); SKIP(RPAR) END;%
IF CURSY#RPAR THEN BEGIN ERROR(46); SKIP(RPAR) END;%
IF CURSY=RPAR THEN INSYMBOL;%
END ELSE BEGIN ERROR(3); INSYMBOL END;%
GEN(", ",1,7); GENINT(CARDCNT); GEN(", ",1,7);%
END OF PACK;%
%
%
PROCEDURE UNPACK;%

```

```

02049
02050
02051
02052
02053
02054
02055
02056
02057
02058
02059
02060
02061
02062
02063
02064
02065
02066
02067
02068
02069
02070
02071
02072
02073
02074
02075
02076
02077
02078
02079
02080
02081
02082
02083
02084
02085
02086
02087
02088
02089
02090
02091
02092
02093
02094
02095
02096
02097
02098
02099
02100
02101
02102
02103
02104
02105

```

BEGIN%

02106

INTEGER IT,T;%

02107

GEN("UNPACK(",7,1); INSYMBOL;%

02108

IF CURSY=LPAR THEN%

02109

BEGIN%

02110

INSYMBOL; EXPRESSION;%

02111

IF CURTYPE>0 THEN IF TYPETAB1[CURTYPE].FORM#ALFA THEN ERROR(17);%

02112

IF CURSY=COMMA THEN%

02113

BEGIN%

02114

GEN(",",1,7); INSYMBOL;%

02115

IF CURSY=IDENTIFIER THEN%

02116

BEGIN%

02117

SEARCH;%

02118

IF FOUND THEN%

02119

BEGIN%

02120

IF THISID.IDCLASS=VAR THEN%

02121

BEGIN%

02122

T:=TYPETAB1[THISID.TYPE];%

02123

IF T.FORM=ARRAYS THEN%

02124

BEGIN%

02125

IT:=T.INXTYPE;%

02126

IF TYPETAB1[T.ARRTYPE].FORM#CHAR THEN ERROR(88);%

02127

IF THISLEVEL>1 AND THISLEVEL#CURLEVEL THEN ERROR(5);%

02128

GENID("V",1000\*THISLEVEL+THISINDEX,5);%

02129

GEN(",",1,7); GENINT(TYPETAB2[THISID.TYPE]);%

02130

GEN(",",1,7); GENINT(TYPETAB3[THISID.TYPE]);%

02131

END ELSE ERROR(88);%

02132

END ELSE ERROR(88);%

02133

END ELSE ERROR(1);%

02134

END ELSE ERROR(9);%

02135

INSYMBOL;%

02136

IF CURSY=COMMA THEN%

02137

BEGIN%

02138

GEN(",",1,7);%

02139

INSYMBOL; EXPRESSION; CHECKTYPES(IT,CURTYPE);%

02140

END ELSE BEGIN ERROR(89); SKIP(RPAR) END;%

02141

END ELSE BEGIN ERROR(39); SKIP(RPAR) END;%

02142

IF CURSY#RPAR THEN BEGIN ERROR(89); SKIP(RPAR) END;%

02143

IF CURSY=RPAR THEN INSYMBOL;%

02144

END ELSE BEGIN ERROR(3); INSYMBOL END;%

02145

GEN(",",1,7); GENINT(CARDCNT); GEN(",",1,7);%

02146

END OF UNPACK;%

02147

%

02148

%

02149

PROCEDURE NEWDISP;

\*\*\* "NEW", "DISPOSE"

02150

BEGIN%

02151

INTEGER T1;%

02152

IF CURNAME1="300NEW" THEN GEN("NEW(",4,4) ELSE%

02153

BEGIN GEN("DISPOSE",7,1); GEN(",",1,7) END;%

02154

INSYMBOL;%

02155

IF CURSY=LPAR THEN%

02156

BEGIN%

02157

INSYMBOL;%

02158

IF CURSY=IDENTIFIER THEN%

02159

BEGIN%

02160

SEARCH;%

02161

IF FOUND THEN%

02162

14121  
Moore Business Forms, Inc. sv

```
BEGIN% 02163
VARIABLE;% 02164
IF CURTYPE>0 THEN IF TYPETAB1[CURTYPE].FORM=POINTERS THEN% 02165
  BEGIN% 02166
  WRITEEXPR; GEN(" ",1,7);% 02167
  T1:=TYPETAB1[CURTYPE].POINTTYPE;% 02168
  T1:=TYPETAB1[T1].SIZE;% 02169
  IF T1>1023 THEN ERROR(86);% 02170
  GENINT(T1); GEN(")",1,7);% 02171
  END ELSE ERROR(81);% 02172
  END ELSE BEGIN ERROR(1); INSYMBOL END;% 02173
END ELSE ERROR(9);% 02174
WHILE CURSY=COMMA DO% 02175
  BEGIN INSYMBOL;% 02176
  IF CURSY NEQ IDENTIFIER THEN ERROR(9);% 02177
  IF CURSY NEQ RPAR THEN INSYMBOL;% 02178
  END;% 02179
END ELSE BEGIN ERROR(58); SKIP(RPAR) END;% 02180
IF CURSY≠RPAR THEN BEGIN ERROR(46); SKIP(RPAR) END;% 02181
IF CURSY=RPAR THEN INSYMBOL;% 02182
END OF NEWDISP;% 02183
$ PAGE% 02184
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% 02185
% 02186
% 02187
% 02188
PART 6: THE STATEMENT PARSER. 02189
----- 02190
% 02191
% 02192
% 02193
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% 02194
% 02195
% 02196
% 02197
PROCEDURE STATEMENT; FORWARD;% 02198
% 02199
PROCEDURE ASSIGNMENT;% 02200
BEGIN% 02201
  INTEGER LEFTTYPE;% 02202
  LABEL ASSIGN,EXIT;% 02203
  IF FOUND THEN% 02204
  BEGIN% 02205
  IF THISID.IDCLASS=VAR OR% 02206
  THISID.IDCLASS=CONST AND BOOLEAN(THISID.FORMAL) THEN% 02207
  BEGIN% 02208
  VARIABLE; LEFTTYPE:=CURTYPE;% 02209
  ASSIGN; IF CURSY≠ASSIGNSY THEN% 02210
  BEGIN ERROR(28); SKIP(ASSIGNSY);% 02211
  IF SYMKIND[CURSY]=TERMINAL THEN GO TO EXIT;% 02212
  END;% 02213
  INSYMBOL;% 02214
  IF TYPETAB1[LEFTTYPE].STRUCT>0 THEN% 02215
  BEGIN% 02216
  ERROR(95);% 02217
  END ELSE% 02218
  BEGIN% 02219
```

```

WRITEEXPR; GEN("=",2,6);%
IF CHECKOPTION AND TYPETAB1[LEFTTYPE].FORMSCHAR THEN%
CHECKEXPR(TYPETAB2[LEFTTYPE],TYPETAB3[LEFTTYPE]) ELSE%
EXPRESSION;%
WRITEEXPR;%
CHECKTYPES(LEFTTYPE,CURTYPE);%
END;%
END ELSE%
BEGIN % FUNCTION ASSIGNMENT.
IF THISLEVEL<CURLEVEL-1 OR THISINDEX<CURFUNC THEN ERROR(5);%
GENID("V",1000*THISLEVEL+THISINDEX,5); LEFTTYPE:=THISID.TYPE;%
INSYMBOL; GO TO ASSIGN;%
END;%
END ELSE%
BEGIN%
SKIP(ASSIGNSY);%
IF CURSY=ASSIGNSY THEN GO TO ASSIGN;%
END;%
EXIT;%
END OF ASSIGNMENT;%
%
%
PROCEDURE COMPSTAT;%
BEGIN%
INTEGER BEGINNUM;%
LABEL STATM;%
%
BEGINNUM:=NUMBEGINS:=NUMBEGINS+1; MARGIN(" B",BEGINNUM);%
GEN("BEGIN",6,3);%
DO BEGIN%
IF CURSY=SEMICOLON OR CURSY=BEGINSY THEN INSYMBOL;%
STATM: STATEMENT;%
GEN(";",1,7);%
IF CURSY=ELSESY THEN BEGIN ERROR(20); INSYMBOL; GO STATM END;%
IF SYMKIND[CURSY]=INITIAL THEN BEGIN ERROR(21); GO STATM END;%
END UNTIL CURSY<SEMICOLON;%
IF CURSY<ENDSY THEN%
BEGIN ERROR(24); SKIP(ENDSY);%
IF CURSY<ENDSY THEN BEGIN INSYMBOL; GO TO STATM END;%
END;%
GEN(" END",5,4); MARGIN(" E",BEGINNUM);%
INSYMBOL;%
END OF COMPSTAT;%
%
%
PROCEDURE IFSTAT;%
BEGIN%
LABEL EXIT;%
GEN("IF",3,6);%
INSYMBOL; BOULEXPR;%
IF CURSY<THENSY THEN%
BEGIN IF CURTYPE>0 THEN ERROR(27);%
SKIP(THENSY);%
IF CURSY<THENSY THEN%
BEGIN IF CURTYPE=0 THEN ERROR(27);%
IF SYMKIND[CURSY]=TERMINAL THEN GO TO EXIT;%
END; END;%

```

```

02220
02221
02222
02223
02224
02225
02226
02227
02228
02229
02230
02231
02232
02233
02234
02235
02236
02237
02238
02239
02240
02241
02242
02243
02244
02245
02246
02247
02248
02249
02250
02251
02252
02253
02254
02255
02256
02257
02258
02259
02260
02261
02262
02263
02264
02265
02266
02267
02268
02269
02270
02271
02272
02273
02274
02275
02276

```



```

GEN(" THEN",6,3);%
INSYMBOL; STATEMENT;%
IF CURSY=ELSESY THEN%
BEGIN GEN(" ELSE",6,3); INSYMBOL; STATEMENT END;%
EXIT;%
END OF IFSTAT;%
%
%
PROCEDURE CASESTAT;%
BEGIN%
DEFINE CASEHASH(N)=(N),[38;39] MOD MAXCASES#;%
INTEGER ARRAY CASETABLO:MAXCASES#;%
INTEGER CASENUM,CASETYPE,NCASELABS,TEMPVARNUM,CONVAL,CONTYPE,C,T;%
BOOLEAN ZEROLAB,FIRST;%
%
CASENUM:=NUMCASES:=NUMCASES+1; MARGIN("CB",CASENUM);%
TEMPVARNUM:=NUMTEMPS:=NUMTEMPS+1;%
IF TEMPVARNUM>MAXTEMPS THEN ERROR(16);%
GEN("BEGIN",6,3); GENID("T",TEMPVARNUM,2); GEN("=",2,6);%
INSYMBOL; EXPRESSION;%
GEN(";",1,7); CASETYPE:=CURTYPE;%
IF TYPETAB1[CASETYPE].FORM≧FLOATING THEN%
BEGIN ERROR(17); CASETYPE:=0 END;%
IF CURSY≠OFSY THEN%
BEGIN IF CASETYPE>0 THEN ERROR(18);%
SKIP(OFSY);%
IF CURSY=OFSY THEN INSYMBOL ELSE%
IF CASETYPE=0 THEN ERROR(18);%
END ELSE INSYMBOL;%
DO BEGIN%
WHILE CURSY=SEMICOLON DO INSYMBOL;%
FIRST:=TRUE;%
IF CURSY≠ENDSY THEN%
BEGIN%
GEN("IF",3,6);%
DO BEGIN%
IF FIRST THEN FIRST:=FALSE ELSE INSYMBOL;%
CONSTANT(CONVAL,CONTYPE);%
IF CONTYPE>0 THEN%
BEGIN%
IF CASETYPE=0 THEN CASETYPE:=CONTYPE ELSE%
CHECKTYPES(CASETYPE,CONTYPE);%
GENID("T",TEMPVARNUM,2); GEN("=",1,7); GENINT(CONVAL);%
NCASELABS:=NCASELABS+1;%
IF NCASELABS<MAXCASES THEN%
BEGIN%
IF CONVAL=0 THEN%
IF ZEROLAB THEN ERROR(31) ELSE ZEROLAB:=TRUE ELSE%
BEGIN%
T:=CASEHASH(CONVAL);%
FOR C:=CASETAB[T] WHILE C≠CONVAL AND C≠0 DO%
T:=IF T=0 THEN MAXCASES ELSE T-1;%
IF C≠0 THEN ERROR(31) ELSE CASETAB[T]:=CONVAL;%
END;%
END ELSE IF NCASELABS=MAXCASES THEN ERROR(30);%
IF CURSY=COMMA THEN GEN(" OK",4,5);%
END;%

```

```

02277
02278
02279
02280
02281
02282
02283
02284
02285
02286
02287
02288
02289
02290
02291
02292
02293
02294
02295
02296
02297
02298
02299
02300
02301
02302
02303
02304
02305
02306
02307
02308
02309
02310
02311
02312
02313
02314
02315
02316
02317
02318
02319
02320
02321
02322
02323
02324
02325
02326
02327
02328
02329
02330
02331
02332
02333

```

```

END UNTIL CURSY≠COMMA;%
GEN(" THEN",6,3);%
IF CURSY≠COLON THEN BEGIN ERROR(26); SKIP(COLON) END;%
IF CURSY=COLON THEN INSYMBOL;%
STATEMENT;%
IF CURSY≠SEMICOLON AND CURSY≠ENDSY THEN%
BEGIN ERROR(21); SKIP(SEMICOLON) END;%
END;%
IF CURSY=SEMICOLON THEN GEN(" ELSE",6,3);%
END UNTIL CURSY≠SEMICOLON;%
IF CURSY≠ENDSY THEN BEGIN ERROR(24); SKIP(ENDSY) END;%
GEN(" END",5,4); MARGIN("CE",CASENUM);%
NUMTEMPS:=NUMTEMPS-1;%
INSYMBOL;%
END OF CASESTAT;%
%
%
PROCEDURE WHILESTAT;%
BEGIN%
LABEL-STATM,EXIT;%
GEN("WHILE",6,3);%
INSYMBOL; BOOLEXP;%
IF CURSY≠DOSY THEN%
BEGIN IF CURTYPE>0 THEN ERROR(19);%
SKIP(DOSY);%
IF CURSY≠DOSY THEN%
BEGIN IF CURTYPE=0 THEN ERROR(19);%
GO TO IF SYMKIND[CURSY]=INITIAL THEN STATM ELSE EXIT;%
END; END;%
GEN(" DO",4,5);%
INSYMBOL;%
STATM: STATEMENT;%
EXIT;%
END OF WHILESTAT;%
%
%
PROCEDURE REPEATSTAT;%
BEGIN%
INTEGER REPNUM;%
LABEL NEWTRY;%
%
REPNUM:=NUMREPS:=NUMREPS+1;%
MARGIN(" R",REPNUM);%
GEN("DO",3,6); GEN("BEGIN",6,3);%
DO BEGIN%
INSYMBOL;%
NEWTRY: STATEMENT;%
GEN(";",1,7);%
IF CURSY=ELSESY THEN BEGIN ERROR(20);INSYMBOL; GO NEWTRY END;%
IF SYMKIND[CURSY]=INITIAL THEN BEGIN ERROR(21); GO NEWTRY END;%
END UNTIL CURSY≠SEMICOLON;%
IF CURSY≠UNTILSY THEN%
BEGIN%
ERROR(22);%
WHILE CURSY≠UNTILSY AND SYMKIND[CURSY]≠INITIAL DO%
BEGIN-INSYMBOL; SKIP(UNTILSY). END;%
IF CURSY≠UNTILSY THEN GO TO NEWTRY;%

```

```

02334
02335
02336
02337
02338
02339
02340
02341
02342
02343
02344
02345
02346
02347
02348
02349
02350
02351
02352
02353
02354
02355
02356
02357
02358
02359
02360
02361
02362
02363
02364
02365
02366
02367
02368
02369
02370
02371
02372
02373
02374
02375
02376
02377
02378
02379
02380
02381
02382
02383
02384
02385
02386
02387
02388
02389
02390

```

```
END;% 02391
GEN("END",5,4); GEN("UNTIL",6,3); MARGIN(" U",REPNUM);% 02392
INSYMBOL; BOOLEXP; % 02393
END OF REPEATSTAT;% 02394
% 02395
% 02396
PROCEDURE FURSTAT;% 02397
BEGIN% 02398
INTEGER VARTYPE,VARNUM,LLIM,ULIM;% 02399
BOOLEAN DOWN;% 02400
LABEL STATM;% 02401
% 02402
GEN("BEGIN",6,3);% 02403
INSYMBOL;% 02404
IF CURSY=JIDENTIFIER THEN% 02405
BEGIN% 02406
SEARCH;% 02407
IF FOUND THEN% 02408
BEGIN% 02409
VARNUM:=1000*THISLEVEL+THISINDEX;% 02410
IF THISID.IDCLASS=VAR OR% 02411
THISID.IDCLASS=CONST AND BOOLEAN(THISID.FORMAL) THEN% 02412
BEGIN% 02413
IF THISLEVEL>1 AND THISLEVEL<CURLEVEL THEN ERROR(5);% 02414
IF THISLEVEL>CURLEVEL THEN ERROR(83);% 02415
VARTYPE:=THISID.TYPE;% 02416
IF TYPETAB1[VARTYPE].FORMSCHAR THEN% 02417
BEGIN% 02418
LLIM:=TYPETAB2[VARTYPE]; ULIM:=TYPETAB3[VARTYPE];% 02419
END ELSE BEGIN ERROR(12); VARTYPE:=0 END;% 02420
END ELSE ERROR(8);% 02421
END ELSE ERROR(1);% 02422
END ELSE ERROR(9);% 02423
INSYMBOL;% 02424
IF CURSY≠ASSIGNSY THEN% 02425
BEGIN ERROR(28);% 02426
SKIP(ASSIGNSY);% 02427
IF CURSY=ASSIGNSY THEN INSYMBOL ELSE% 02428
IF SYMKIND[CURSY]=INITIAL THEN GO TO STATM;% 02429
END ELSE INSYMBOL;% 02430
GENID("V",VARNUM,5); GEN("<",1,7);% 02431
IF CHECKOPTION THEN CHECKEXPR(LLIM,ULIM) ELSE EXPRESSION;% 02432
WRITEEXPR;% 02433
GEN(";",1,7);% 02434
IF VARTYPE=0 THEN VARTYPE:=CURTYPE ELSE CHECKTYPES(VARTYPE,CURTYPE); 02435
NUMTEMPS:=NUMTEMPS+1; IF NUMTEMPS>MAXTEMPS THEN ERROR(16);% 02436
IF CURSY=TOSY THEN INSYMBOL ELSE% 02437
IF CURSY=DOWNTOSY THEN BEGIN DOWN:=TRUE; INSYMBOL END ELSE% 02438
BEGIN IF CURTYPE>0 THEN ERROR(23);% 02439
SKIP(TOSY);% 02440
IF CURSY=TOSY THEN INSYMBOL ELSE% 02441
BEGIN IF CURTYPE=0 THEN ERROR(23);% 02442
IF SYMKIND[CURSY]=INITIAL THEN GO TO STATM;% 02443
END; END;% 02444
GENID("T",NUMTEMPS,2); GEN("<",1,7);% 02445
IF CHECKOPTION THEN CHECKEXPR(LLIM,ULIM) ELSE EXPRESSION;% 02446
WRITEEXPR;% 02447
```

```

GENC(";",1,7);% 02448
IF VARTYPE=0 THEN VARTYPE:=CURTYPE ELSE CHECKTYPES(VARTYPE,CURTYPE); 02449
IF CURSY≠DOSY THEN% 02450
BEGIN IF CURTYPE>0 THEN ERROR(19);% 02451
SKIP(DOSY);% 02452
IF CURSY=DOSY THEN INSYMBOL ELSE% 02453
IF CURTYPE=0 THEN ERROR(19);% 02454
END-ELSE-INSYMBOL;% 02455
GENC("FOR",4,5); GENID("V",VARNUM,5); GENC("←",1,7);% 02456
GENID("V",VARNUM,5); GENC(" ",1,7);% 02457
IF DOWN THEN GENC("DOWNT0",7,2) ELSE GENC("UPT0",5,4);% 02458
GENID("T",NUMTEMPS,2); GENC(" DU",4,5);% 02459
STATM: STATEMENT;% 02460
GENC(" END",5,4);% 02461
NUMTEMPS:=NUMTEMPS-1;% 02462
END OF FORSTAT;% 02463
% 02464
% 02465
PROCEDURE GOTOSTAT;% 02466
BEGIN% 02467
INTEGER I;% 02468
INSYMBOL;% 02469
IF CURSY=INTCONST THEN% 02470
BEGIN I:=NUMLABS;% 02471
WHILE I≥1 AND LABTAB[I].LABVAL≠CURVAL DO I:=I-1;% 02472
IF I=0 THEN ERROR(15);% 02473
GENC("GO",3,6); GENID("L",CURVAL,4);% 02474
INSYMBOL;% 02475
END ELSE ERROR(10);% 02476
END OF GOTOSTAT;% 02477
% 02478
% 02479
PROCEDURE WITHSTAT;% 02480
BEGIN% 02481
INTEGER STARTLEVEL,VERYFIRSTWITHSYM,I;% 02482
REAL D;% 02483
STARTLEVEL:=TOPLEVEL; VERYFIRSTWITHSYM:=NWITHSYMS;% 02484
DO-BEGIN% 02485
INSYMBOL;% 02486
IF CURSY=IDENTIFIER THEN% 02487
BEGIN% 02488
SEARCH;% 02489
IF FOUND THEN% 02490
BEGIN% 02491
IF THISID.IDCLASS=VAR THEN% 02492
BEGIN% 02493
VARIABLE;% 02494
IF CURTYPE>0 THEN% 02495
IF TYPETAB1[CURTYPE].FORM≠RECORD THEN ERROR(98);% 02496
IF SIMPLEVARIABLE THEN% 02497
BEGIN PUTSYM("["); INSIDEBRACKETS:=TRUE END;% 02498
IF TOPLEVEL<MAXLEVEL THEN% 02499
BEGIN% 02500
TOPLEVEL:=TOPLEVEL+1;% 02501
D.NAMETAB:=TYPETAB1[CURTYPE].RECTAB;% 02502
D.RECTYPE:=CURTYPE;% 02503
D.NUMPNTRSINWITH:=NUMPOINTERS;% 02504

```

```

D.FIRSTWITHSYM:=NWITHSYMS;%
D.BRACKETSINWITH:=REAL(INSIDEBRACKETS);%
IF NWITHSYMS+NUMSYMS>MAXWITHSYMS THEN ERROR(63) ELSE%
FOR I:=1 STEP 1 UNTIL NUMSYMS DO%
BEGIN%
WITHTAB[NWITHSYMS]:=SYMTAB[I];%
NWITHSYMS:=NWITHSYMS+1;%
END;%
D.LASTWITHSYM:=NWITHSYMS-1;%
DISPLAY[TOPLEVEL]:=D;%
END-ELSE-ERROR(84);%
END ELSE BEGIN ERROR(8); INSYMBOL END;%
END ELSE BEGIN ERROR(1); INSYMBOL END;%
END ELSE BEGIN ERROR(9); INSYMBOL END;%
NUMSYMS:=0;%
NUMPOINTERS := 0;%
END UNTIL CURSY#COMMA;%
IF CURSY#DOSY THEN%
BEGIN ERROR(19); SKIP(DOSY);%
IF CURSY=DOSY THEN INSYMBOL;%
END ELSE INSYMBOL;%
STATEMENT;%
TOPLEVEL:=STARTLEVEL; NWITHSYMS:=VERYFIRSTWITHSYM;%
END OF WITHSTAT;%
%
%
PROCEDURE STATEMENT;%
BEGIN%
INTEGER I;%
LABEL LABFOUND;%
%
IF CURSY=INTCONST THEN % *** LABELED STATEMENT ***
BEGIN%
FOR I:=FIRSTLAB STEP 1 UNTIL NUMLABS DO%
IF LABTAB[I].LABVAL=CURVAL THEN%
BEGIN IF LABTAB[I].LABDEF=1 THEN ERROR(31);%
LABTAB[I].LABDEF:=1;%
GO TO LABFOUND;%
END;%
ERROR(15);%
LABFOUND:=GENID("L",CURVAL,4); GEN(":",1,7);%
INSYMBOL;%
IF CURSY#COLON THEN%
BEGIN-ERROR(26);%
SKIP(COLON); IF CURSY=COLON THEN INSYMBOL;%
END ELSE INSYMBOL;%
END;%
%
COMMENT *** START OF STATEMENT *** ;%
%
IF CURSY=IDENTIFIER THEN%
BEGIN%
SEARCH;%
IF FOUND THEN%
BEGIN%
IF THISID.IDCLASS=VAR OR%
THISID.IDCLASS=CONST AND BOOLEAN(THISID.FORMAL) OR%

```

```

02505
02506
02507
02508
02509
02510
02511
02512
02513
02514
02515
02516
02517
02518
02519
02520
02521
02522
02523
02524
02525
02526
02527
02528
02529
02530
02531
02532
02533
02534
02535
02536
02537
02538
02539
02540
02541
02542
02543
02544
02545
02546
02547
02548
02549
02550
02551
02552
02553
02554
02555
02556
02557
02558
02559
02560
02561

```

```

THISID.IDCLASS=FUNC THEN ASSIGNMENT ELSE% 02562
IF THISID.IDCLASS=PROC THEN% 02563
BEGIN% 02564
  IF THISLEVEL=0 THEN % *** INTRINSIC PROCEDURE *** 02565
  BEGIN% 02566
    IF CURNAME1="50WRITE" THEN PWRITE(FALSE) ELSE% 02567
    IF CURNAME1="7WRITEL" AND% 02568
    CURNAME2="000000N" THEN PWRITE(TRUE) ELSE% 02569
    IF CURNAME1="400READ" THEN PREAD(FALSE) ELSE% 02570
    IF CURNAME1="6READLN" THEN PREAD(TRUE) ELSE% 02571
    IF CURNAME1="400PAGE" THEN FILEHANDLING(5) ELSE% 02572
    IF CURNAME1="3000GET" THEN FILEHANDLING(2) ELSE% 02573
    IF CURNAME1="3000PUT" THEN FILEHANDLING(1) ELSE% 02574
    IF CURNAME1="50RESET" THEN FILEHANDLING(3) ELSE% 02575
    IF CURNAME1="7REWRI" AND% 02576
    CURNAME2="000000E" THEN FILEHANDLING(4) ELSE% 02577
    IF CURNAME1="3000NEW" THEN NEWDISP ELSE% 02578
    IF CURNAME1="7DISPOS" AND% 02579
    CURNAME2="000000E" THEN NEWDISP ELSE% 02580
    IF CURNAME1="400PACK" THEN PACK ELSE% 02581
    IF CURNAME1="6UNPACK" THEN UNPACK ELSE ERROR(0);% 02582
  END ELSE PASSPARAMS;% 02583
  WRITEEXPR;% 02584
  END ELSE BEGIN ERROR(13); SKIP(99) END;% 02585
  END ELSE BEGIN ERROR(1); ASSIGNMENT END;% 02586
END-OF-IDENTIFIER-ELSE% 02587
IF CURSY=BEGINSY THEN COMPSTAT ELSE% 02588
IF CURSY=IFSY THEN IFSTAT ELSE% 02589
IF CURSY=CASESY THEN CASESTAT ELSE% 02590
IF CURSY=WHILESY THEN WHILESTAT ELSE% 02591
IF CURSY=REPEATSY THEN REPEATSTAT ELSE% 02592
IF CURSY=FORSY THEN FORSTAT ELSE% 02593
IF CURSY=WITHSY THEN WITHSTAT ELSE% 02594
IF CURSY=GOTOSY THEN GOTOSTAT ELSE% 02595
IF SYMKINDICURSYP≠TERMINAL THEN% 02596
  BEGIN ERROR(13); INSYMBOL; SKIP(SEMICOLON) END;% 02597
END OF STATEMENT;% 02598
$ PAGE% 02599
%%%%%%%%%% 02600
% 02601
% 02602
% 02603
PART 7: TYPE DECLARATIONS. 02604
% 02605
% 02606
% 02607
% 02608
%%%%%%%%%% 02609
% 02610
% 02611
REAL VALX1,VALX2;% 02612
INTEGER TYPEX1,TYPEX2;% 02613
BOOLEAN PACKED;% 02614
% 02615
PROCEDURE FIELDLIST(RECTAB,FIRSTADDR,LASTADDR);% 02616
VALUE RECTAB,FIRSTADDR;% 02617
INTEGER RECTAB,FIRSTADDR,LASTADDR;% 02618

```

```

FORWARD;%
%
DEFINE SUBRANGE=          %*** SUBRANGE DECLARATION ***
BEGIN                    %*****
CONSTANT(VALX1,TYPEX1);%
IF TYPETAB1[TYPEX1].FORM>CHAR THEN ERROR(11);%
IF CURSY#DOUBLEDOT THEN ERROR(53);%
INSYMBOL;%
CONSTANT(VALX2,TYPEX2);%
IF TYPEX1>0 AND TYPEX2>0 THEN%
IF TYPEX1#TYPEX2 THEN ERROR(11) ELSE%
IF VALX1>VALX2 THEN ERROR(54);%
T1:=TYPETAB1[TYPEX1].FORM; IF T1=SYMBOLIC THEN T1:=SUBTYPE;%
NEWTYPE; TTYPE:=TYPEINDEX;%
T1.SIZE:=TSIZE:=1; T1.STRUCT:=0; T1.MAINTYPE:=TYPEX1;%
TYPETAB1[TYPEINDEX]:=T1;%
TYPETAB2[TYPEINDEX]:=VALX1; TYPETAB3[TYPEINDEX]:=VALX2;%
END OF SUBRANGE#;%
%
%
PROCEDURE TYPEDECL(TTYPE,TSIZE);%
INTEGER TTYPE,TSIZE;%
BEGIN%
PROCEDURE TYPERR(ERRNUM,TTYPE,TSIZE);%
VALUE ERRNUM;%
INTEGER ERRNUM,TTYPE,TSIZE;%
BEGIN ERROR(ERRNUM);%
TTYPE:=TSIZE:=0;%
END;%
%
INTEGER RECINX,ARRSTRUCT,FX,SX,T1,T2,T3,T,N;%
BOOLEAN FIRST;%
%
PACKED:=FALSE;%
IF CURSY#IDENTIFIER THEN %*** SIMPLE TYPE DECLARATION ***
BEGIN %*****
SEARCH;%
IF FOUND THEN%
BEGIN%
IF THISID.IDCLASS=TYPES THEN%
BEGIN%
TTYPE:=THISID.TYPE; TSIZE:=TYPETAB1[TTYPE].SIZE;%
INSYMBOL;%
END ELSE IF THISID.IDCLASS=CONST THEN SUBRANGE%
ELSE TYPERR(7,TTYPE,TSIZE);%
END ELSE BEGIN TYPERR(1,TTYPE,TSIZE); INSYMBOL END;%
END ELSE%
IF CURSY#CHARCONST OR CURSY=PLUS OR CURSY=MINUS THEN SUBRANGE ELSE%
IF CURSY=LPAR THEN%
BEGIN%
N:=0;%
NEWTYPE; T3.IDCLASS:=CONST; T3.TYPE:=TYPEINDEX;%
DO-BEGIN%
INSYMBOL;%
IF CURSY=IDENTIFIER THEN%
BEGIN%
NEWNAME(CURNAME1,CURNAME2,CURLEVEL);%

```

```

02619
02620
02621
02622
02623
02624
02625
02626
02627
02628
02629
02630
02631
02632
02633
02634
02635
02636
02637
02638
02639
02640
02641
02642
02643
02644
02645
02646
02647
02648
02649
02650
02651
02652
02653
02654
02655
02656
02657
02658
02659
02660
02661
02662
02663
02664
02665
02666
02667
02668
02669
02670
02671
02672
02673
02674
02675

```

```

T3.INFO:=N; NAMETAB3[CURLEVEL,THISINDEX]:=T3;% 02676
N:=N+1;--INSYMBOL;% 02677
END ELSE ERROR(9);% 02678
END UNTIL CURSY#COMMA;% 02679
IF CURSY#RPAR THEN BEGIN ERROR(46); SKIP(RPAR) END;% 02680
T1:=SYMBOLIC; T1.STRUCT:=0;% 02681
T1.SIZE:=TSIZE:=1; TTYPE:=TYPEINDEX;% 02682
TYPETAB1[TYPEINDEX]:=T1;% 02683
TYPETAB2[TYPEINDEX]:=0; TYPETAB3[TYPEINDEX]:=N-1;% 02684
IF CURSY=RPAR THEN INSYMBOL;% 02685
END ELSE% 02686
% 02687
IF CURSY=ARROW THEN %*** POINTER DECLARATION *** 02688
BEGIN %***** 02689
INSYMBOL;% 02690
IF CURSY=IDENTIFIER THEN% 02691
BEGIN% 02692
NEWTYPE; TTYPE:=TYPEINDEX; T1:=POINTERS;% 02693
T1.SIZE:=TSIZE:=1; T1.STRUCT:=0;% 02694
TYPETAB1[TYPEINDEX]:=T1;% 02695
SEARCH;% 02696
IF FOUND THEN% 02697
BEGIN% 02698
IF THISID.IDCLASS=TYPES THEN% 02699
TYPETAB1[TYPEINDEX].POINTTYPE:=THISID.TYPE ELSE% 02700
TYPERR(7,TTYPE,TSIZE);% 02701
END ELSE% 02702
BEGIN% 02703
IF NUMPNTRS<MAXPNTRS THEN NUMPNTRS:=NUMPNTRS+1 ELSE ERROR(52); 02704
PNTRTAB1[NUMPNTRS]:=CURNAME1; PNTRTAB2[NUMPNTRS]:=CURNAME2;% 02705
PNTRTAB3[NUMPNTRS]:=TYPEINDEX;% 02706
END;% 02707
INSYMBOL;% 02708
END ELSE TYPERR(9,TTYPE,TSIZE);% 02709
END OF POINTER DECLARATION ELSE% 02710
BEGIN% 02711
IF CURSY=PACKEDSY THEN BEGIN PACKED:=TRUE; INSYMBOL END;% 02712
% 02713
IF CURSY=ARRAYSY THEN %*** ARRAY DECLARATION *** 02714
BEGIN %***** 02715
INSYMBOL;% 02716
IF CURSY#LBRACKET THEN ERROR(47) ELSE INSYMBOL;% 02717
T:=0; FIRST:=TRUE;% 02718
DO BEGIN% 02719
IF FIRST THEN FIRST:=FALSE ELSE INSYMBOL;% 02720
TYPEDECL(TX,SX);% 02721
IF TX>0 THEN% 02722
BEGIN% 02723
IF TYPETAB1[TX].FORM>CHAR THEN ERROR(48);% 02724
T1:=ARRAYS; T1.INXTYPE:=TX; T1.ARRTYPE:=T;% 02725
T2:=TYPETAB2[TX]; T3:=TYPETAB3[TX];% 02726
IF T3-T2>1022 THEN ERROR(61);% 02727
T1.SIZE:=MIN(1023,T3-T2+1);% 02728
NEWTYPE;% 02729
TYPETAB1[TYPEINDEX]:=T1;% 02730
TYPETAB2[TYPEINDEX]:=T2; TYPETAB3[TYPEINDEX]:=T3;% 02731
T:=TYPEINDEX;% 02732

```

14121 Moore Business Forms, Inc. 3v



```

END;%
END UNTIL CURSY≠COMMA;%
IF CURSY≠RBRACKET THEN ERROR(59) ELSE INSYMBOL;%
IF CURSY≠OFSY THEN BEGIN ERROR(18); SKIP(OFSY) END;%
INSYMBOL;%
TYPEDECL(TX,SX);%
IF TYPETAB1[TX].FORM≥FILES THEN ERROR(60);%
ARRSTRUCT:=TYPETAB1[TX].STRUCT;%
WHILE T>0 DO%
BEGIN%
T1:=TYPETAB1[T]; T3:=T1.ARRTYPE;%
T1.ARRTYPE:=TX; T1.STRUCT:=ARRSTRUCT:=ARRSTRUCT+1;%
T1.SIZE:=SX:=MIN(1024,SX×T1.SIZE);%
TYPETAB1[T]:=T1; TX:=T; T:=T3;%
END;%
TTYPE:=TX; TSIZE:=SX;%
END OF ARRAY DECLARATION ELSE%
%
IF CURSY=FILESY THEN %*** FILE DECLARATION ***
BEGIN %*****
INSYMBOL;%
IF CURSY≠OFSY THEN%
BEGIN ERROR(18);%
IF CURSY≠IDENTIFIER THEN INSYMBOL;%
END ELSE INSYMBOL;%
TYPEDECL(TX,SX);%
IF TX>0 THEN%
BEGIN T:=TYPETAB1[TX];%
IF T.FORM≥FILES THEN ERROR(50) ELSE%
IF T.STRUCT>1 THEN ERROR(49);%
END;%
NEWTYPE; TTYPE:=TYPEINDEX;%
T1:=IF T.FORM=CHAR THEN TEXTFILE ELSE FILES;%
T1.SIZE:=TSIZE:=SX; T1.FILETYPE:=TX;%
T1.STRUCT:=1;%
TYPETAB1[TYPEINDEX]:=T1;%
END OF FILE DECLARATION ELSE%
%
IF CURSY=SETSY THEN %*** SET DECLARATION ***
BEGIN %*****
INSYMBOL;%
IF CURSY≠OFSY THEN%
BEGIN ERROR(18);%
IF CURSY>CHARCONST THEN INSYMBOL;%
END ELSE INSYMBOL;%
TYPEDECL(TX,SX);%
IF TX>0 THEN%
BEGIN%
IF TYPETAB1[TX].FORM>CHAR THEN ERROR(48) ELSE%
IF TYPETAB2[TX]<0 OR TYPETAB3[TX]>38 THEN ERROR(51);%
END;%
NEWTYPE; TTYPE:=TYPEINDEX;%
T1:=SET; T1.SETTYPE:=TX; T1.STRUCT:=0;%
T1.SIZE:=TSIZE:=1; TYPETAB1[TYPEINDEX]:=T1;%
TYPETAB2[TYPEINDEX]:=TYPETAB2[TX];%
TYPETAB3[TYPEINDEX]:=TYPETAB3[TX];%
END OF SET DECLARATION ELSE%

```

```

02733
02734
02735
02736
02737
02738
02739
02740
02741
02742
02743
02744
02745
02746
02747
02748
02749
02750
02751
02752
02753
02754
02755
02756
02757
02758
02759
02760
02761
02762
02763
02764
02765
02766
02767
02768
02769
02770
02771
02772
02773
02774
02775
02776
02777
02778
02779
02780
02781
02782
02783
02784
02785
02786
02787
02788
02789

```

Moore Business Forms, Inc. sv 1-1121

```

%
IF CURSY=RECORDSY THEN      %*** RECURD DECLARATION ***
BEGIN                        %*****
  IF LASTREC-1>CURLEVEL THEN LASTREC:=LASTREC-1 ELSE ERROR(55);%
  RECINX:=LASTREC;%
  BLOCKTAB[RECINX]:=NUMBLOCKS:=NUMBLOCKS+1;%
  INSYMBOL;%
  FIELDLIST(RECINX,0,SX);%
  IF SX>1022 THEN BEGIN ERROR(56); SX:=1022 END;%
  NEWTYPE; TTYPE:=TYPEINDEX;%
  T1:=RECORD; T1.RECTAB:=RECINX; T1.STRUCT:=1;%
  T1.SIZE:=TSIZE:=SX; TYPETAB1[TYPEINDEX]:=T1;%
  TYPETAB2[TYPEINDEX]:=0; TYPETAB3[TYPEINDEX]:=SX-1;%
  IF CURSY<ENDSY THEN BEGIN ERROR(24); SKIP(ENDSY) END;%
  IF CURSY=ENDSY THEN INSYMBOL;%
  END ELSE BEGIN ERROR(4); SKIP(99) END;%
END;%
END OF TYPEDECL;%
%
%
PROCEDURE FIELDLIST(RECTAB,FIRSTADDR,LASTADDR);%
VALUE RECTAB,FIRSTADDR;%
INTEGER RECTAB,FIRSTADDR,LASTADDR;%
BEGIN%
  INTEGER ARRAY ILIST[0:LISTLENGTH];%
  INTEGER LISTINX;%
  INTEGER CASETYPE,ADDR,MAXADDR,INDEX,CTYPE,TX,SX,T1,T3,LLIM,ULIM,I;%
  BOOLEAN FIRST;%
  REAL CVAL;%
  LABEL CASETYPEID,CASEPART,EXIT;%
%
  ADDR:=FIRSTADDR;%
  DO BEGIN%
    WHILE CURSY=SEMICOLON DO INSYMBOL;%
    IF CURSY=CASESY THEN GO TO CASEPART;%
    IF CURSY=IDENTIFIER THEN%
      BEGIN%
        LISTINX:=0; FIRST:=TRUE;%
        DO BEGIN%
          IF FIRST THEN FIRST:=FALSE ELSE INSYMBOL;%
          IF CURSY=IDENTIFIER THEN%
            BEGIN%
              IF LISTINX>LISTLENGTH THEN BEGIN ERROR(37); LISTINX:=0 END;%
              LISTINX:=LISTINX+1;%
              NEWNAME(CURNAME1,CURNAME2,RECTAB);%
              ILIST[LISTINX]:=THISINDEX;%
              INSYMBOL;%
            END ELSE%
              BEGIN ERROR(9);%
            IF CURSY<COMMA THEN INSYMBOL;%
          END;%
        END UNTIL CURSY<COMMA;%
        IF CURSY<COLON THEN BEGIN ERROR(26); SKIP(COLON) END;%
        INSYMBOL;%
        TYPEDECL(TX,SX);%
        IF TX>0 THEN IF TYPETAB1[TX].FORM>FILES THEN ERROR(57);%
        T3.IDCLASS:=VAR; T3.TYPE:=IX;%
      END;

```

```

02790
02791
02792
02793
02794
02795
02796
02797
02798
02799
02800
02801
02802
02803
02804
02805
02806
02807
02808
02809
02810
02811
02812
02813
02814
02815
02816
02817
02818
02819
02820
02821
02822
02823
02824
02825
02826
02827
02828
02829
02830
02831
02832
02833
02834
02835
02836
02837
02838
02839
02840
02841
02842
02843
02844
02845
02846

```

```

FOR I:=1 STEP 1 UNTIL LISTINX DO%
BEGIN%
T3.INFO:=ADDR; ADDR:=MIN(ADDR+SX,1024);%
NAMETAB3[RECTAB,ILIST[I]]:=T3;%
END;%
END;%
END UNTIL CURSY#SEMICOLON;%
LASTADDR:=ADDR;%
GO TO EXIT;%

%
CASEPART:%
LISTINX:=0; LASTADDR:=ADDR; INDEX:=-1;%
INSYMBOL;%
IF CURSY=IDENTIFIER THEN%
BEGIN%
SEARCH;%
IF FOUND AND THISID.IDCLASS=TYPES THEN GO TO CASETYPEID;%
NEWNAME(CURNAME1,CURNAME2,RECTAB); INDEX:=THISINDEX;%
INSYMBOL;%
IF CURSY#COLON THEN ERROR(26);%
INSYMBOL;%
IF CURSY=IDENTIFIER THEN%
BEGIN%
SEARCH;%
IF FOUND THEN%
BEGIN%
IF THISID.IDCLASS=TYPES THEN%
BEGIN%
CASETYPEID: CASETYPE:=THISID.TYPE; T1:=TYPETAB1[CASETYPE];%
LLIM:=TYPETAB2[CASETYPE]; ULIM:=TYPETAB3[CASETYPE];%
IF T1.FORM>CHAR THEN ERRUR(48);%
IF INDEX>0 THEN%
BEGIN%
T3.IDCLASS:=VAR; T3.TYPE:=CASETYPE; T3.INFO:=ADDR;%
ADDR:=LASTADDR:=ADDR+1; NAMETAB3[RECTAB,INDEX]:=T3;%
END;%
INSYMBOL;%
END ELSE BEGIN ERROR(7); SKIP(OFSY) END;%
END ELSE BEGIN ERROR(1); SKIP(OFSY) END;%
END ELSE BEGIN ERROR(9); SKIP(OFSY) END;%
END ELSE BEGIN ERROR(9); SKIP(OFSY) END;%
IF CURSY#OFSY THEN BEGIN ERROR(18); SKIP(RPAR) END;%
IF CURSY=OFSY THEN INSYMBOL;%
IF CASETYPE=0 THEN BEGIN LLIM:=MAXINT; ULIM:=MAXINT END;%
DO BEGIN%
WHILE CURSY=SEMICOLON DO INSYMBOL;%
IF CURSY<CHARCONST OR CURSY=PLUS OR CURSY=MINUS THEN%
BEGIN%
FIRST:=TRUE;%
DO BEGIN%
IF FIRST THEN FIRST:=FALSE ELSE INSYMBOL;%
CONSTANT(CVAL,CTYPE);%
IF CTYPE>0 THEN%
BEGIN%
IF CASETYPE=0 THEN CASETYPE:=CTYPE ELSE%
IF CVAL<LLIM OR CVAL>ULIM THEN ERROR(14) ELSE%
CHECKTYPES(CASETYPE,CTYPE);%

```

```

02847
02848
02849
02850
02851
02852
02853
02854
02855
02856
02857
02858
02859
02860
02861
02862
02863
02864
02865
02866
02867
02868
02869
02870
02871
02872
02873
02874
02875
02876
02877
02878
02879
02880
02881
02882
02883
02884
02885
02886
02887
02888
02889
02890
02891
02892
02893
02894
02895
02896
02897
02898
02899
02900
02901
02902
02903

```

```

IF LISTINX>=LISTLENGTH THEN BEGIN ERRUR(30); LISTINX:=0 END;% 02904
LISTINX:=LISTINX+1;% 02905
ILIST[LISTINX]:=CVAL; I:=1;% 02906
WHILE ILIST[I]≠CVAL DO I:=I+1;% 02907
IF I<LISTINX THEN ERROR(31);% 02908
END;% 02909
END UNTIL CURSY≠COMMA;% 02910
IF CURSY≠COLON THEN BEGIN ERROR(26); SKIP(LPAR) END;% 02911
IF CURSY=COLON THEN INSYMBOL;% 02912
IF CURSY=LPAR THEN% 02913
BEGIN% 02914
INSYMBOL; FIELDLIST(RECTAB,ADDR,MAXADDR);% 02915
IF MAXADDR>LASTADDR THEN LASTADDR:=MAXADDR;% 02916
IF CURSY≠RPAR THEN BEGIN ERROR(46); SKIP(RPAR) END;% 02917
INSYMBOL;% 02918
END ELSE ERROR(58);% 02919
END;% 02920
END UNTIL CURSY NEQ SEMICOLON; % 02921
EXIT;% 02922
END-OF-FIELDLIST;% 02923
$ PAGE% 02924
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% 02925
% 02926
% 02927
% 02928
PART 8:-----THE PROCEDURE BLOCK. 02929
----- 02930
% 02931
% 02932
% 02933
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% 02934
% 02935
% 02936
% 02937
PROCEDURE DECLAREVARS(PARAM,TAB,FIRST,LAST,LEVEL);% 02938
VALUE PARAM,FIRST,LAST,LEVEL;% 02939
INTEGER ARRAY TAB[0];% 02940
INTEGER FIRST,LAST,LEVEL;% 02941
BOOLEAN PARAM;% 02942
BEGIN% 02943
INTEGER LEVEL1000,TYP,NAM,NAMTAB,T1,I,J,RECSIZE;% 02944
BOOLEAN REALVAR,ARRAYVAR,FIRSTDIM,EXTFILE;% 02945
ALPHA FNAME;% 02946
INTEGER FNLENGTH,FNSTART; % 02947
% 02948
LEVEL1000:=LEVEL*1000;% 02949
FOR I:=FIRST-STEP 1 UNTIL LAST DO% 02950
BEGIN% 02951
NAM:=TAB[I].[9:10]; NAMTAB:=NAMETAB3[LEVEL,NAM];% 02952
TYP:=NAMTAB.TYPE; T1:=TYPETAB1[TYP];% 02953
IF NAMTAB.IDCLASS GEQ FUNC THEN% 02954
BEGIN% 02955
IF REALVAR OR ARRAYVAR THEN% 02956
BEGIN% 02957
GEN(";",1,7);% 02958
REALVAR:=ARRAYVAR:=FALSE;% 02959
END;% 02960

```

IF NAMTAB.IDCLASS=FUNC THEN GEN("REAL",5,4);%	02961
GEN("PROCEDU",8,1);%	02962
GENID("V",LEVEL1000+NAM,5); GEN(";",1,7);%	02963
END ELSE%	02964
IF T1.STRUCT=0 THEN %*** SIMPLE TYPE ***	02965
BEGIN%	02966
IF ARRAYVAR THEN BEGIN GEN(";",1,7); ARRAYVAR:=FALSE END;%	02967
IF REALVAR THEN GEN(";",1,7) ELSE%	02968
BEGIN GEN("REAL",5,4); REALVAR:=TRUE END;%	02969
GENID("V",LEVEL1000+NAM,5);%	02970
END ELSE%	02971
BEGIN%	02972
IF REALVAR THEN BEGIN GEN(";",1,7); REALVAR:=FALSE END;%	02973
IF T1.FORM<FILES THEN %*** ARRAY/RECORD ***	02974
BEGIN%	02975
IF ARRAYVAR THEN GEN(";",1,7) ELSE%	02976
BEGIN GEN("ARRAY",6,3); ARRAYVAR:=TRUE END;%	02977
GENID("V",LEVEL1000+NAM,5); GEN("[",1,7);%	02978
FIRSTDIM:=TRUE;%	02979
DO BEGIN%	02980
IF FIRSTDIM THEN FIRSTDIM:=FALSE ELSE GEN(";",1,7);%	02981
GENINT(TYPETAB2[TYP]);%	02982
IF NOT PARAM THEN%	02983
BEGIN GEN(";",1,7); GENINT(TYPETAB3[TYP]) END;%	02984
TYP:=IF T1.FORM=ARRAYS THEN T1.ARRTYPE ELSE REALTYPE;%	02985
T1:=TYPETAB1[TYP];%	02986
END UNTIL T1.STRUCT=0;%	02987
GEN("]",1,7);%	02988
END ELSE%	02989
BEGIN %*** FILE ***	02990
IF REALVAR OR ARRAYVAR THEN%	02991
BEGIN GEN(";",1,7); REALVAR:=ARRAYVAR:=FALSE END;%	02992
IF T1.FORM=TEXTFILE AND NOT PARAM THEN%	02993
BEGIN%	02994
IF NUMFILES>=MAXFILES THEN ERROR(97);%	02995
ELSE NUMFILES:=NUMFILES+1;%	02996
FILETAB[NUMFILES]:=NAM;%	02997
END;%	02998
EXTFILE:=FALSE;%	02999
FNAME:=NAMETAB1[LEVEL,NAM];%	03000
FNLENGTH := FNAME.NAMELENGTH; FNSTART := 8-FNLENGTH; %	03001
IF FNLENGTH LEQ 6 THEN %	03002
BEGIN%	03003
FOR J:=1 STEP 1 UNTIL NUMEXTFILES DO%	03004
IF FNAME=EXTFILETAB[J] THEN EXTFILE:=TRUE;%	03005
END;%	03006
IF EXTFILE AND NOT PARAM THEN%	03007
BEGIN%	03008
IF NUMFILES GEQ MAXFILES THEN ERROR(97);%	03009
ELSE%	03010
NUMFILES := NUMFILES + 1;%	03011
FILETAB[NUMFILES] := "NAM " 1;%	03012
GEN("DEFINE",7,2); GENID("F",LEVEL1000+NAM,5);%	03013
GEN("=",1,7);%	03014
GEN(FNAME,FNLENGTH,FNSTART); %	03015
GEN("#",2,6); GEN("SAVE",5,4); GEN("FILE",5,4);%	03016
GEN(FNAME,FNLENGTH,FNSTART); %	03017

END ELSE%	03018
BEGIN%	03019
GEN("FILE",5,4); GENID("F",LEVEL1000+NAM,5);%	03020
END;%	03021
IF NOT PARAM THEN%	03022
BEGIN%	03023
GEN("DISK",6,3); GEN("SERIAL",7,2);%	03024
IF EXTFILE THEN%	03025
BEGIN%	03026
IF ALGOLCNT LSS 13 THEN WRITEALGOL;%	03027
GEN("[0:0]",5,3);%	03028
GEN(";",1,7);%	03029
GEN(FNAME,FNLENGTH,FNSTART); %	03030
GEN(";",1,7); GEN("/ ",1,7);%	03031
IF ALGOLCNT<9 THEN WRITEALGOL;%	03032
GEN(";",1,7); GEN(USER,7,1); GEN(";",1,7);%	03033
END ELSE%	03034
BEGIN%	03035
GEN("[20:",4,4); GEN("300]",4,4);%	03036
END;%	03037
GEN("(1,",3,5);%	03038
RECSIZE:=IF T1.FORM=TEXTFILE THEN 10 ELSE%	03039
IF TYPETAB1[T1.FILETYPE].STRUCT=0 THEN 1 ELSE%	03040
TYPETAB3[T1.FILETYPE]-TYPETAB2[T1.FILETYPE]+1;%	03041
GENINT(RECSIZE); GEN(";",1,7);%	03042
IF RECSIZE=1 OR RECSIZE=10 THEN GENINT(150)%	03043
ELSE GENINT(RECSIZE);%	03044
IF ALGOLCNT LSS 10 THEN WRITEALGOL;%	03045
GEN(",SAVE",6,3); GEN("30",2,6);%	03046
GEN(");",2,6);%	03047
END ELSE GEN(";",1,7);%	03048
GEN("ARRAY",6,3); GENID("V",LEVEL1000+NAM,5);%	03049
GEN("[",1,7);%	03050
IF TYPETAB1[T1.FILETYPE].STRUCT=0 THEN%	03051
BEGIN%	03052
IF PARAM THEN GEN("0",1,7) ELSE GEN("0:0",3,5);%	03053
END ELSE%	03054
BEGIN%	03055
GENINT(TYPETAB2[T1.FILETYPE]);%	03056
IF NOT PARAM THEN%	03057
BEGIN GEN(";",1,7); GENINT(TYPETAB3[T1.FILETYPE]) END;%	03058
END;%	03059
GEN(");",2,6);%	03060
GEN("INTEGER",8,1); GENID("I",LEVEL1000+NAM,5);%	03061
GEN(";",1,7);%	03062
END;%	03063
END;%	03064
END OF LOOP;%	03065
IF REALVAR OR ARRAYVAR THEN GEN(";",1,7);%	03066
END OF DECLAREVARS;%	03067
%	03068
%	03069
PROCEDURE PARAMETERLIST;%	03070
BEGIN%	03071
INTEGER FIRSTPARAM,CURKIND,P1,PX,I,T;%	03072
BOOLEAN FIRST;%	03073
%	03074

```

DEFINE NEWPARAM=%
BEGIN%
  IF Numparams>=MAXPARAMS THEN%
  BEGIN ERROR(70); Numparams:=MAXPARAMS-10 END;%
  Numparams:=Numparams+1;%
END OF NEWPARAM#;%

%
NEWPARAM; FIRSTPARAM:=Numparams;%
IF CURSY=LPAR THEN%
BEGIN%
  DO BEGIN%
  INSYMBOL;%
  IF CURSY=VARSY OR CURSY=FUNCSY OR CURSY=PROCSY THEN%
  BEGIN%
  CURKIND:=IF CURSY=VARSY THEN VAR ELSE%
  IF CURSY=FUNCSY THEN FUNC ELSE PROC;%
  INSYMBOL;%
  END ELSE CURKIND:=CONST;%
  FIRST:=TRUE; P1:=Numparams+1;%
  DO BEGIN%
  IF FIRST THEN FIRST:=FALSE ELSE INSYMBOL;%
  IF CURSY=IDENTIFIER THEN%
  BEGIN%
  NEWNAME(CURNAME1,CURNAME2,CURLEVEL+1);%
  PX:=THISINDEX; PX.PARAMKIND:=CURKIND;%
  PX.PARAMLEVEL:=CURLEVEL+1;%
  NEWPARAM; PARAMTAB[Numparams]:=PX;%
  END ELSE ERROR(9);%
  INSYMBOL;%
  END UNTIL CURSY<=COMMA;%
  IF CURSY=COLON THEN%
  BEGIN%
  IF CURKIND=PROC THEN ERROR(90);%
  INSYMBOL;%
  IF CURSY=IDENTIFIER THEN%
  BEGIN%
  SEARCH;%
  IF FOUND THEN%
  BEGIN%
  IF THISID.IDCLASS=TYPES THEN%
  BEGIN%
  T3:=THISID.TYPE;%
  FOR I:=P1 STEP 1 UNTIL Numparams DO%
  PARAMTAB[I].PARAMTYPE:=T3;%
  IF CURKIND=CONST OR CURKIND=VAR THEN%
  BEGIN%
  T:=TYPETAB1[T3];%
  IF T.FORM>FILES THEN%
  FOR I:=P1 STEP 1 UNTIL Numparams DO%
  PARAMTAB[I].PARAMFILE:=1;%
  IF T.STRUCT>0 AND CURKIND=CONST THEN ERROR(94);%
  END ELSE IF T.STRUCT>0 THEN ERROR(38);%
  END ELSE BEGIN ERROR(7); T3:=0 END;%
  END ELSE BEGIN ERROR(1); T3:=0 END;%
  END ELSE BEGIN ERROR(9); T3:=0 END;%
  INSYMBOL;%
  END ELSE%

```

```

03075
03076
03077
03078
03079
03080
03081
03082
03083
03084
03085
03086
03087
03088
03089
03090
03091
03092
03093
03094
03095
03096
03097
03098
03099
03100
03101
03102
03103
03104
03105
03106
03107
03108
03109
03110
03111
03112
03113
03114
03115
03116
03117
03118
03119
03120
03121
03122
03123
03124
03125
03126
03127
03128
03129
03130
03131

```

```

BEGIN%                                03132
  IF CURKIND≠PROC THEN ERROR(7);%      03133
  T3:=0;%                               03134
END;%                                   03135
T3.IDCLASS:=CURKIND; T3.FORMAL:=1;%    03136
FOR I:=P1 STEP 1 UNTIL NUMPARAMS DO%   03137
  NAMETAB3[CURLEVEL+1,PARAMTAB[I],PARAMNAME]:=T3;% 03138
END UNTIL CURSY≠SEMICOLON;%           03139
IF CURSY≠RPAR THEN%                   03140
  BEGIN ERROR(46); SKIP(RPAR);%        03141
  IF CURSY=RPAR THEN INSYMBOL;%        03142
  END ELSE INSYMBOL;%                  03143
END;%                                   03144
PARAMTAB[FIRSTPARAM]:=NUMPARAMS-FIRSTPARAM;% 03145
END OF PARAMETERLIST;%                03146
%                                       03147
%                                       03148
PROCEDURE BLOCK;%                     03149
BEGIN%                                  03150
  INTEGER INDEX,CTYPE,NUMFORWARDS,T,T3,IX,I;% 03151
  REAL CVAL;%                           03152
  ALPHA C1,C2;%                         03153
  BOOLEAN VALUEPARAMS,FUN;%            03154
  LABEL START;%                         03155
%                                       03156
  INTEGER LABTABTOP,CONSTTABTOP,TYPETABTOP,PARAMTABTOP,TOPREC,% 03157
  FORMERFIRSTLAB,FIRSTFILE;%           03158
%                                       03159
  FORMERFIRSTLAB:=FIRSTLAB;%           03160
  LABTABTOP:=NUMLABS; FIRSTLAB:=LABTABTOP+1;% 03161
  CONSTTABTOP:=NUMCONSTS;%             03162
  TYPETABTOP:=NUMTYPES;%               03163
  PARAMTABTOP:=NUMPARAMS;%             03164
  TOPREC:=LASTREC;%                    03165
  FIRSTFILE:=NUMFILES+1;%              03166
%                                       03167
  TOPLEVEL:=CURLEVEL;%                 03168
  IF CURLEVEL>1 THEN GEN("BEGIN",6,3);% 03169
START;%                                  03170
  IF CURSY=LABELSY THEN                 %*** LABEL DECLARATION *** 03171
  BEGIN                                  %***** 03172
    GEN("LABEL",6,3);%                  03173
    DO BEGIN%                            03174
      INSYMBOL;%                        03175
      IF CURSY=INTCONST THEN%           03176
      BEGIN%                             03177
        GENID("L",CURVAL,4);%           03178
        IF CURVAL>9999 THEN ERROR(33);% 03179
        FOR I:=FIRSTLAB STEP 1 UNTIL NUMLABS DO% 03180
          IF LABTAB[I].LABVAL=CURVAL THEN ERROR(31);% 03181
          IF NUMLABS≥MAXLABS THEN BEGIN ERROR(34); NUMLABS:=0 END;% 03182
          NUMLABS:=NUMLABS+1;%          03183
          LABTAB[NUMLABS]:=CURVAL;%     03184
        INSYMBOL;%                      03185
      END ELSE BEGIN ERROR(10); SKIP(COMMA) END;% 03186
      IF CURSY=COMMA THEN GEN(", ",1,7);% 03187
    END UNTIL CURSY≠COMMA;%            03188
  END

```



14121  
McGraw-Hill Business Forms, Inc. sv

```
IF CURSY≠SEMICOLON THEN BEGIN ERROR(25); SKIP(SEMICOLON) END;% 03189
GEN(";",1,7);% 03190
IF SYMKIND[CURSY]≠INITIAL THEN INSYMBOL;% 03191
END OF LABEL DECLARATION;% 03192
% 03193
IF CURSY=CONSTSY THEN %*** CONSTANT DECLARATION *** 03194
BEGIN %***** 03195
INSYMBOL;% 03196
DO BEGIN% 03197
IF CURSY=IDENTIFIER THEN% 03198
BEGIN% 03199
NEWNAME(CURNAME1,CURNAME2,CURLEVEL); INDEX:=THISINDEX;% 03200
INSYMBOL;% 03201
IF CURSY=EQLSY THEN% 03202
BEGIN% 03203
INSYMBOL; CONSTANT(CVAL,CTYPE);% 03204
T3:=CTYPE; T3.IDCLASS:=CONST;% 03205
IF CVAL.[46:8]≠0 OR CVAL>1023 THEN% 03206
BEGIN% 03207
IF NUMCONSTS≥MAXCONSTS THEN% 03208
BEGIN ERROR(35); NUMCONSTS:=0 END;% 03209
NUMCONSTS:=NUMCONSTS+1;% 03210
CONSTTAB[ NUMCONSTS ]:=CVAL;% 03211
T3.INFO:=1023+NUMCONSTS;% 03212
END ELSE T3.INFO:=CVAL;% 03213
NAMETAB3[ CURLEVEL, INDEX ]:=T3;% 03214
END ELSE BEGIN ERROR(36); SKIP(SEMICOLON) END;% 03215
END ELSE BEGIN ERROR(9); SKIP(SEMICOLON) END;% 03216
IF CURSY≠SEMICOLON THEN BEGIN ERROR(25); SKIP(SEMICOLON) END;% 03217
IF SYMKIND[CURSY]≠INITIAL THEN INSYMBOL;% 03218
END UNTIL CURSY≠IDENTIFIER;% 03219
END OF CONSTANT DECLARATION;% 03220
% 03221
IF CURSY=TYPE SY THEN %*** TYPE DECLARATION *** 03222
BEGIN %***** 03223
INSYMBOL;% 03224
DO BEGIN% 03225
IF CURSY=IDENTIFIER THEN% 03226
BEGIN% 03227
NEWNAME(CURNAME1,CURNAME2,CURLEVEL); INDEX:=THISINDEX;% 03228
INSYMBOL;% 03229
IF CURSY=EQLSY THEN% 03230
BEGIN% 03231
INSYMBOL;% 03232
TYPEDECL(CTYPE, TX);% 03233
T3:=CTYPE; T3.IDCLASS:=TYPE S;% 03234
NAMETAB3[ CURLEVEL, INDEX ]:=T3;% 03235
END ELSE BEGIN ERROR(36); SKIP(SEMICOLON) END;% 03236
END ELSE BEGIN ERROR(9); SKIP(SEMICOLON) END;% 03237
IF CURSY≠SEMICOLON THEN BEGIN ERROR(25); SKIP(SEMICOLON) END;% 03238
IF SYMKIND[CURSY]≠INITIAL THEN INSYMBOL;% 03239
END UNTIL CURSY≠IDENTIFIER;% 03240
END OF TYPE DECLARATION;% 03241
% 03242
IF CURSY=VAR SY THEN %*** VARIABLE DECLARATION *** 03243
BEGIN %***** 03244
VARINDEX:=0;% 03245
```



```

IF(THISID.IDCLASS=PROC AND FUN)OR% 03303
  (THISID.IDCLASS=FUNC AND NOT FUN) THEN ERROR(43);% 03304
  INSYMBOL;% 03305
  END ELSE BEGIN ERROR(2); SKIP(SEMICOLON) END;% 03306
END-ELSE% 03307
BEGIN% 03308
  NEWNAME(CURNAME1,CURNAME2,CURLEVEL); INDEX:=THISINDEX;% 03309
  T3:=0; T3.INFO:=NUMPARAMS+1;% 03310
  T3.IDCLASS:=IF FUN THEN FUNC ELSE PROC;% 03311
  NAMETAB3[CURLEVEL,INDEX]:=T3;% 03312
  INSYMBOL; PARAMETERLIST;% 03313
  IF CURSY=COLON THEN% 03314
  BEGIN% 03315
  IF NOT FUN THEN ERROR(48);% 03316
  INSYMBOL;% 03317
  IF CURSY=IDENTIFIER THEN% 03318
  BEGIN% 03319
  SEARCH;% 03320
  IF FOUND THEN% 03321
  BEGIN% 03322
  IF THISID.IDCLASS=TYPES THEN% 03323
  BEGIN% 03324
  T:=TYPETAB1[THISID,TYPE];% 03325
  IF T.FORMSALFA OR T.FORM=POINTERS THEN% 03326
  BEGIN% 03327
  NAMETAB3[CURLEVEL,INDEX].TYPE:=THISID.TYPE;% 03328
  END ELSE ERROR(38);% 03329
  END ELSE ERROR(7);% 03330
  END ELSE ERROR(1);% 03331
  END ELSE ERROR(9);% 03332
  INSYMBOL;% 03333
  END-ELSE-IF FUN THEN% 03334
  BEGIN ERROR(26); SKIP(SEMICOLON) END;% 03335
  END;% 03336
  END-ELSE BEGIN ERROR(9); SKIP(SEMICOLON) END;% 03337
  IF CURSY#SEMICOLON THEN BEGIN ERROR(25); SKIP(SEMICOLON) END;% 03338
  IF FUN THEN GEN("FUNCTN",7,2) ELSE% 03339
  GEN("PROCEDU",8,1); GENID("V",1000×CURLEVEL+INDEX,5);% 03340
  T:=NAMETAB3[CURLEVEL,INDEX].INFO; TX:=T+PARAMTAB[T];% 03341
  IF TX>T THEN% 03342
  BEGIN% 03343
  GEN("(",1,7);% 03344
  FOR I:=T+1 STEP 1 UNTIL TX DO% 03345
  BEGIN GENID("V",1000×(CURLEVEL+1)+PARAMTAB[I].PARAMNAME,5);% 03346
  IF BOOLEAN(PARAMTAB[I].PARAMFILE) THEN% 03347
  BEGIN% 03348
  GEN("(",1,7);% 03349
  GENID("F",1000×(CURLEVEL+1)+PARAMTAB[I].PARAMNAME,5);% 03350
  GEN("(",1,7);% 03351
  GENID("I",1000×(CURLEVEL+1)+PARAMTAB[I].PARAMNAME,5);% 03352
  END;% 03353
  IF I LSS TX THEN GEN("(",1,7);% 03354
  END;% 03355
  GEN(")",2,6);% 03356
  VALUEPARAMS:=FALSE;% 03357
  FOR I:=T+1 STEP 1 UNTIL TX DO% 03358
  IF PARAMTAB[I].PARAMKIND=CUNST THEN% 03359

```

```

BEGIN%                                03360
  IF NOT VALUEPARAMS THEN%           03361
  BEGIN GEN("VALUE",6,3);%           03362
    VALUEPARAMS:=TRUE;%             03363
  END ELSE GEN(";",1,7);%           03364
  GENID("V",1000*(CURLEVEL+1)+PARAMTAB[I].PARAMNAME,5);% 03365
  END;%                                03366
  IF VALUEPARAMS THEN GEN(";",1,7);% 03367
  DECLAREVARS(TRUE,PARAMTAB,T+1,IX,CURLEVEL+1);% 03368
  END ELSE GEN(";",1,7);%           03369
%                                     03370
  INSYMBOL;%                          03371
  IF CURNAME1="7FORWAR" AND CURNAME2="D" THEN% 03372
  BEGIN%                                03373
    NAMETAB3[CURLEVEL,INDEX].FORWARDDEF:=1;% 03374
    NUMFORWARDS:=NUMFORWARDS+1;%      03375
    GEN("FORWARD",8,1);%             03376
    INSYMBOL;%                          03377
  END ELSE%                              03378
  BEGIN%                                03379
    CURLEVEL:=CURLEVEL+1;%           03380
    IF CURLEVEL>=LASTREC THEN ERROR(55);% 03381
    BLOCKTAB[CURLEVEL]:=NUMBLOCKS:=NUMBLOCKS+1;% 03382
    T:=CURFUNC; CURFUNC:=IF FUN THEN INDEX ELSE -1;% 03383
    BLOCK; %*** COMPILE PROCEDURE BODY *** 03384
    REPLACE POINTER(NAMETAB1[CURLEVEL,*]) BY 0% 03385
      FOR MAXNAMES+1 WORDS;%          03386
    CURLEVEL:=CURLEVEL-1; CURFUNC:=T;% 03387
    TOPLEVEL:=CURLEVEL;%             03388
  END;%                                03389
  IF CURSY<SEMICOLON THEN BEGIN ERROR(25); SKIP(SEMICOLON) END;% 03390
  GEN(";",1,7);%                       03391
  IF SYMKIND[CURSY]≠INITIAL THEN INSYMBOL;% 03392
  END OF PROCEDURE DECLARATION;%      03393
%                                     03394
%                                     03395
  IF NUMFORWARDS>0 THEN ERROR(44);%   03396
  GEN("INTEGER",8,1);%                 03397
  FOR I:=1 STEP 1 UNTIL MAXTEMPS DO%   03398
  BEGIN GENID("T",I,2);%              03399
    IF I<MAXTEMPS THEN GEN(";",1,7) ELSE GEN(";",1,7);% 03400
  END;%                                03401
  IF CURSY≠BEGINSY THEN%              03402
  BEGIN ERROR(39);%                    03403
    WHILE SYMKIND[CURSY]≠INITIAL DO%   03404
    BEGIN INSYMBOL; SKIP(SEMICOLON) END;% 03405
    IF(CURSY=TYPE$Y)OR(CONST$Y$CURSY AND CURSY$PROCSY)THEN% 03406
    GO TO START;%                       03407
  END;%                                03408
  IF CURLEVEL=1 THEN%                  03409
  BEGIN%                                03410
    GEN("INIT(",5,3);%                 03411
    IF INPUTDECL THEN GEN("TRUE",4,4) ELSE GEN("FALSE",5,3);% 03412
    GEN(")",2,6);%                     03413
  END;%                                03414
  FOR I:=FIRSTFILE STEP 1 UNTIL NUMFILES DO% 03415
  IF FILETAB[I] LSS 0 THEN%           03416

```

```

BEGIN%
  GEN("CHFIL(",6,2);%
  GENID("F",1000×CURLEVEL-FILETAB[1]-1,5);%
  GEN(")",2,6);%
END%
ELSE%
BEGIN%
  GENID("I",1000×CURLEVEL+FILETAB[1],5);%
  GEN(".",1,7); GEN("BUFSIZE",7,1); GEN(":=80;",5,3);%
END;%
NUMFILES:=FIRSTFILE-1;%
%
COMPSTAT;   %*** COMPILER STATEMENT PART ***
%
FOR I:=LASTREC STEP 1 UNTIL TOPREC-1 DO%
  REPLACE POINTER(NAMETAB[I,*]) BY 0 FOR MAXNAMES+1 WORDS;%
FOR I:=FIRSTLAB STEP 1 UNTIL NUMLABS DO%
  IF LABTAB[I].LABDEF=0 THEN ERROR(93);%
  LASTREC:=TOPREC;%
  NUMLABS:=LABTABTOP;%
  FIRSTLAB:=FORMERFIRSTLAB;%
  NUMCONSTS:=CONSTTABTOP;%
  NUMTYPES:=TYPETABTOP;%
  NUMPARAMS:=PARAMTABTOP;%
  IF CURLEVEL>1 THEN GEN("END",4,5);%
END OF BLOCK;%
$ PAGE%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%
%
PART 9:  THE MAIN PROGRAM.
%
%
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%
INTEGER PROGNAMELENGTH;%
ALPHA PRUGNAME,ALGOLNAME;%
%
ALGOLNAME:="PASC000"&ENTIER(TIME(4)MOD 10)[17:5:6];%
ALGOLNAME:=ALGOLNAME&ENTIER(TIME(4)DIV 7)[11:5:6];%
ALGOLNAME:=ALGOLNAME&ENTIER(TIME(4)MOD 9)[5:5:6];%
USER:=TIME(-1);%
FILL PASCALGOL WITH ALGOLNAME,USER;%
BEGIN%
  FILE PASCALGOL SERIAL "PASCALGOL"/"DISK" (2,10,150);%
  ARRAY BUF[0:9];%
  LABEL EOF;%
%
  WHILE TRUE DO%
  BEGIN%
    READ(PASCALGOL,9,BUF[*]) [EOF];%
    WRITE(PASCALGOL,10,BUF[*]);%
  END;%

```

```

03417
03418
03419
03420
03421
03422
03423
03424
03425
03426
03427
03428
03429
03430
03431
03432
03433
03434
03435
03436
03437
03438
03439
03440
03441
03442
03443
03444
03445
03446
03447
03448
03449
03450
03451
03452
03453
03454
03455
03456
03457
03458
03459
03460
03461
03462
03463
03464
03465
03466
03467
03468
03469
03470
03471
03472
03473

```

```

EOF;% 03474
END-OF-TRANSFER OF RUN TIME SYSTEM;% 03475
CARDLENGTH:=72;% 03476
INITIALIZE; NEWCARD;% 03477
LISTOPTION:=CHECKOPTION:=TRUE;% 03478
C:=" "; INSYMBOL;% 03479
IF CURSY=PROGRAMSY THEN% 03480
BEGIN% 03481
INSYMBOL;% 03482
IF CURSY=IDENTIFIER THEN% 03483
BEGIN% 03484
PROGRAM:=CURNAME1.[35:36]; PROGRAMELENGTH:=MIN(6,CURLENGTH);% 03485
INSYMBOL;% 03486
IF CURSY=LPAR THEN% 03487
BEGIN% 03488
DO BEGIN% 03489
INSYMBOL;% 03490
IF CURSY=IDENTIFIER THEN% 03491
BEGIN% 03492
IF CURNAME1="50INPUT" THEN INPUTDECL:=TRUE ELSE% 03493
IF CURNAME1="60OUTPUT" THEN OUTPUTDECL:=TRUE ELSE% 03494
BEGIN% 03495
IF CURLENGTH>6 THEN ERROR(77);% 03496
NUMEXTFILES:=NUMEXTFILES+1;% 03497
IF NUMEXTFILES<=MAXEXTFILES THEN% 03498
EXTFILETAB(NUMEXTFILES):=CURNAME1 ELSE% 03499
IF NUMEXTFILES=MAXEXTFILES+1 THEN ERROR(73);% 03500
END;% 03501
END ELSE ERROR(9);% 03502
INSYMBOL;% 03503
END UNTIL CURSY<=COMMA;% 03504
IF CURSY<=RPAR THEN BEGIN ERROR(46); SKIP(SEMICOLON) END;% 03505
IF CURSY=RPAR THEN INSYMBOL;% 03506
IF CURSY<=SEMICOLON THEN BEGIN ERROR(25); SKIP(SEMICOLON) END;% 03507
END ELSE BEGIN ERROR(58); SKIP(SEMICOLON) END;% 03508
END ELSE BEGIN ERROR(9); SKIP(SEMICOLON) END;% 03509
END ELSE BEGIN ERROR(75); SKIP(SEMICOLON) END;% 03510
INSYMBOL;% 03511
CURLEVEL:=1;% 03512
LASTREC:=MAXTABLES+1;% 03513
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% 03514
% 03515
BLOCK; % COMPILE USER PROGRAM. % 03516
% 03517
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% 03518
IF CURSY<=DOT THEN% 03519
BEGIN% 03520
ERROR(76);% 03521
DO BLOCK UNTIL CURSY=DOT;% 03522
END;% 03523
IF FALSE THEN% 03524
BEGIN% 03525
ENDOFINPUT: ERROR(87); CHARCNT:=-1;% 03526
WRITE(LINES,TERMMESS);% 03527
END;% 03528
IF LISTOPTION AND CHARCNT<=0 THEN PRINTLINE;% 03529
IF ERRINX>0 THEN PRINTERRORS;% 03530

```

14121  
Roore Business Forms, Inc. NY

```

WRITE(LINESIDBL));%
WRITE(LINESIDBL));%
IF NUMERRS=0 THEN%
BEGIN%
  ARRAY ZIPARRAY[0:19],Z[0:0];%
  POINTER ZIPPNT;%
%
  DEFINE ZIPTEXT(TEXT,L)=%
  BEGIN%
    Z[0]:=TEXT;%
    REPLACE ZIPPNT:ZIPPNT BY POINTER(Z[*])+(8=L) FOR L;%
  END#;%
%
  PROCEDURE ZIPNUM(N);% TRANSFERS A NUMBER TO THE ZIP BUFFER.
  VALUE N; INTEGER N;%
  IF N<9 THEN ZIPTEXT(N,1) ELSE%
  BEGIN ZIPNUM(N DIV 10); ZIPTEXT(ENTIER(N MOD 10),1) END;%
%
  WRITEALGOL;%
  WRITE(PASCALGOL, LASTLINE);%
  LOCK(PASCALGOL,SAVE);%
  ZIPPNT:=POINTER(ZIPARRAY[*]);%
  REPLACE ZIPPNT BY " " FOR 20 WORDS;%
  WRITE(LINES,NOERRORS);%
  ZIPTEXT("CC ",3); ZIPTEXT("COMPILE",7);%
  ZIPTEXT(" ",1); ZIPTEXT(PRUGNAME,PROGNAMELENGTH);%
  ZIPTEXT("/ ",1); ZIPTEXT(USER,7);%
  ZIPTEXT(" XALGOL",7); ZIPTEXT(" ",1);%
  IF SAVEFACTOR>0 THEN ZIPTEXT("LIBRARY",7);%
  IF SAVEFACTOR<0 THEN ZIPTEXT("SYNTAX",6);%
  ZIPTEXT(";",1);%
  ZIPTEXT("XALGOL",6); ZIPTEXT(" FILE",5);%
  ZIPTEXT(" CARD=",6); ZIPTEXT(ALGOLNAME,7);%
  ZIPTEXT("/ ",1); ZIPTEXT(USER,7);%
  ZIPTEXT(" SERIAL",7); ZIPTEXT(";",1);%
  IF SAVEFACTOR>0 THEN%
  BEGIN%
    ZIPTEXT("SAVE=",5); ZIPNUM(SAVEFACTOR);%
    ZIPTEXT(";",1);%
  END;%
  ZIPTEXT("END.",4);%
  ZIP WITH ZIPARRAY[*];%
END OF COMPILER ZIP ELSE%
BEGIN%
  INTEGER I;%
  SWITCH FORMAT ERRORMESS1 :=%
  (" 0 *** COMPILER ERROR *** CONTACT THE COMPUTER CENTRE."),%
  (" 1 IDENTIFIER NOT DEFINED."),%
  (" 2 IDENTIFIER ALREADY DEFINED."),%
  (" 3 WRONG NUMBER OF PARAMETERS."),%
  (" 4 SYNTAX ERROR."),%
  (" 5 VARIABLE NOT ACCESSIBLE (HARDWARE RESTRICTION)."),%
  (" 6 STRINGS MAY NOT BE CONTINUED FROM ONE CARD TO ANOTHER."),%
  (" 7 A TYPE EXPECTED."),%
  (" 8 VARIABLE EXPECTED."),%
  (" 9 IDENTIFIER EXPECTED."),%
  (" 10 INTEGER CONSTANT EXPECTED."),%

```

03531  
03532  
03533  
03534  
03535  
03536  
03537  
03538  
03539  
03540  
03541  
03542  
03543  
03544  
03545  
03546  
03547  
03548  
03549  
03550  
03551  
03552  
03553  
03554  
03555  
03556  
03557  
03558  
03559  
03560  
03561  
03562  
03563  
03564  
03565  
03566  
03567  
03568  
03569  
03570  
03571  
03572  
03573  
03574  
03575  
03576  
03577  
03578  
03579  
03580  
03581  
03582  
03583  
03584  
03585  
03586  
03587

(" 11	CONSTANT OF OTHER TYPE THAN EXPECTED.)",%	03588
(" 12	VARIABLE OF ILLEGAL TYPE.)",%	03589
(" 13	UNRECOGNIZABLE STATEMENT.)",%	03590
(" 14	CONSTANT TOO BIG OR TOO SMALL.)",%	03591
(" 15	UNDEFINED LABEL.)",%	03592
(" 16	FOR AND CASE STATEMENTS NESTED TOO DEEP.)",%	03593
(" 17	EXPRESSION IS OF WRONG TYPE.)",%	03594
(" 18	""OF"" EXPECTED.)",%	03595
(" 19	""DU"" EXPECTED.)",%	03596
(" 20	""ELSE"" WITHOUT CORRESPONDING ""THEN"".)",%	03597
(" 21	ILLEGAL TERMINATION OF STATEMENT.)",%	03598
(" 22	""UNTIL"" EXPECTED.)",%	03599
(" 23	""TO""/""DOWNTO"" EXPECTED.)",%	03600
(" 24	""END"" EXPECTED.)",%	03601
(" 25	"";"" EXPECTED.)",%	03602
(" 26	"":"" EXPECTED.)",%	03603
(" 27	""THEN"" EXPECTED.)",%	03604
(" 28	"":="" EXPECTED.)",%	03605
(" 29	ONLY NUMBERS MAY BE SIGNED.)",%	03606
(" 30	TOO MANY CASES.)",%	03607
(" 31	LABEL USED MORE THAN ONCE.)",%	03608
(" 32	CONSTANT EXPECTED.)",%	03609
(" 33	LABEL NOT IN RANGE 0..9999.)",%	03610
(" 34	TOO MANY LABELS DECLARED.)",%	03611
(" 35	TOO MANY CONSTANTS DECLARED.)",%	03612
(" 36	""=""" EXPECTED.)",%	03613
(" 37	THE LIST IS TOO LONG.)",%	03614
(" 38	INVALID TYPE FOR A FUNCTION.)",%	03615
(" 39	""BEGIN"" EXPECTED.)",%	03616
(" 40	TOO MANY IDENTIFIERS DECLARED.)",%	03617
(" 41	ALFA CONSTANS MAY NOT BE LONGER THAN 7 CHARACTERS.)",%	03618
(" 42	EXPRESSION IS NOT OF TYPE BOOLEAN.)",%	03619
(" 43	NOT PROPER FORWARD DECLARATION.)",%	03620
(" 44	UNSATISFIED FORWARD DECLARATION.)",%	03621
(" 45	TOO MANY DIFFERENT TYPES DECLARED.)",%	03622
(" 46	""()"" EXPECTED.)",%	03623
(" 47	""[""" EXPECTED.)",%	03624
(" 48	A SIMPLE TYPE EXPECTED.)",%	03625
(" 49	""ARRAY OF ARRAY"" AND ""ARRAY OF RECORD"" ILLEGAL",% " AS FILE TYPE.)",%	03626 03627
(" 50	""FILE OF FILE"" IS ILLEGAL.)",%	03628
(" 51	SET BOUNDRY IS TOO BIG OR TOO SMALL.)",%	03629
(" 52	TOO MANY UNDECLARED POINTERS.)",%	03630
(" 53	"".,"" EXPECTED.)",%	03631
(" 54	FIRST VALUE IS GREATER THAN SECOND VALUE.)",%	03632
(" 55	TOO MANY RECORDS DECLARED AT ONE TIME.)",%	03633
(" 56	THE RECORD CONTAINS MORE THAN 1023 WORDS.)",%	03634
(" 57	FILES NOT ALLOWED IN RECORDS.)",%	03635
(" 58	""(""" EXPECTED.)",%	03636
(" 59	"")]"" EXPECTED.)",%	03637
	%	03638
	SWITCH FORMAT ERRORMESS2 :=%	03639
(" 60	""ARRAY OF FILE"" NOT ALLOWED.)",%	03640
(" 61	RANGE OF INDEX IS GREATER THAN 1023.)",%	03641
(" 62	UNSATISFIED POINTER DECLARATION.)",%	03642
(" 63	EXPRESSION IS TOO LONG.)",%	03643
(" 64	ILLEGAL OPERATOR FOR THIS TYPE OF EXPRESSION.)",%	03644



```

(" 65 INTEGER EXPRESSION EXPECTED."):,% 03645
(" 66 A SET EXPECTED."):,% 03646
(" 67 PARAMETER OF ILLEGAL TYPE."):,% 03647
(" 68 PROCEDURES NOT ALLOWED IN THIS CONTEXT."):,% 03648
(" 69 ILLEGAL USE OF THIS TYPE OF IDENTIFIER."):,% 03649
(" 70 TOO MANY PARAMETERS DECLARED IN THE PROGRAM."):,% 03650
(" 71 ""ARRAY OF CHAR"" EXPECTED."):,% 03651
(" 72 WRONG TYPE OF SET EXPRESSION."):,% 03652
(" 73 TOO MANY EXTERNAL FILES."):,% 03653
(" 74 ILLEGAL IDENTIFIER FOR EXTERNAL FILE."):,% 03654
(" 75 ""PROGRAM"" EXPECTED."):,% 03655
(" 76 """, """" EXPECTED AT END OF PROGRAM."):,% 03656
(" 77 EXTERNAL FILE IDENTIFIER MAY NOT EXCEED 6 CHARACTERS."):,% 03657
(" 78 ILLEGAL FILE PARAMETER."):,% 03658
(" 79 ILLEGAL USE OF FILE HANDLING PROCEDURE."):,% 03659
(" 80 TEXT-FILE EXPECTED."):,% 03660
(" 81 POINTER VARIABLE EXPECTED."):,% 03661
(" 82 ONLY VALUES OF TYPE REAL, INTEGER OR CHAR MAY BE READ."):,% 03662
(" 83 VARIABLES IN RECORDS ILLEGAL IN THIS CONTEXT."):,% 03663
(" 84 DISPLAY OVERFLOW."):,% 03664
(" 85 READ AND WRITE MAY ONLY BE USED ON TEXT-FILES."):,% 03665
(" 86 REFERENCED OBJECT IS TOO BIG."):,% 03666
(" 87 END-OF-INPUT DISCOVERED."):,% 03667
(" 88 CHARACTER ARRAY EXPECTED."):,% 03668
(" 89 """, """" EXPECTED."):,% 03669
(" 90 PROCEDURES MAY NOT HAVE ANY TYPE."):,% 03670
(" 91 PARAMETER OF WRONG KIND."):,% 03671
(" 92 ONLY COMPLETE ARRAYS AND RECORDS MAY BE TRANSMITTED."):,% 03672
(" 93 DECLARED LABEL NOT USED."):,% 03673
(" 94 PARAMETERS OF THIS TYPE SHOULD NOT BE VALUE PARAMETERS."):,% 03674
(" 95 ASSIGNMENT OF STRUCTURED VARIABLES NOT IMPLEMENTED."):,% 03675
(" 96 INPUT/OUTPUT NOT DECLARED."):,% 03676
(" 97 TOO MANY FILES IN USE."):,% 03677
(" 98 RECORD IDENTIFIER EXPECTED."):,% 03678
(" 99 UNRECOGNIZABLE ITEM."):,% 03679
( );,% 03680
,% 03681
,% 03682
WRITE(LINES, ERRORS, NUMERRS);,% 03683
FOR I:=0 STEP 1 UNTIL 59 DO IF ERR[I] THEN% 03684
WRITE(LINES, ERRORMESS1[I]);,% 03685
FOR I:=60 STEP 1 UNTIL 119 DO IF ERR[I] THEN% 03686
WRITE(LINES, ERRORMESS2[I-60]);,% 03687
END OF ERROR MESSAGES;% 03688
IF XREFOPTION THEN% 03689
BEGIN% 03690
REPLACE POINTER(XREFLINE[*]) BY " " FOR 17 WORDS;% 03691
HEADING;% 03692
SORT(PRINTXREF, XREFFILE, 0, XREFMAX, XREFCOMPARE, 3, 1000, 6000);% 03693
END;% 03694
END OF COMPILER,% 03695

```

LABEL 00000000LINE 00176362CC EX 0/R;COMMON=1;FILE S=ELPASO/819005 DISK SERIAL;END\*

0 /R