

?USER = SITE

PACKET 221  
INPUT 8 CARDS FROM CRA  
TIME 1346  
DATE 78109 WEDNESDAY, 04/19/78

\*\*\* BURROUGHS B5700 DCMCP MARK XVI.0.178 AND INTRINSICS MARK XVI.0.132 \*\*\*

#NO MESSAGES TODAY

13146130 ?USER= SITE  
13146130 ?COMPILE PASCAL/JUNK XALGOL XYNTAX  
13146130 ?XALGOL FILE TAPE= ELPASO/U819005 DISK SERIAL  
13146131 ?XALGOL STACK= 1000  
13146131 ?DATA029 CARD  
13146131 51XALGOL/PASCAL= 2 BOJ 1346 04/17/78  
13146133 CDA IN CARD DA1XALGOL/PASCAL= 2  
13146134 #NO FIL ON DISK ELPASO U8190051XALGOL/PASCAL= 2  
13146149 IL OK DS  
13146149 #NO FIL ON DISK ELPASO U8190051XALGOL/PASCAL= 2  
13147142 +OPERATOR KEYED IN: 20K  
13147143 DKA IN SER ELPASO U8190051XALGOL/PASCAL= 2  
13147144 PBD0222 OUT 011 LINE1XALGOL/PASCAL= 2  
13147148 DKA OUT SER DSK2 SITE1XALGOL/PASCAL= 2  
13147155 DKA OUT RDM PASCAL JUNK1XALGOL/PASCAL= 2  
13147158 DKA OUT SER DSK1 SITE1XALGOL/PASCAL= 2  
13154118 +OPERATOR KEYED IN: 2TL  
13154118 TIME LIMITS FOR XALGOL/PASCAL= 2 ARE: PRT=NO LIMIT; IOT=NO LIMIT.  
13154127 +OPERATOR KEYED IN: 2TI  
13154127 TIME FOR XALGOL/PASCAL= 2 IS: CP= 5147, IO= 1155 IN 7:57  
13154128 DKA LOK PASCAL JUNK1XALGOL/PASCAL= 2  
13154129 CDA REL CARD DA1XALGOL/PASCAL= 2  
13154129 ?END  
13154129 DKA REL ELPASO U8190051XALGOL/PASCAL= 2  
13154133 DKA OUT SER DSRT1 SITE1XALGOL/PASCAL= 2  
13154145 DKA OUT SER DSRT2 SITE1XALGOL/PASCAL= 2  
13154155 DKA REL DSRT1 SITE1XALGOL/PASCAL= 2  
13154155 DKA REL DSRT2 SITE1XALGOL/PASCAL= 2  
13154155 DKA OUT SER DSRT1 SITE1XALGOL/PASCAL= 2  
13157130 DKA OUT SER DSRT2 SITE1XALGOL/PASCAL= 2  
14100122 DKA REL DSRT1 SITE1XALGOL/PASCAL= 2  
14100123 DKA REL DSRT2 SITE1XALGOL/PASCAL= 2  
14100125 DKA REL DSK2 SITE1XALGOL/PASCAL= 2  
14100126 DKA REL DSK1 SITE1XALGOL/PASCAL= 2  
14100127 PBD0222 REL 011 LINE 62101XALGOL/PASCAL= 2  
14100129 XALGOL/PASCAL= 2 EOJ 1400  
14100129 FOR XALGOL/PASCAL= 2: PROCESS= 613 SECS, IO= 203 SECS, OLAY= 13  
14100131 51PASCAL/JUNK=01 SCHEDULED 1400, NEEDS 27264  
14101104 1ES  
14101105 -OPRTR ES=ED PASCAL/JUNK= 2, S= 1, A= 010

4:01:07  
4:01:08  
4:01:09

PASCAL/JUNK= 2 DS-ED 1401  
FOR PASCAL/JUNK= 2: PROCESS= 0 SECS, IO= 8 SECS, OLAY= 0  
PKT#0221 REMOVED

PASCAL /JUNK  
 =====

SOURCE FILE: ELPASO /U819005

```

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
%
%
%
% * P A S C A L   C O M P I L E R *
% *****
%
%
% WRITTEN 1975 BY
%   DAG F. LANGMYHR,
%   HERIOT-WATT UNIVERSITY,
%   EDINBURGH,
%
%
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
%
%
% PART 1:  DECLARATIONS,
%   -----
%
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
%
%
% BEGIN%
  
```

00000001	T	0000
00000002	T	0000
00000003	T	0000
00000004	T	0000
00000005	T	0000
00000006	T	0000
00000007	T	0000
00000008	T	0000
00000009	T	0000
00000010	T	0000
00000011	T	0000
00000012	T	0000
00000013	T	0000
00000014	T	0000
00000015	T	0000
00000016	T	0000
00000017	T	0000
00000018	T	0000
00000019	T	0000
00000020	T	0000
00000021	T	0000
00000022	T	0000
00000023	T	0000
00000024	T	0000
00000025	T	0000

PRT(22) = \*LIST, LABEL, OR SEGMENT DESCRIPTOR\*  
 PRT(23) = \*OUTER BLOCK DESCRIPTOR\*  
 PRT(24) = \*SEGMENT DESCRIPTOR\*

		START OF SEGMENT *****	2
	00000026	T	0000
	00000027	T	0000
	00000028	T	0000
	00000029	T	0000
	00000030	T	0000
	00000031	T	0000
	00000032	T	0000
	00000033	T	0000
	00000034	T	0003
	00000035	T	0007

```

          DEFINE EDITION="2,3"%;
          INTEGER NUMERRS,          % @R+21: NUMBER OF ERRORS IN PROGRAM,
PRT(25) = NUMERRS
          SAVEFACTOR,             % @R+22: SAVEFACTOR FOR CODE FILE,
PRT(26) = SAVEFACTOR
          %
          % >0 COMPILE TO LIBRARY,
          % =0 COMPILE AND RUN,
          % <0 COMPILE FOR SYNTAX,
          CARDCNT;               % @R+23: NUMBER OF CARDS READ,
PRT(27) = CARDCNT
          FILE CARD "SOURCE" (2,10,150);          % SOURCE CODE INPUT FILE
PRT(30) = CARD
          FILE LINES 1 (2,17);                    % PRINT FILE,
PRT(31) = LINES
          FILE PASCALGOL DISK SERIAL [20:600] (2,10,150,SAVE 0); % CODE FILE
  
```

PRT(32) = PASCALGOL  
PRT(33) =  
PRT(34) = FILE ATTRBUTS

DEFINE LINESPERPAGE=52#,%  
MAXINT=549755813887#;%

%  
%\*\*\* COMPILER CONSTANTS \*\*\*

DEFINE MAXTABLES =50#,% %MAX NUMBER OF NAME TABLES,  
MAXNAMES =997#,% %MAX NAMES IN EACH TABLE,  
MAXLEVEL =15#,% %MAX DEPTH OF PROCEDURE DECLARATIONS,  
MAXCASES =211#,% %MAX LABELS IN A CASE=STATEMENT,  
MAXLABS =100#,% %MAX NUMBER OF LABELS,  
MAXPARAMS =200#,% %MAX NUMBER OF PARAMETERS IN WHOLE PROGRAM,  
MAXTYPES =1022#,% %MAX NUMBER OF DIFFERENT TYPES,  
MAXCONSTS =200#,% %SIZE OF CONSTANT TABLE,  
MAXTEMPS =5#,% %NUMBER OF EXTRA VARS IN EACH PROCEDURE,  
MAXWITHSYMS=250#,% %MAX NUMBER OF SYMBOLS USED BY WITH=STATMS,  
MAXSYMS =800#,% %MAX NUMBER OF SYMBOLS IN ONE EXPRESSION,  
LISTLENGTH =800#,% %MAX LENGTH OF VAR AND PARAM LISTS,  
MAXEXTFILES=20#,% %MAX NUMBER OF EXTERNAL FILES,  
MAXFILES =20#,% %MAX NUMBER OF FILES DECLARED AT ONE TIME,  
MAXPNTRS =50#,% %MAX NUMBER OF UNDECLARED POINTERS,

%  
%\*\*\* NAME TABLES \*\*\*

ARRAY NAMETAB1,NAMETAB2,NAMETAB3[0:MAXTABLES,0:MAXNAMES];%

PRT(35) = NAMETAB1  
PRT(36) = NAMETAB2  
PRT(37) = NAMETAB3

DEFINE NAMELENGTH =[41:6]#,%  
TYPE =[9:10]#,%  
IDCLASS =[12:3]#,%  
VAR =0#,%  
CONST=1#,%  
FUNC =2#,%  
PROC =3#,%  
TYPES=4#,%

INFO =[23:11]#,%  
FORMAL =[24:1]#,%  
FORWARDDEF =[25:1]#,%  
EXTERNALFILE=[26:1]#;%

%  
%\*\*\* DISPLAY VECTOR \*\*\*

ARRAY DISPLAY[0:MAXLEVEL];%

PRT(40) = DISPLAY

DEFINE RECTYPE =[9:10]#,%  
FIRSTWITHSYM =[19:10]#,%  
LASTWITHSYM =[29:10]#,%  
NUMPNTRSINWITH=[35:6]#,%  
BRACKETSINWITH=[36:1]#,%  
NAMETAB =[46:7]#;%

%  
%\*\*\* TYPE TABLES \*\*\*

ARRAY TYPETAB1,TYPETAB2,TYPETAB3[0:MAXTYPES];%

PRT(41) = TYPETAB1  
PRT(42) = TYPETAB2  
PRT(43) = TYPETAB3

DEFINE FORM =[3:4]#,%  
NUMERIC =0#,%

00000036 P 0013  
00000037 T 0013  
00000038 T 0013  
00000039 T 0013  
00000040 T 0013  
00000041 T 0013  
00000042 T 0013  
00000043 T 0013  
00000044 T 0013  
00000045 T 0013  
00000046 T 0013  
00000047 T 0013  
00000048 T 0013  
00000049 T 0013  
00000050 T 0013  
00000051 T 0013  
00000052 T 0013  
00000053 T 0013  
00000054 T 0013  
00000055 T 0013  
00000056 T 0013  
00000057 T 0013

00000058 T 0015  
00000059 T 0015  
00000060 T 0015  
00000061 T 0015  
00000062 T 0015  
00000063 T 0015  
00000064 T 0015  
00000065 T 0015  
00000066 T 0015  
00000067 T 0015  
00000068 T 0015  
00000069 T 0015  
00000070 T 0015  
00000071 T 0015  
00000072 T 0015

00000073 T 0017  
00000074 T 0017  
00000075 T 0017  
00000076 T 0017  
00000077 T 0017  
00000078 T 0017  
00000079 T 0017  
00000080 T 0017  
00000081 T 0017

00000082 T 0019  
00000083 T 0019

SYMBOLIC=1#,%	00000084 T	0019
SUBTYPE =2#,%	00000085 T	0019
MAINTYPE=[33:10]#,%	00000086 T	0019
CHAR =3#,%	00000087 T	0019
FLOATING=4#,%	00000088 T	0019
ALFA =5#,%	00000089 T	0019
SET =6#,%	00000090 T	0019
SETTYPE = [33:10]#,%	00000091 T	0019
POINTERS=7#,%	00000092 T	0019
POINTTYPE=[33:10]#,%	00000093 T	0019
ARRAYS =8#,%	00000094 T	0019
INXTYPE = [33:10]#,%	00000095 T	0019
ARRTYPE = [43:10]#,%	00000096 T	0019
RECORD =9#,%	00000097 T	0019
RECTAB = [33:10]#,%	00000098 T	0019
FILES =10#,%	00000099 T	0019
FILETYPE=[33:10]#,%	00000100 T	0019
TEXTFILE=11#,%	00000101 T	0019
SIZE = [15:12]#,%	00000102 T	0019
STRUCT=[23:8]#,%	00000103 T	0019
INTEGER NUMTYPES;%	00000104 T	0019
PRT(44) = NUMTYPES		
%	00000105 T	0019
%*** PARAMETER TABLE ***	00000106 T	0019
ARRAY PARAMTAB[0:MAXPARAMS];%	00000107 T	0019
PRT(45) = PARAMTAB		
DEFINE PARAMNAME = [9:10]#,%	00000108 T	0021
PARAMKIND = [13:4]#,%	00000109 T	0021
PARAMLEVEL=[23:10]#,%	00000110 T	0021
PARAMTYPE = [33:10]#,%	00000111 T	0021
PARAMFILE = [34:1]#,%	00000112 T	0021
INTEGER NUMPARAMS;%	00000113 T	0021
PRT(46) = NUMPARAMS		
%	00000114 T	0021
%*** CONSTANT TABLE ***	00000115 T	0021
ARRAY CONSTTAB[0:MAXCONSTS];%	00000116 T	0021
PRT(47) = CONSTTAB		
INTEGER NUMCONSTS;%	00000117 T	0023
PRT(50) = NUMCONSTS		
%	00000118 T	0023
%*** LABEL TABLE ***	00000119 T	0023
ARRAY LABTAB[0:MAXLABS];%	00000120 T	0023
PRT(51) = LABTAB		
DEFINE LABVAL=[14:15]#,%	00000121 T	0025
LABDEF=[15:1]#,%	00000122 T	0025
INTEGER NUMLABS,FIRSTLAB;%	00000123 T	0025
PRT(52) = NUMLABS		
PRT(53) = FIRSTLAB		
%	00000124 T	0025
%*** TABLES FOR I/O AND CHARACTER HANDLING ***	00000125 T	0025
ARRAY CH[0:0], TEXT[0:1], STRING[0:11];%	00000126 T	0025
PRT(54) = CH		
PRT(55) = TEXT		
PRT(56) = STRING		
POINTER CHARPNT,TEXTPNT,TEXTPNT0,STRINGPNT;%	00000127 T	0030
PRT(57) = CHARPNT		
PRT(60) = TEXTPNT		

PRT(61) =	TEXTPNT0		
PRT(62) =	STRIN&PNT		
	ARRAY ICARD[0:9], LINE[0:16], XLINE[0:10], ALGOLCARD[0:9];%	00000128	T 0030
PRT(63) =	ICARD		
PRT(64) =	LINE		
PRT(65) =	XLINE		
PRT(66) =	ALGOLCARD		
	POINTER CARDPNT,LINEPNT,XLINEPNT,ALGOLPNT;%	00000129	T 0037
PRT(67) =	CARDPNT		
PRT(70) =	LINEPNT		
PRT(71) =	XLINEPNT		
PRT(72) =	ALGOLPNT		
	INTEGER CHARCNT,ALGOLCNT,MARGINCNT;%	00000130	T 0037
PRT(73) =	CHARCNT		
PRT(74) =	ALGO&CNT		
PRT(75) =	MARGINCNT		
	ARRAY HEADTEXT[0:10], ERRLINE[0:16];%	00000131	T 0037
PRT(76) =	HEADTEXT		
PRT(77) =	ERRLINE		
	INTEGER LINECNT,PAGECNT,ERRINX;%	00000132	T 0040
PRT(100) =	LINECNT		
PRT(101) =	PAGECNT		
PRT(102) =	ERRINX		
	%	00000133	T 0040
	*** XREF FILE AND TABLE ***	00000134	T 0040
	FILE XREFFILE DISK SERIAL [20:3000] (2,3,150);%	00000135	T 0040
PRT(103) =	XREFFILE		
	ARRAY BLOCKTAB[0:MAXTABLES], XREFLINE[0:16]; %	00000136	T 0044
PRT(104) =	BLOCKTAB		
PRT(105) =	XREFLINE		
	INTEGER NUMXREF,NUMBLOCKS; POINTER XREFPNT;%	00000137	T 0047
PRT(106) =	NUMXREF		
PRT(107) =	NUMBLOCKS		
PRT(110) =	XREFPNT		
	%	00000138	T 0047
	*** OTHER TABLES ***	00000139	T 0047
	INTEGER ARRAY VARLIST[0:LISTLENGTH]; % TEMPORARY LIST OF VARIABLES,	00000140	T 0047
PRT(111) =	VARLIST		
	INTEGER VARINDEX,FIRSTVAR;%	00000141	T 0049
PRT(112) =	VARINDEX		
PRT(113) =	FIRSTVAR		
	ARRAY SYMTAB[0:MAXSYMS]; % USED BY "EXPRESSION",	00000142	T 0049
PRT(114) =	SYMTAB		
	INTEGER NUMSYMS;%	00000143	T 0051
PRT(115) =	NUMSYMS		
	ARRAY WITHTAB[0:MAXWITHSYMS]; % USED BY "WITHSTAT",	00000144	T 0051
PRT(116) =	WITHTAB		
	INTEGER NWITHSYMS;%	00000145	T 0053
PRT(117) =	NWITHSYMS		
	INTEGER ARRAY SYMBOL[0:64]; % USED BY "INSYMBOL",	00000146	T 0053
PRT(120) =	SYMBOL		
	INTEGER ARRAY SYMKIND[0:61]; % USED IN ERROR RECOVERY,	00000147	T 0054
PRT(121) =	SYMKIND		
	ARRAY PNTRTAB1,PNTRTAB2,PNTRTAB3[0:MAXPNTRS];% USED FOR FORWARD POINTERS	00000148	T 0056
PRT(122) =	PNTRTAB1		
PRT(123) =	PNTRTAB2		
PRT(124) =	PNTRTAB3		
	INTEGER NumpNTRS;%	00000149	T 0058

PRT(125) = NUMPTRS	ARRAY EXTFILETAB[0:MAXEXTFILES];	% EXTERNAL FILES,	00000150 T	0058
PRT(126) = EXTFILETAB	INTEGER NUMEXTFILES;%		00000151 T	0060
PRT(127) = NUMEXTFILES	ARRAY FILETAB[0:MAXFILES];	% FILES IN USE,	00000152 T	0060
PRT(130) = FILETAB	INTEGER NUMFILES;%		00000153 T	0062
PRT(131) = NUMFILES	BOOLEAN ARRAY ERR[0:119];	% RECORDS ERROR MESSAGES,	00000154 T	0062
PRT(132) = ERR	%		00000155 T	0064
	*** COMPILE TIME OPTIONS ***		00000156 T	0064
	BOOLEAN LISTOPTION,RESWORDOPTION,CHECKOPTION,DUMPOPTION,XREFOPTION;%		00000157 T	0064
PRT(133) = LISTOPTION				
PRT(134) = RESWORDOPTION				
PRT(135) = CHECKOPTION				
PRT(136) = DUMPOPTION				
PRT(137) = XREFOPTION				
	INTEGER CARLENGTH;%		00000158 T	0064
PRT(140) = CARLENGTH	%		00000159 T	0064
	*** INTRINSIC TYPES ***		00000160 T	0064
	INTEGER INTTYPE,REALTYPE,ALFATYPE,CHARTYPE,BOOLTYPE,NILTYPE,TEXTTYPE,%		00000161 T	0064
PRT(141) = INTTYPE				
PRT(142) = REALTYPE				
PRT(143) = ALFATYPE				
PRT(144) = CHARTYPE				
PRT(145) = BOOLTYPE				
PRT(146) = NILTYPE				
PRT(147) = TEXTTYPE				
	INPUTFILE,OUTPUTFILE,EMPTYSET;%		00000162 T	0064
PRT(150) = INPUTFILE				
PRT(151) = OUTPUTFILE				
PRT(152) = EMPTYSET				
	BOOLEAN INPUTDECL,OUTPUTDECL;%		00000163 T	0064
PRT(153) = INPUTDECL				
PRT(154) = OUTPUTDECL				
	%		00000164 T	0064
	*** TEMPORARY VARIABLES ***		00000165 T	0064
	INTEGER T1,T2,T3,T4,T5;%		00000166 T	0064
PRT(155) = T1				
PRT(156) = T2				
PRT(157) = T3				
PRT(160) = T4				
PRT(161) = T5				
	%		00000167 T	0064
	*** OTHER VARIABLES ***		00000168 T	0064
	ALPHA USER;	% THE USER NUMBER FOUND ON THE USER CARD,	00000169 T	0064
PRT(162) = USER				
	%		00000170 T	0064
	INTEGER CURLEVEL,	% CURRENT PROCEDURE LEVEL,	00000171 T	0064
PRT(163) = CURLEVEL				
	TOPLEVEL,	% TOP LEVEL IN DISPLAY VECTOR,	00000172 T	0064
PRT(164) = TOPLEVEL				
	NUMBEGINS,	% NUMBER OF "BEGIN" S IN THE PROGRAM,	00000173 T	0064
PRT(165) = NUMBEGINS				

```

PRT(166) = NUMCASES      NUMCASES,          % NUMBER OF CASE-STATEMENTS IN PROGRAM,      00000174 T 0064
PRT(167) = NUMREPS      NUMREPS,           % NUMBER OF REPEAT-STATEMENTS IN PROGRAM,    00000175 T 0064
PRT(170) = NUMTEMPS    NUMTEMPS,         % NUMBER OF TEMPORARY VARIABLES IN USE,      00000176 T 0064
PRT(171) = CURFUNC     CURFUNC,          % INDEX OF FUNCTION CURRENTLY COMPILED,      00000177 T 0064
PRT(172) = CURSY       CURSY,            % LAST SYMBOL READ BY SCANNER,              00000178 T 0064
PRT(173) = CURTYPE     CURTYPE,         % TYPE OF ENTITY LAST COMPILED,            00000179 T 0064
PRT(174) = CURMODE     CURMODE,         % CURRENT EXPRESSION MODE,                 00000180 T 0064
PRT(175) = LASTREC     LASTREC,         % LAST RECORD TABLE DEFINED,             00000181 T 0064

```

```

%
LABEL ENDOFINPUT;%
%
FORMAT NOERRORS ("NO ERRORS DETECTED,");%
00000182 T 0064
00000183 T 0064
00000184 T 0064
00000185 T 0064

```

```
PRT(176) = NOERRORS
```

```

ERRORS (15," ERRORS DETECTED"/),%
ALIST ("S SET LIST SINGLE"),%
NOALIST ("S RESET LIST"),%
LASTLINE (") TERMINATE; END OF PASCAL PROGRAM,");%
TERMMESS ("**** END-OF-INPUT, COMPILATION TERMINATED,");%

```

```
START OF SEGMENT ***** 3
```

```

00000186 T 0064
00000187 T 0064
00000188 T 0064
00000189 T 0064
00000190 T 0064

```

```

MONITOR EXPOVRI=REALOVERFLOW;%
PRT(177) = REALOVERFLOW

```

```

%
**** SCANNER SYMBOLS ****
DEFINE IDENTIFIER=1#, INTCONST=2#, REALCONST=3#, ALFACONST=4#,%
CHARCONST=5#, NOTSY=6#, ASTERISK=7#, SLASH=8#,%
ANDSY=9#, DIVSY=10#, MODSY=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#, COMMA=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#, WHILSY=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#, PROCSY=58#, WITHSY=59#, PROGRAMSY=60#,%
PACKEDSY=61#;%

```

```
3 IS 46 LONG, NEXT SEG 2
```

```

00000191 T 0064
00000192 T 0066
00000193 T 0066
00000194 T 0066
00000195 T 0066
00000196 T 0066
00000197 T 0066
00000198 T 0066
00000199 T 0066
00000200 T 0066
00000201 T 0066
00000202 T 0066
00000203 T 0066
00000204 T 0066
00000205 T 0066
00000206 T 0066
00000207 T 0066
00000208 T 0066
00000209 T 0066
00000210 T 0066
00000211 T 0066
00000212 T 0066

```

```

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

```



```

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
%
%
%
%
%
%
%
%
%
%
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
%
%
PROCEDURE INSYMBOL; FORWARD;%
PRT(200) = INSYMBOL
PROCEDURE WRITEALGOL; FORWARD;%
PRT(201) = WRITEALGOL
PROCEDURE NEWXREF(NAME1,NAME2,TABLE,DECL);%
PRT(202) = NEWXREF
VALUE NAME1, NAME2, TABLE, DECL;%
REAL NAME1, NAME2;%
INTEGER TABLE;%
BOOLEAN DECL;%
FORWARD;%
%
DEFINE NDIGITS(N)=%
IF N≤ 9 THEN 1 ELSE%
IF N≤99 THEN 2 ELSE 3 DIGITS#;%
%
DEFINE HEADING=%
BEGIN COMMENT *** PRINTS A HEADING ON TOP OF A NEW PAGE, %
PAGECNT:=PAGECNT+1;%
REPLACE POINTER(HEADTEXT[*])+85 BY PAGECNT FOR NDIGITS(PAGECNT);%
WRITE(LINES[PAGE]);%
WRITE(LINES[DBL],11,HEADTEXT[*]);%
LINECNT:=2;%
END OF HEADING#;%
%
%
DEFINE PRINTLINE= %*** PRINTS A SOURCE CODE LINE,
BEGIN%
REPLACE LINEPNT=8 BY CARDNT FOR 5 DIGITS;%
IF LINECNT≥LINESPERPAGE THEN HEADING;%
IF RESWORDOPTION THEN%
BEGIN%
WRITE(LINES[NO],11,XLINE[*]);%
WRITE(LINES[NO],11,XLINE[*]);%
END;%
WRITE(LINES,17,LINE[*]);%
LINECNT:=LINECNT+1;%
END OF PRINTLINE#;%
%
%
DEFINE NEWCARD= %*** READS A NEW SOURCE CODE CARD,
BEGIN%
IF LISTOPTION THEN PRINTLINE;%
IF ERRINX>0 THEN PRINTERRORS;%
READ(CARD,10,ICARD[*]) [ENDOFINPUT];%

```

```

00000214 T 0066
00000215 T 0066
00000216 T 0066
00000217 T 0066
00000218 T 0066
00000219 T 0066
00000220 T 0066
00000221 T 0066
00000222 T 0066
00000223 T 0066
00000224 T 0066
00000225 T 0066
00000226 T 0066
00000227 T 0069
00000228 T 0069
00000229 T 0069
00000230 T 0069
00000231 T 0069
00000232 T 0069
00000233 T 0069
00000234 T 0069
00000235 T 0069
00000236 T 0069
00000237 T 0069
00000238 T 0069
00000239 T 0069
00000240 T 0069
00000241 T 0069
00000242 T 0069
00000243 T 0069
00000244 T 0069
00000245 T 0069
00000246 T 0069
00000247 T 0069
00000248 T 0069
00000249 T 0069
00000250 T 0069
00000251 T 0069
00000252 T 0069
00000253 T 0069
00000254 T 0069
00000255 T 0069
00000256 T 0069
00000257 T 0069
00000258 T 0069
00000259 T 0069
00000260 T 0069
00000261 T 0069
00000262 T 0069
00000263 T 0069
00000264 T 0069
00000265 T 0069
00000266 T 0069
00000267 T 0069

```

```

CARDPNT:=POINTER(ICARD[*]);%
REPLACE LINEPNT BY CARDPNT FOR 10 WORDS, " " FOR 6 WORDS;%
REPLACE XLINEPNT BY " " FOR 10 WORDS;%
CHARCNT:=CARDLENGTH;%
MARGINCNT:=85;%
CARDCNT:=CARDCNT+1;%
END#;%
%
%
DEFINE GEN(T,N,START)=      %*** GENERATE A TEXT "T", CONSISTING OF
BEGIN                        %*** "N" LETTERS, STARTING AT "START".
    IF ALGOLCNT<N THEN WRITEALGOL;%
    TEXT[0]:=T;%
    REPLACE ALGOLPNTIALGOLPNT 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 ALGOLPNTIALGOLPNT BY CHARPNT FOR 1, NUM FOR NDIG DIGITS;%
    ALGOLCNT:=ALGOLCNT+NDIG-1;%
END#;%
%
%
INTEGER NABS,NSIZE;%

PRT(203) = NABS
PRT(204) = NSIZE

%
DEFINE GENINT(N)=%
BEGIN%
    NABS:=ABS(N); IF N<0 THEN GEN("=",1,7);%
    NSIZE:=IF NABS< 9 THEN 1 ELSE%
            IF NABS< 999 THEN 3 ELSE%
            IF NABS< 99999 THEN 5 ELSE%
            IF NABS<999999999 THEN 8 ELSE 12;%
    IF ALGOLCNT<NSIZE THEN WRITEALGOL;%
    IF NSIZE=12 THEN%
        REPLACE ALGOLPNTIALGOLPNT BY (NABS DIV 1000000) FOR 6 DIGITS,%
                ENTIER(NABS MOD 1000000) FOR 6 DIGITS ELSE%
        REPLACE ALGOLPNTIALGOLPNT BY NABS FOR NSIZE DIGITS;%
    ALGOLCNT:=ALGOLCNT+NSIZE;%
END OF GENINT#;%
%
%
PROCEDURE GENREAL(X);%
PRT(205) = GENREAL
VALUE X; REAL X;%
BEGIN%
    REAL ABSX;%

STACK(F+2) = ABSX
                INTEGER POWER,V1,V2;%
STACK(F+3) = POWER
STACK(F+4) = V1

```

```

00000268 T 0069
00000269 T 0069
00000270 T 0069
00000271 T 0069
00000272 T 0069
00000273 T 0069
00000274 T 0069
00000275 T 0069
00000276 T 0069
00000277 T 0069
00000278 T 0069
00000279 T 0069
00000280 T 0069
00000281 T 0069
00000282 T 0069
00000283 T 0069
00000284 T 0069
00000285 T 0069
00000286 T 0069
00000287 T 0069
00000288 T 0069
00000289 T 0069
00000290 T 0069
00000291 T 0069
00000292 T 0069
00000293 T 0069
00000294 T 0069
00000295 T 0069

```

```

00000296 T 0069
00000297 T 0069
00000298 T 0069
00000299 T 0069
00000300 T 0069
00000301 T 0069
00000302 T 0069
00000303 T 0069
00000304 T 0069
00000305 T 0069
00000306 T 0069
00000307 T 0069
00000308 T 0069
00000309 T 0069
00000310 T 0069
00000311 T 0069
00000312 T 0069
00000313 T 0069

```

```

00000314 T 0069
00000315 T 0069
00000316 T 0069

```

START OF SEGMENT \*\*\*\*\*

```

00000317 T 0000

```

STACK(F+5) = V2

```
%
  IF X,[46:5]=0 THEN%
  BEGIN%
    IF ALGOLCNT<9 THEN WRITEALGOL;%
    TEXT[0]=X;%
    REPLACE ALGOLPNT:ALGOLPNT BY "", TEXTPNT FOR 7, "";%
PRT(206) = STRN6 REPLACE
PRT(207) = *DESTINATION POINTER*
    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>=7 DO BEGIN ABSX:=ABSX/10; POWER:=POWER+1 END;%
      WHILE ABSX<=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, ",",;%
      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;%
```

00000318 T 0000  
00000319 T 0000  
00000320 T 0001  
00000321 T 0001  
00000322 T 0003  
00000323 T 0004

00000324 T 0013  
00000325 T 0015  
00000326 T 0015  
00000327 T 0015  
00000328 T 0017  
00000329 T 0027  
00000330 T 0028  
00000331 T 0029  
00000332 T 0029  
00000333 T 0037  
00000334 T 0043  
00000335 T 0044  
00000336 T 0047  
00000337 T 0055  
00000338 T 0061  
00000339 T 0062  
00000340 T 0073  
00000341 T 0074  
00000342 T 0078  
00000343 T 0079  
00000344 T 0091  
00000345 T 0101  
00000346 T 0101

4 IS 105 LONG, NEXT SEG 2

```
%
%
PRT(210) = TYPEINDEX
%
  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;%
```

00000347 T 0069  
00000348 T 0069  
00000349 T 0069  
  
00000350 T 0069  
00000351 T 0069  
00000352 T 0069  
00000353 T 0069  
00000354 T 0069  
00000355 T 0069  
00000356 T 0069  
00000357 T 0069  
00000358 T 0069  
00000359 T 0069  
00000360 T 0069  
00000361 T 0075  
00000362 T 0079  
00000363 T 0084  
00000364 T 0086  
00000365 T 0090

%	00000366 T 0091
%	00000367 T 0091
DEFINE MARGIN(LETTER,NUM)=%	00000368 T 0091
BEGIN COMMENT *** PLACES INFORMATION IN THE MARGIN, %	00000369 T 0091
IF MARGINCNTS118 THEN%	00000370 T 0091
BEGIN TEXT(0)=LETTER;%	00000371 T 0091
REPLACE LINEPNT+MARGINCNT BY TEXTPNT+5 FOR 2,%	00000372 T 0091
NUM FOR NDIGITS(NUM);%	00000373 T 0091
MARGINCNT:=MARGINCNT+6;%	00000374 T 0091
END;%	00000375 T 0091
END OF MARGIN#;%	00000376 T 0091
%	00000377 T 0091
%	00000378 T 0091
PROCEDURE SKIP(SYMBOL);	00000379 T 0091
*** SKIP SYMBOLS TO RECOVER FROM ERROR	
VALUE SYMBOL; INTEGER SYMBOL; %*** CONDITION,	00000380 T 0091
BEGIN%	00000381 T 0091
WHILE CURSY#SYMBOL AND SYMKIND[CURSY]=MIDDLE DO%	00000382 T 0091
IF CURSY=RECORDSY THEN%	00000383 T 0093
BEGIN DO BEGIN INSYMBOL;%	00000384 T 0094
SKIP(99);%	00000385 T 0095
END UNTIL CURSY#SEMICOLON AND CURSY#CASESY;%	00000386 T 0096
END ELSE INSYMBOL;%	00000387 T 0098
END OF SKIP;%	00000388 T 0100
%	00000389 T 0100
%	00000390 T 0100
PROCEDURE ERROR(ERRNUM);%	00000391 T 0100
PRT(212) = ERROR	
VALUE ERRNUM; INTEGER ERRNUM;%	00000392 T 0100
BEGIN COMMENT *** ARRANGE ERROR INDICATOR, %	00000393 T 0100
NUMERRS:=NUMERRS+1;%	00000394 T 0100
ERR[ERRNUM]=TRUE;%	00000395 T 0102
ERRINX:=MAX(ERRINX,CARDLENGTH-2-CHARCNT);%	00000396 T 0103
IF ERRINX<115 THEN%	00000397 T 0107
BEGIN REPLACE POINTER(ERRLINE[1])+ERRINX BY "X",%	00000398 T 0108
PRT(213) = *DESTINATION POINTER*	
ERRNUM FOR NDIGITS(ERRNUM);%	00000399 T 0116
ERRINX:=ERRINX+(IF ERRNUM<9 THEN 2 ELSE%	00000400 T 0123
IF ERRNUM<99 THEN 3 ELSE 4);%	00000401 T 0126
END END OF ERROR;%	00000402 T 0129
%	00000403 T 0129
%	00000404 T 0129
PROCEDURE PRINTERRORS;%	00000405 T 0129
PRT(214) = PRINTERRORS	
BEGIN COMMENT *** PRINT ERROR INDICATORS, %	00000406 T 0129

```

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;%
PRT(215) = THISLEVEL
PRT(216) = THISTAB
PRT(217) = THISINDEX
ALPHA THISID,TNAME;%
PRT(220) = THISID
PRT(221) = TNAME
BOOLEAN FOUND;%
PRT(222) = 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".
    THISINDEX:=HASH(NAME1);%
    TNAME:=NAMETAB1[TAB,THISINDEX];%
    WHILE(IF TNAME=NAME1 THEN NAMETAB2[TAB,THISINDEX]#NAME2%

```

```

00000407 T 0129
00000408 T 0174
00000409 T 0178
00000410 T 0179
00000411 T 0185
00000412 T 0186

```

```

00000413 T 0186
00000414 T 0186
00000415 T 0186
00000416 T 0186
00000417 T 0186

```

```
00000418 T 0186
```

```
00000419 T 0186
```

```

00000420 T 0186
00000421 T 0186
00000422 T 0186
00000423 T 0186
00000424 T 0186
00000425 T 0186
00000426 T 0186
00000427 T 0186
00000428 T 0186
00000429 T 0186
00000430 T 0186
00000431 T 0186
00000432 T 0186
00000433 T 0186
00000434 T 0186
00000435 T 0186
00000436 T 0186
00000437 T 0186
00000438 T 0186
00000439 T 0186
00000440 T 0186
00000441 T 0186
00000442 T 0186
00000443 T 0186
00000444 T 0186
00000445 T 0186
00000446 T 0186
00000447 T 0186
00000448 T 0186
00000449 T 0186
00000450 T 0186
00000451 T 0186
00000452 T 0186
00000453 T 0186

```

```

ELSE TNAME#0) DO%
BEGIN%
  THISINDEX:=IF THISINDEX#0 THEN MAXNAMES ELSE THISINDEX=1;%
  TNAME1=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;
PRT(223) = INITIALIZE
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,LBSSY,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;%
PRT(224) = *STRING POINTER*
PRT(225) = *DESTINATION POINTER*
  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, ")/B=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;%

```

```

00000454 T 0186
00000455 T 0186
00000456 T 0186
00000457 T 0186
00000458 T 0186
00000459 T 0186
00000460 T 0186
00000461 T 0186
00000462 T 0186
00000463 T 0186
00000464 T 0186
00000465 T 0186
00000466 T 0186
00000467 T 0186
00000468 T 0186
START OF SEGMENT ***** 5
00000469 T 0000
00000470 T 0000
START OF SEGMENT ***** 6
00000471 T 0001
00000472 T 0001
00000473 T 0001
6 IS 62 LONG, NEXT SEG 5
00000474 T 0001
00000475 T 0001
START OF SEGMENT ***** 7
00000476 T 0003
00000477 T 0003
00000478 T 0003
7 IS 65 LONG, NEXT SEG 5
00000479 T 0003
00000480 T 0003
00000481 T 0006
00000482 T 0009
00000483 T 0018
00000484 T 0024
00000485 T 0033
00000486 T 0036
00000487 T 0040
00000488 T 0043
00000489 T 0049
00000490 T 0053
00000491 T 0055
00000492 T 0064
00000493 T 0075
00000494 T 0076
00000495 T 0087
00000496 T 0098
00000497 T 0100
00000498 T 0111
00000499 T 0115

```

```

%
%*** INITIALIZE INTRINSIC TYPES, CONSTANTS ETC. ***
%
INTTYPE:=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;%
REALTYPE:=T3:=2;                                     %*** "REAL" ***
T1,FORM:=FLOATING; TYPETAB1[2]:=T1;%
NEWNAME("400REAL",0,0); T3,IDCLASS:=TYPES;%
NAMETAB3[0,THISINDEX]:=T3;%
ALFATYPE:=T3:=3;                                     %*** "ALFA" ***
T1,FORM:=ALFA; TYPETAB1[3]:=T1;%
NEWNAME("400ALFA",0,0); T3,IDCLASS:=TYPES;%
NAMETAB3[0,THISINDEX]:=T3;%
BOOLTYPE:=T3:=4;                                     %*** "BOOLEAN" ***
T1,FORM:=SYMBOLIC; TYPETAB1[4]:=T1; TYPETAB3[4]:=1;%
NEWNAME("7BOOLEA","N",0); T3,IDCLASS:=TYPES;%
NAMETAB3[0,THISINDEX]:=T3;%
CHARTYPE:=T3:=5;                                     %*** "CHAR" ***
T1,FORM:=CHAR; TYPETAB1[5]:=T1; TYPETAB3[5]:=63;%
NEWNAME("400CHAR",0,0); T3,IDCLASS:=TYPES;%
NAMETAB3[0,THISINDEX]:=T3;%
T3:=BOOLTYPE; T3,IDCLASS:=CONST; %*** "FALSE" ***
NEWNAME("50FALSE",0,0); NAMETAB3[0,THISINDEX]:=T3;%
T3,INFO:=1; %*** "TRUE" ***
NEWNAME("400TRUE",0,0); NAMETAB3[0,THISINDEX]:=T3;%
NUMTYPES:=5;%
NILTYPE:=1; %*** TYPE OF "NIL" ***
EMPTYSET:=2; %*** TYPE OF [] ***
NEWNAME("6MAXINT",0,0); T3:=INTTYPE; %*** "MAXINT" ***
T3,IDCLASS:=CONST; T3,INFO:=1024;%
NAMETAB3[0,THISINDEX]:=T3;%
NUMCONSTS:=1; CONSTTAB[1]:=MAXINT;%
%
T3:=0; T3,IDCLASS:=PROC; %*** PROCEDURES ***
FOR A1="3000GET", "3000NEW", "400PACK", "400PAGE", "3000PUT",%
"400READ", "6READLN", "50RESET", "6UNPACK", "50WRITE" DO%
BEGIN%
STACK(F+5) = *TEMPORARY STORAGE*
NEWNAME(A,0,0); NAMETAB3[0,THISINDEX]:=T3;%
END;%
NEWNAME("7DISPOS","E",0); NAMETAB3[0,THISINDEX]:=T3;%
NEWNAME("7REWRIT","E",0); NAMETAB3[0,THISINDEX]:=T3;%
NEWNAME("7WRITEL","N",0); NAMETAB3[0,THISINDEX]:=T3;%
%
T3,IDCLASS:=FUNC; %*** FUNCTIONS ***
FOR A1="3000ABS", "6ARCTAN", "3000CHR", "3000COS", "3000EOF",%
"400EOLN", "3000EXP", "20000LN", "30000DD", "400PRED",%
"400SUCC", "50ROUND", "3000SIN", "3000SQR", "400SQRT",%
"50TRUNC", "6CONCAT", "400TIME", "400DATE", "610TIME",%
"400USER", "3000ORD"%
DO BEGIN%
STACK(F+6) = *TEMPORARY STORAGE*
NEWNAME(A,0,0); NAMETAB3[0,THISINDEX]:=T3;%
END;%

```

```

00000500 T 0136
00000501 T 0136
00000502 T 0136
00000503 T 0136
00000504 T 0138
00000505 T 0142
00000506 T 0146
00000507 T 0177
00000508 T 0179
00000509 T 0180
00000510 T 0183
00000511 T 0210
00000512 T 0212
00000513 T 0213
00000514 T 0216
00000515 T 0243
00000516 T 0245
00000517 T 0246
00000518 T 0250
00000519 T 0277
00000520 T 0279
00000521 T 0280
00000522 T 0284
00000523 T 0311
00000524 T 0313
00000525 T 0315
00000526 T 0342
00000527 T 0344
00000528 T 0371
00000529 T 0372
00000530 T 0373
00000531 T 0374
00000532 T 0400
00000533 T 0403
00000534 T 0405
00000535 T 0407
00000536 T 0407
00000537 T 0410
00000538 T 0420
00000539 T 0430
00000540 T 0430
00000541 T 0468
00000542 T 0468
00000543 T 0495
00000544 T 0522
00000545 T 0549
00000546 T 0549
00000547 T 0551
00000548 T 0561
00000549 T 0571
00000550 T 0581
00000551 T 0591
00000552 T 0594
00000553 T 0595
00000554 T 0643

```

```

NEWNAME("7ELAPSE","D",0); NAMETAB3[0,THISINDEX]:=T3;%
NEWNAME("7WEEKDA","Y",0); NAMETAB3[0,THISINDEX]:=T3;%
%
TEXTTYPE:=T3:=NUMTYPES:=NUMTYPES+1; %*** "TEXT" ***
T1 := TEXTFILE; T1,STRUCT := 1; TYPETAB1[TEXTTYPE] := T1; %
T3, IDCLASS := TYPES; %
NEWNAME("400TEXT",0,0); NAMETAB3[0,THISINDEX]:=T3;%
T3:=TEXTTYPE; T3, IDCLASS:=VAR; %*** "INPUT" ***
T3, EXTERNALFILE:=1;%
NEWNAME("50INPUT",0,0); INPUTFILE:=THISINDEX;%
NAMETAB3[0,THISINDEX]:=T3;%
NEWNAME("60OUTPUT",0,0); %*** "OUTPUT" ***
NAMETAB3[0,THISINDEX]:=T3; OUTPUTFILE:=THISINDEX;%
END OF INTIALIZE;%

```

```

00000555 T 0643
00000556 T 0670
00000557 T 0697
00000558 T 0697
00000559 T 0699
00000560 T 0703
00000561 T 0705
00000562 T 0732
00000563 T 0734
00000564 T 0736
00000565 T 0762
00000566 T 0764
00000567 T 0789
00000568 T 0792

```

5 IS 797 LONG, NEXT SEG 2

```

%
%
%
%*** XREF ROUTINES ***
%*****
%
DEFINE XREFCARD=[16:17]#,%
XREFBLOCK=[26:10]#;%
REAL A0,B0,A1,B1,LASTA0,LASTA1;%

```

```

00000569 T 0186
00000570 T 0186
00000571 T 0186
00000572 T 0186
00000573 T 0186
00000574 T 0186
00000575 T 0186
00000576 T 0186
00000577 T 0186

```

```

PRT(226) = A0
PRT(227) = B0
PRT(230) = A1
PRT(231) = B1
PRT(232) = LASTA0
PRT(233) = LASTA1
INTEGER NL, LASTBLOCK, A2, AX;%
PRT(234) = NL
PRT(235) = LASTBLOCK
PRT(236) = A2
PRT(237) = AX

```

```

00000578 T 0186

```

```

%
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[41:41:16]&NAME1[35:6×NL-1:6×NL];%

```

```

00000579 T 0186
00000580 T 0186
00000581 T 0186
00000582 T 0186
00000583 T 0186
00000584 T 0186
00000585 T 0186
00000586 T 0186
00000587 T 0188

```

```

PRT(240) = DYNAMIC DIALS
ELSE NAME2:=0&NAME2[35:6×(NL-6)-1:6×(NL-6)];%
AX:=CARDNT; AX,XREFBLOCK:=BLOCKTAB[TABLE];%
IF DECL THEN AX:=AX-1000000000000;%
WRITE(XREFFILE,*,NAME1,NAME2,AX);%
PRT(241) = *LIST, LABEL, OR SEGMENT DESCRIPTOR*
PRT(242) = OUTPUT(W)
END OF NEWXREF;%

```

```

00000588 T 0195
00000589 T 0202
00000590 T 0204
00000591 T 0206

```

```

00000592 T 0218

```



```

%
PROCEDURE XREFMAX(A);%
PRT(243) = XREFMAX
ARRAY A[0];%
BEGIN%
  A[0]="AZZZZZ"; A[1]="ZZZZZ"; A[2]=9999999999;%
END OF XREFMAX;%

```

```

00000593 T 0218
00000594 T 0218
00000595 T 0218
00000596 T 0218
00000597 T 0218
00000598 T 0222

```

```

%
%
BOOLEAN PROCEDURE XREFCOMPARE(A,B);%
PRT(244) = XREFCOMPARE
ARRAY A,B[0];%
BEGIN%
  A0:=A[0]; B0:=B[0]; A1:=A[1]; B1:=B[1];%
  XREFCOMPARE1=%
  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;%

```

```

00000599 T 0226
00000600 T 0226
00000601 T 0226
00000602 T 0226
00000603 T 0226
00000604 T 0226
00000605 T 0230
00000606 T 0230
00000607 T 0234
00000608 T 0237
00000609 T 0240
00000610 T 0242

```

```

%
%
PROCEDURE PRINTXREF(FINIS,A);%
PRT(245) = PRINTXREF
VALUE FINIS; BOOLEAN FINIS;%
ARRAY A[0];%
BEGIN%
  IF FINIS THEN%
  BEGIN%
    WRITE(LINES,17,XREFLINE[*]);%
    CLOSE(LINES);%
    CLOSE(XREFFILE);%
  END%
  ELSE%
  BEGIN%
    A0:=A[0]; A1:=A[1]; A2:=A[2];%
    IF A0=LASTA0 AND A1=LASTA1 AND A2,XREFBLOCK=LASTBLOCK THEN%
    BEGIN%
      IF NUMXREF=15 THEN%
      BEGIN%
        WRITE(LINES,17,XREFLINE[*]); LINECNT:=LINECNT+1;%
        IF LINECNT>LINESPERPAGE THEN HEADING;%
        XREFPNT:=POINTER(XREFLINE[*]); NUMXREF:=0;%
        REPLACE XREFPNT BY " " FOR 17 WORDS; XREFPNT:=XREFPNT+24;%
      END;%
      REPLACE XREFPNT BY A2,XREFCARD FOR 5 DIGITS;%
      XREFPNT:=XREFPNT+7; NUMXREF:=NUMXREF+1;%
    END%
  END%

```

```

00000611 T 0245
00000612 T 0245
00000613 T 0245
00000614 T 0245
00000615 T 0245
00000616 T 0245
00000617 T 0245
00000618 T 0245
00000619 T 0245
00000620 T 0250
00000621 T 0251
00000622 T 0253
00000623 T 0253
00000624 T 0253
00000625 T 0254
00000626 T 0257
00000627 T 0260
00000628 T 0260
00000629 T 0261
00000630 T 0262
00000631 T 0267
00000632 T 0289
00000633 T 0292
00000634 T 0298
00000635 T 0298
00000636 T 0303

```

```

END ELSE%
IF A2<0 THEN%
BEGIN%
  A2:=A2+1000000000000;%
  WRITE(LINES,17,XREFLINE[*]); LINECNT:=LINECNT+1;%
  IF LINECNT>LINESPERPAGE THEN HEADING;%
  XREFPNT:=POINTER(XREFLINE[*]); NUMXREF:=0;%
  REPLACE XREFPNT BY " " FOR 17 WORDS;%
  TEXT[0]:=A0,[35:36]; LASTA0:=A0;%
  REPLACE XREFPNT BY TEXTPNT+1 FOR A0,NAMELENGTH;%
  TEXT[0]:=LASTA1:=A1;%
  IF A0,NAMELENGTH>6 THEN%
  REPLACE XREFPNT+6 BY TEXTPNT+1 FOR A0,NAMELENGTH-6;%
  REPLACE XREFPNT+17 BY A2,XREFCARD FOR 5 DIGITS;%
  XREFPNT:=XREFPNT+24; LASTBLOCK:=A2,XREFBLOCK;%
END;%
END;%
END OF PRINTXREF;%

```

```

00000637 T 0307
00000638 T 0307
00000639 T 0308
00000640 T 0309
00000641 T 0310
00000642 T 0315
00000643 T 0338
00000644 T 0340
00000645 T 0344
00000646 T 0346
00000647 T 0352
00000648 T 0353
00000649 T 0355
00000650 T 0363
00000651 T 0369
00000652 T 0373
00000653 T 0373
00000654 T 0373

```

```

PRT(246) = TT1
PRT(247) = TT2
PRT(250) = F1
PRT(251) = F2
PRT(252) = LT
PRT(253) = RT

```

```

%
%
%
INTEGER TT1,TT2,F1,F2,LT,RT;%

```

```

00000655 T 0375
00000656 T 0375
00000657 T 0375
00000658 T 0375

```

```

%
DEFINE CHECKTYPES(LEFTTYPE,RIGHTTYPE)=%
BEGIN%
  IF LEFTTYPE>0 AND RIGHTTYPE=0 THEN%
  IF LEFTTYPE#RIGHTTYPE THEN%
  BEGIN%
    LT:=LEFTTYPE; RT:=RIGHTTYPE;%
    TT1:=TYPETAB1[LT]; TT2:=TYPETAB1[RT];%
    F1:=TT1,FORM; F2:=TT2,FORM;%
    IF LT#REALTYPE OR F2#NUMERIC THEN%
    IF(F1#SET AND LT#EMPTYSET)OR(F2#SET AND RT#EMPTYSET)THEN%
    IF(F1#POINTERS AND LT#NILTYPE)OR(F2#POINTERS AND RT#NILTYPE)THEN%
    BEGIN%
      IF F1=SET AND F2=SET THEN%
      BEGIN%
        LT:=TT1,SETTYPE; RT:=TT2,SETTYPE;%
        TT1:=TYPETAB1[LT]; TT2:=TYPETAB1[RT];%
        F1:=TT1,FORM; F2:=TT2,FORM;%
      END;%
      IF F1=POINTERS AND F2=POINTERS THEN%
      BEGIN%
        LT:=TT1,POINTTYPE; RT:=TT2,POINTTYPE;%
        TT1:=TYPETAB1[LT]; TT2:=TYPETAB1[RT];%
        F1:=TT1,FORM; F2:=TT2,FORM;%
      END;%
    END;%
  END;%

```

```

00000659 T 0375
00000660 T 0375
00000661 T 0375
00000662 T 0375
00000663 T 0375
00000664 T 0375
00000665 T 0375
00000666 T 0375
00000667 T 0375
00000668 T 0375
00000669 T 0375
00000670 T 0375
00000671 T 0375
00000672 T 0375
00000673 T 0375
00000674 T 0375
00000675 T 0375
00000676 T 0375
00000677 T 0375
00000678 T 0375
00000679 T 0375
00000680 T 0375
00000681 T 0375
00000682 T 0375
00000683 T 0375

```

```

        WHILE F1=SUBTYPE D0%
        BEGIN LT1=TT1,MAINTYPE; TT1:=TYPETAB1[LT]; F1:=TT1,FORM END;%
        WHILE F2=SUBTYPE D0%
        BEGIN RT1=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;%
PRT(254) = FILENAME
BOOLEAN LPARFOUND;%
PRT(255) = LPARFOUND
%
DEFINE FILEPARAM(DEFAULTFILE)=%*** CHECKS THE FIRST PARAMETER TO SEE
BEGIN
    %*** IF IT IS A FILE.
    INSYMBOL; FILENAME:=CURTYPE;CURSY:=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 TFORM;%
PRT(256) = TFORM
BOOLEAN SIGNED,NEGATIVE;%
PRT(257) = SIGNED
PRT(260) = NEGATIVE
%
DEFINE CONSTANT(CVAL,CTYPE)=
BEGIN
    %*** <CONSTANT> ***
    %*****
    IF CURSY=MINUS OR CURSY=PLUS THEN%
    BEGIN SIGNED:=TRUE; NEGATIVE:=CURSY=MINUS;%
    INSYMBOL;%

```

```

00000684 T 0375
00000685 T 0375
00000686 T 0375
00000687 T 0375
00000688 T 0375
00000689 T 0375
00000690 T 0375
00000691 T 0375
00000692 T 0375
00000693 T 0375
00000694 T 0375
00000695 T 0375
00000696 T 0375
00000697 T 0375
00000698 T 0375
00000699 T 0375
00000700 T 0375
00000701 T 0375
00000702 T 0375
00000703 T 0375
00000704 T 0375
00000705 T 0375
00000706 T 0375
00000707 T 0375
00000708 T 0375
00000709 T 0375
00000710 T 0375
00000711 T 0375
00000712 T 0375
00000713 T 0375
00000714 T 0375
00000715 T 0375
00000716 T 0375
00000717 T 0375
00000718 T 0375
00000719 T 0375
00000720 T 0375
00000721 T 0375
00000722 T 0375
00000723 T 0375
00000724 T 0375
00000725 T 0375
00000726 T 0375
00000727 T 0375
00000728 T 0375
00000729 T 0375
00000730 T 0375
00000731 T 0375
00000732 T 0375
00000733 T 0375
00000734 T 0375
00000735 T 0375

```

```

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 CURLNGTH>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(1); CVAL:=CTYPE:=0 END;%
END ELSE BEGIN ERROR(32); CVAL:=CTYPE:=0 END;%
INSYMBOL;%
END OF CONSTANT#;%

```

```

00000736 T 0375
00000737 T 0375
00000738 T 0375
00000739 T 0375
00000740 T 0375
00000741 T 0375
00000742 T 0375
00000743 T 0375
00000744 T 0375
00000745 T 0375
00000746 T 0375
00000747 T 0375
00000748 T 0375
00000749 T 0375
00000750 T 0375
00000751 T 0375
00000752 T 0375
00000753 T 0375
00000754 T 0375
00000755 T 0375
00000756 T 0375
00000757 T 0375
00000758 T 0375
00000759 T 0375
00000760 T 0375
00000761 T 0375
00000762 T 0375
00000763 T 0375
00000764 T 0375
00000765 T 0375
00000766 T 0375
00000767 T 0375
00000768 T 0375
00000769 T 0375
00000770 T 0375
00000771 T 0375
00000772 T 0375
00000773 T 0375
00000774 T 0375
00000775 T 0375
00000776 T 0375
00000777 T 0375

```

PART 3: THE SCANNER,  
-----

SYMBOL	INTERNAL NUMBER	INTERNAL NAME	SYMBOL KIND	INTERNAL NUMBER	INTERNAL NAME	SYMBOL KIND	INTERNAL NUMBER	INTERNAL NAME	SYMBOL KIND
IDENTIFIER	1	IDENTIFIER	MIDDLE	00000779	T	0375			
122	2	INTCONST	MIDDLE	00000780	T	0375			
2,5	3	REALCONST	MIDDLE	00000781	T	0375			
"ABCD"	4	ALFACONST	MIDDLE	00000782	T	0375			
"C"	5	CHARCONST	MIDDLE	00000783	T	0375			
NOT	6	NOTSY	MIDDLE	00000784	T	0375			
*	7	ASTERISK	MIDDLE	00000785	T	0375			
/	8	SLASH	MIDDLE	00000786	T	0375			
& AND	9	ANDSY	MIDDLE	00000787	T	0375			
DIV	10	DIVSY	MIDDLE	00000788	T	0375			
MOD	11	MODSY	MIDDLE	00000789	T	0375			
+	12	PLUS	MIDDLE	00000790	T	0375			
-	13	MINUS	MIDDLE	00000791	T	0375			
OR	14	ORSY	MIDDLE	00000792	T	0375			
< LSS	15	LSSSY	MIDDLE	00000793	T	0375			
<= LEQ	16	LEQSY	MIDDLE	00000794	T	0375			
>= GEQ	17	GEQSY	MIDDLE	00000795	T	0375			
> GTR	18	GTRSY	MIDDLE	00000796	T	0375			
<> NEQ	19	NEQSY	MIDDLE	00000797	T	0375			
= EQL	20	EQLSY	MIDDLE	00000798	T	0375			
IN	21	INSY	MIDDLE	00000799	T	0375			
(	22	LPAR	MIDDLE	0000800	T	0375			
)	23	RPAR	MIDDLE	0000801	T	0375			
[	24	LBRACKET	MIDDLE	0000802	T	0375			
]	25	RBRACKET	MIDDLE	0000803	T	0375			
..	26	DOUBLEDOT	MIDDLE	0000804	T	0375			
,	27	COMMA	MIDDLE	0000805	T	0375			
;	28	SEMICOLON	TERMINAL	0000806	T	0375			
.	29	DOT	MIDDLE	0000807	T	0375			
← @	30	ARROW	MIDDLE	0000808	T	0375			
	31	COLON	MIDDLE	0000809	T	0375			
=	32	ASSIGNSY	MIDDLE	0000810	T	0375			
BEGIN	33	BEGINSY	INITIAL	0000811	T	0375			
END	34	ENDSY	TERMINAL	0000812	T	0375			
IF	35	IFSY	INITIAL	0000813	T	0375			
THEN	36	THENSY	MIDDLE	0000814	T	0375			
ELSE	37	ELSESY	TERMINAL	0000815	T	0375			
CASE	38	CASESY	INITIAL	0000816	T	0375			
OF	39	OFSY	MIDDLE	0000817	T	0375			
REPEAT	40	REPEATSY	INITIAL	0000818	T	0375			
UNTIL	41	UNTILSY	TERMINAL	0000819	T	0375			
WHILE	42	WHILESY	INITIAL	0000820	T	0375			
DO	43	DOSY	MIDDLE	0000821	T	0375			
				0000822	T	0375			
				0000823	T	0375			
				0000824	T	0375			
				0000825	T	0375			
				0000826	T	0375			
				0000827	T	0375			
				0000828	T	0375			
				0000829	T	0375			
				0000830	T	0375			
				0000831	T	0375			
				0000832	T	0375			
				0000833	T	0375			
				0000834	T	0375			
				0000835	T	0375			

%	FOR	44	FORSY	INITIAL	00000836	T	0375
%	TO	45	TOSY	MIDDLE	00000837	T	0375
%	DOWNT	46	DOWNTOSY	MIDDLE	00000838	T	0375
%	GOTO	47	GOTOSY	INITIAL	00000839	T	0375
%	NIL	48	NILSY	MIDDLE	00000840	T	0375
%	TYPE	49	TYPESY	INITIAL	00000841	T	0375
%	ARRAY	50	ARRAYSY	MIDDLE	00000842	T	0375
%	RECORD	51	RECORDSY	MIDDLE	00000843	T	0375
%	FILE	52	FILESY	MIDDLE	00000844	T	0375
%	SET	53	SETSY	MIDDLE	00000845	T	0375
%	CONST	54	CONSTSY	INITIAL	00000846	T	0375
%	VAR	55	VARSY	INITIAL	00000847	T	0375
%	LABEL	56	LABELSY	INITIAL	00000848	T	0375
%	FUNCTION	57	FUNCSY	INITIAL	00000849	T	0375
%	PROCEDURE	58	PROCSY	INITIAL	00000850	T	0375
%	WITH	59	WITHSY	INITIAL	00000851	T	0375
%	PROGRAM	60	PROGRAMSY	INITIAL	00000852	T	0375
%	PACKED	61	PACKEDSY	MIDDLE	00000853	T	0375
%					00000854	T	0375
%					00000855	T	0375
	DEFINE BLANK=48#				00000856	T	0375
	EQUAL=61#				00000857	T	0375
	QUOTES=63#				00000858	T	0375
	DOLLAR=42#				00000859	T	0375
	LETTER(C)=(17<C AND C<=25)OR(33<C AND C<=41)OR(50<C AND C<=57)#				00000860	T	0375
	ALFANUM(C)=(LETTER(C) OR C<=9)#						
%							
	REAL CURVAL						
PRT(261)	= CURVAL						
	ALPHA CURNAME1,CURNAME2,C,CX				00000861	T	0375
PRT(262)	= CURNAME1						
PRT(263)	= CURNAME2						
PRT(264)	= C						
PRT(265)	= CX						
	INTEGER CURLLENGTH, LASTCHARPOS				00000862	T	0375
PRT(266)	= CURLLENGTH						
PRT(267)	= LASTCHARPOS						
	BOOLEAN FINIS				00000863	T	0375
PRT(270)	= FINIS						
%					00000864	T	0375
	DEFINE NEXTCHAR=				00000865	T	0375
	BEGIN COMMENT *** READ NEXT CHARACTER, ***				00000866	T	0375
	IF CHARCNT=0 THEN C:=BLANK ELSE				00000867	T	0375
	BEGIN				00000868	T	0375
	REPLACE CHARPNT BY CARDPNT/CARDPNT FOR 1				00000869	T	0375
	C:=CH[0]; CHARCNT:=CHARCNT-1				00000870	T	0375
	END END #				00000871	T	0375
%					00000872	T	0375
%					00000873	T	0375
%					00000874	T	0375
	PROCEDURE INSYMBOL				00000875	T	0375
	BEGIN COMMENT *** READS THE NEXT SYMBOL, ***				00000876	T	0375
	INTEGER SCALE,EXP				00000877	T	0375
	STACK(F+2) = SCALE						
	STACK(F+3) = EXP						
	BOOLEAN NEGEXP				00000878	T	0000
	STACK(F+4) = NEGEXP						
	LABEL START,OVERFLOW				00000879	T	0000
%					00000880	T	0000

START OF SEGMENT \*\*\*\*\*

8

START;%	00000881 T	0000
IF C=BLANK THEN%	00000882 T	0000
BEGIN SCAN CARDPNT;CARDPNT FOR CHARCNT;CHARCNT WHILE =" "%;	00000883 T	0000
PRT(271) = STRNG SCAN		
IF CHARCNT=0 THEN BEGIN NEWCARD; GO TO START END;%	00000884 T	0003
PRT(272) = ENDOP INPUT		
PRT(273) = GO TO SOLVER		
PRT(274) = *DESTINATION POINTER*		
NEXTCHAR;%	00000885 T	0070
END;%	00000886 T	0078
IF LETTER(C) THEN%	00000887 T	0078
BEGIN%	00000888 T	0083
CURLNGTH:=1; CURNAME1:=C; CURNAME2:=0;%	00000889 T	0084
NEXTCHAR;%	00000890 T	0086
WHILE ALFANUM(C) AND CURLNGTH<6 DO%	00000891 T	0094
BEGIN CURNAME1:=C&CURNAME1[35:29:30];%	00000892 T	0102
CURLNGTH:=CURLNGTH+1; NEXTCHAR;%	00000893 T	0104
END;%	00000894 T	0112
IF CURLNGTH=6 THEN%	00000895 T	0113
BEGIN%	00000896 T	0114
WHILE ALFANUM(C) AND CURLNGTH<12 DO%	00000897 T	0114
BEGIN CURNAME2:=C&CURNAME2[35:29:30];%	00000898 T	0123
CURLNGTH:=CURLNGTH+1; NEXTCHAR;%	00000899 T	0125
END;%	00000900 T	0133
WHILE ALFANUM(C) DO NEXTCHAR;%	00000901 T	0134
END;%	00000902 T	0150
CURNAME1,NAMELENGTH:=CURLNGTH;%	00000903 T	0150
CASE CURLNGTH OF%	00000904 T	0152
BEGIN %;	00000905 T	0152
PRT(275) = *CASE STATEMENT DESCRIPTOR*		
CURSY:=IDENTIFIER;%	00000906 T	0152
CURSY:=IF CURNAME1=="20000IF" THEN IFSY ELSE%	00000907 T	0154
IF CURNAME1=="20000DO" THEN DOSY ELSE%	00000908 T	0156
IF CURNAME1=="20000TO" THEN TOSY ELSE%	00000909 T	0158
IF CURNAME1=="20000OR" THEN ORSY ELSE%	00000910 T	0160
IF CURNAME1=="20000OF" THEN OFSY ELSE%	00000911 T	0162
IF CURNAME1=="20000IN" THEN INSY ELSE IDENTIFIER;%	00000912 T	0164
CURSY:=IF CURNAME1=="3000END" THEN ENDSY ELSE%	00000913 T	0167
IF CURNAME1=="3000FOR" THEN FORSY ELSE%	00000914 T	0169
IF CURNAME1=="3000DIV" THEN DIVSY ELSE%	00000915 T	0171
IF CURNAME1=="3000MOD" THEN MODSY ELSE%	00000916 T	0173
IF CURNAME1=="3000NIL" THEN NILSY ELSE%	00000917 T	0175
IF CURNAME1=="3000AND" THEN ANDSY ELSE%	00000918 T	0177
IF CURNAME1=="3000NOT" THEN NOTSY ELSE%	00000919 T	0179
IF CURNAME1=="3000VAR" THEN VARSY ELSE%	00000920 T	0181
IF CURNAME1=="3000SET" THEN SETSY ELSE%	00000921 T	0183
IF CURNAME1=="3000LSS" THEN LSSSY ELSE%	00000922 T	0185
IF CURNAME1=="3000LEQ" THEN LEQSY ELSE%	00000923 T	0187
IF CURNAME1=="3000GEQ" THEN GEQSY ELSE%	00000924 T	0189
IF CURNAME1=="3000GTR" THEN GTRSY ELSE%	00000925 T	0191
IF CURNAME1=="3000NEQ" THEN NEQSY ELSE%	00000926 T	0193
IF CURNAME1=="3000EQL" THEN EQLSY ELSE IDENTIFIER;%	00000927 T	0195
CURSY:=IF CURNAME1=="400THEN" THEN THENSY ELSE%	00000928 T	0198
IF CURNAME1=="400ELSE" THEN ELSESY ELSE%	00000929 T	0200
IF CURNAME1=="400WITH" THEN WITHSY ELSE%	00000930 T	0202
IF CURNAME1=="400CASE" THEN CASESY ELSE%	00000931 T	0204
IF CURNAME1=="400GOTO" THEN GOTOSY ELSE%	00000932 T	0206

```

IF CURNAME1="400TYPE" THEN TYPESY ELSE%
IF CURNAME1="400FILE" THEN FILESY ELSE IDENTIFIER;%
CURSY:=IF CURNAME1="50BEGIN" THEN BEGINSY ELSE%
IF CURNAME1="50WHILE" THEN WHILESY ELSE%
IF CURNAME1="50UNTIL" THEN UNTILSY ELSE%
IF CURNAME1="50ARRAY" THEN ARRAYSY ELSE%
IF CURNAME1="50CONST" THEN CONSTSY ELSE%
IF CURNAME1="50LABEL" THEN LABELSY ELSE IDENTIFIER;%
CURSY:=IF CURNAME1="6REPEAT" THEN REPEATSY ELSE%
IF CURNAME1="6DOWNT0" THEN DOWNTOSY ELSE%
IF CURNAME1="6RECORD" THEN RECORDSY ELSE%
IF CURNAME1="6PACKED" THEN PACKEDSY ELSE IDENTIFIER;%
CURSY:=IF CURNAME1="7PROGRA" AND CURNAME2="M" THEN PROGRAMSY%
ELSE IDENTIFIER;%
CURSY:=IF CURNAME1="8FUNCTI" AND CURNAME2="ON" THEN FUNCSY%
ELSE IDENTIFIER;%
CURSY:=IF CURNAME1="9PROCED" AND CURNAME2="URE" THEN PROCSY%
ELSE IDENTIFIER;%
CURSY:=IDENTIFIER; % 10 CHARACTERS,
CURSY:=IDENTIFIER; % 11 CHARACTERS,
CURSY:=IDENTIFIER; % 12 CHARACTERS,
END OF CASE;%

```

```

00000933 T 0208
00000934 T 0210
00000935 T 0213
00000936 T 0215
00000937 T 0217
00000938 T 0219
00000939 T 0221
00000940 T 0223
00000941 T 0227
00000942 T 0229
00000943 T 0231
00000944 T 0233
00000945 T 0236
00000946 T 0238
00000947 T 0240
00000948 T 0243
00000949 T 0244
00000950 T 0247
00000951 T 0249
00000952 T 0250
00000953 T 0251
00000954 T 0252

```

```

START OF SEGMENT ***** 9
9 IS 14 LONG, NEXT SEG 8

```

```

IF RESWORDOPTION AND CURSY#IDENTIFIER THEN%
BEGIN T1:=CARDLENGTH-CHARCNT-CURLNGTH;%
IF CHARCNT=0 THEN CARDPNT:=CARDPNT+1 ELSE T1:=T1-1;%
REPLACE XLINEPNT+T1 BY CARDPNT-(CURLNGTH+1)%
FOR CURLNGTH;%
END;%
END OF LETTER ELSE%
IF C<=9 THEN%
BEGIN%
CURVAL:=C; CURSY:=INTCONST;%
NEXTCHAR;%
WHILE C<=9 DO BEGIN CURVAL:=10×CURVAL+C; NEXTCHAR END;%
IF C="," THEN%
BEGIN%
NEXTCHAR;%
IF C<=9 THEN%
BEGIN CURSY:=REALCONST;%
DO BEGIN CURVAL:=10×CURVAL+C;%
SCALE:=SCALE-1; NEXTCHAR;%
END UNTIL C>9;%
END ELSE IF C="," THEN C:=64 % SPECIAL MARK FOR ","
ELSE ERROR(4);%
END;%
IF C="E" THEN%
BEGIN%
CURSY:=REALCONST; NEXTCHAR;%
IF C="+" OR C="-" THEN BEGIN NEGEXP:=C="-"; NEXTCHAR END;%
IF C<=9 THEN%
BEGIN EXP:=C; NEXTCHAR;%
WHILE C<=9 DO BEGIN EXP:=10×EXP+C; NEXTCHAR END;%
IF NEGEXP THEN EXP:=-EXP;%
END ELSE ERROR(4);%
SCALE:=SCALE+EXP;%

```

```

00000955 T 0253
00000956 T 0254
00000957 T 0300
00000958 T 0306
00000959 T 0311
00000960 T 0313
00000961 T 0313
00000962 T 0313
00000963 T 0315
00000964 T 0315
00000965 T 0317
00000966 T 0324
00000967 T 0336
00000968 T 0336
00000969 T 0337
00000970 T 0344
00000971 T 0345
00000972 T 0346
00000973 T 0348
00000974 T 0357
00000975 T 0358
00000976 T 0360
00000977 T 0362
00000978 T 0362
00000979 T 0363
00000980 T 0363
00000981 T 0372
00000982 T 0383
00000983 T 0383
00000984 T 0392
00000985 T 0404
00000986 T 0405
00000987 T 0407

```



PRT(276) = OVERFLOW  
 PRT(277) = LN  
 PRT(300) = EXP  
 PRT(301) = X TO THE I

```

END;%
IF CURSY=REALCONST THEN%
BEGIN%
  REALOVERFLOW:=OVERFLOW;%
  CURVAL:=CURVAL*10*SCALE;%
  REALOVERFLOW:=0;%
END ELSE%
IF CURVAL>MAXINT THEN%
BEGIN%
OVERFLOW: ERROR(14); CURVAL:=0; REALOVERFLOW:=0;%
END;%
END OF DIGIT ELSE%
IF C=QUOTES THEN%
BEGIN%
CURSY:=ALFACONST; CURLNGTH:=0; NEXTCHAR;%
FINIS:=FALSE;%
DO BEGIN%
  IF C=QUOTES THEN BEGIN NEXTCHAR; FINIS:=C#QUOTES END ELSE%
  IF CHARCNT#0 THEN BEGIN ERROR(6); FINIS:=TRUE END;%
  IF NOT FINIS THEN%
  BEGIN%
    REPLACE STRINGPNT+CURLNGTH BY CHARPNT FOR 1;%
    CURLNGTH:=CURLNGTH+1;%
    NEXTCHAR;%
  END END UNTIL FINIS;%
  IF CURLNGTH=0 THEN ERROR(4) ELSE%
  IF CURLNGTH=1 THEN%
  BEGIN CURSY:=CHARCONST;%
  REPLACE CHARPNT BY STRINGPNT FOR 1; CURVAL:=CH[0];%
  END ELSE%
  IF CURLNGTH#7 THEN%
  BEGIN TEXT[0]:=" " %;%
  REPLACE TEXTPNT BY STRINGPNT FOR CURLNGTH;%
  CURVAL:=TEXT[0];%
  END;%
END OF STRINGS ELSE%
BEGIN%
CURSY:=SYMBOL[C]; 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
  NEXTCHAR;%
  IF C=DOLLAR THEN % DOLLAR INDICATES COMPILER OPTIONS.
  DO BEGIN%

```

```

00000988 T 0408
00000989 T 0408
00000990 T 0409
00000991 T 0409
00000992 T 0411
00000993 T 0414
00000994 T 0415
00000995 T 0415
00000996 T 0416
00000997 T 0417
00000998 T 0419
00000999 T 0419
00001000 T 0419
00001001 T 0421
00001002 T 0422
00001003 T 0431
00001004 T 0432
00001005 T 0432
00001006 T 0442
00001007 T 0445
00001008 T 0445
00001009 T 0446
00001010 T 0451
00001011 T 0452
00001012 T 0459
00001013 T 0460
00001014 T 0462
00001015 T 0463
00001016 T 0465
00001017 T 0468
00001018 T 0468
00001019 T 0470
00001020 T 0471
00001021 T 0474
00001022 T 0475
00001023 T 0475
00001024 T 0475
00001025 T 0477
00001026 T 0485
00001027 T 0487
00001028 T 0496
00001029 T 0498
00001030 T 0507
00001031 T 0509
00001032 T 0518
00001033 T 0520
00001034 T 0529
00001035 T 0531
00001036 T 0540
00001037 T 0542
00001038 T 0542
00001039 T 0550
00001040 T 0551

```

```

NEXTCHAR; CX:=C; NEXTCHAR;%
IF CX="L" THEN IF C=1 THEN HEADING%
                        ELSE LISTOPTION:=C="+ " ELSE%
IF CX="R" THEN RESWORDOPTION:=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 - CARLENGTH;%
  CARLENGTH:=10xC;%
  NEXTCHAR; CARLENGTH:=CARLENGTH+C;%
  IF CARLENGTH<9 OR CARLENGTH>80 THEN%
  BEGIN ERROR(14); CARLENGTH:=72 END;%
  CHARCNT:=MAX(0,LASTCHARPOS+CARLENGTH-1);%
END;%
NEXTCHAR;%
END UNTIL C#" "%;%
FINIS:=FALSE;%
DO BEGIN%
  IF C#" "% 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;%

```

```

00001041 T 0552
00001042 T 0567
00001043 T 0591
00001044 T 0593
00001045 T 0596
00001046 T 0599
00001047 T 0602
00001048 T 0605
00001049 T 0606
00001050 T 0610
00001051 T 0614
00001052 T 0615
00001053 T 0617
00001054 T 0618
00001055 T 0627
00001056 T 0629
00001057 T 0631
00001058 T 0635
00001059 T 0635
00001060 T 0642
00001061 T 0644
00001062 T 0644
00001063 T 0645
00001064 T 0645
00001065 T 0648
00001066 T 0715
00001067 T 0723
00001068 T 0732
00001069 T 0733
00001070 T 0734
00001071 T 0741
00001072 T 0742
00001073 T 0742
00001074 T 0742

```

8 IS 754 LONG, NEXT SEG 2

```

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
%
%
%
%
%
%
%
%
%
%
%
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
%
%
PROCEDURE EXPRESSION; FORWARD;%
PRT(302) = EXPRESSION
PROCEDURE CONCAT; FORWARD;%
PRT(303) = CONCAT
%
ALPHA TEMPSYM;%
PRT(304) = TEMPSYM
REAL SX;%
PRT(305) = SX
INTEGER EXPRLEVEL, TX;%
PRT(306) = EXPRLEVEL
PRT(307) = TX
%
DEFINE PUTTEXT(T)=%
BEGIN%
IF NUMSYMS=MAXSYMS THEN%
BEGIN ERROR(71);%
NUMSYMS:=1;%
END ELSE NUMSYMS:=NUMSYMS+1;%
SYMTAB[NUMSYMS]:=T;%
END OF PUTTEXT #;%
%
DEFINE PUTSYM(S)=%
BEGIN%
TEMPSYM:=(S)&1[41;5;6];%
PUTTEXT(TEMPSYM);%
END OF PUTSYM #;%
%
DEFINE PUTCONST(VAL)=%
BEGIN%
PUTTEXT("2000000");%
PUTTEXT(VAL);%
END OF PUTCONST #;%
%
DEFINE PUTDUMMY=%
BEGIN%
PUTTEXT("3000000");%
END OF PUTDUMMY #;%
%
DEFINE PUTID(L,NUM,NUMDIG)=%
BEGIN%
TEXT[0]:=" " & L [35;5;6];%
REPLACE TEXTPNT+2 BY NUM FOR NUMDIG DIGITS;%
PUTTEXT(TEXT[0]);%
END OF PUTID#;%
00001076 T 0375
00001077 T 0375
00001078 T 0375
00001079 T 0375
00001080 T 0375
00001081 T 0375
00001082 T 0375
00001083 T 0375
00001084 T 0375
00001085 T 0375
00001086 T 0375
00001087 T 0375
00001088 T 0375
00001089 T 0375
00001090 T 0375
00001091 T 0375
00001092 T 0375
00001093 T 0375
00001094 T 0375
00001095 T 0375
00001096 T 0375
00001097 T 0375
00001098 T 0375
00001099 T 0375
00001100 T 0375
00001101 T 0375
00001102 T 0375
00001103 T 0375
00001104 T 0375
00001105 T 0375
00001106 T 0375
00001107 T 0375
00001108 T 0375
00001109 T 0375
00001110 T 0375
00001111 T 0375
00001112 T 0375
00001113 T 0375
00001114 T 0375
00001115 T 0375
00001116 T 0375
00001117 T 0375
00001118 T 0375
00001119 T 0375
00001120 T 0375
00001121 T 0375
00001122 T 0375
00001123 T 0375
00001124 T 0375
00001125 T 0375
00001126 T 0375
PART 4: EXPRESSION PARSER,
-----

```

```

%
DEFINE WRITEEXPR=%
BEGIN%
  FOR T1:=1 STEP 1 UNTIL NUMSYMS DO%
  BEGIN%
    SX:=SYMTAB[T1]; TX:=SX,[41:6];%
    IF TX=0 THEN GEN(SX,7,2) ELSE%
    IF TX=3 THEN ELSE%
    IF TX=1 THEN GEN(SX,1,7) ELSE%
    BEGIN%
      T1:=T1+1; SX:=SYMTAB[T1];%
      IF SX,[44:6]=0 THEN GEN(INT(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;%
PRT(310) = SIMPLEVARIABLE
PRT(311) = INSIDEBRACKETS
  INTEGER NUMPOINTERS;%
PRT(312) = NUMPOINTERS
%
  PROCEDURE VARIABLE;%
PRT(313) = VARIABLE
  BEGIN%
    INTEGER STARTSYM,LLIM,ULIM;%
STACK(F+2) = STARTSYM
STACK(F+3) = LLIM
STACK(F+4) = ULIM
  REAL T;%
STACK(F+5) = T
  BOOLEAN INBRACKET,INRECORD;%
STACK(F+6) = INBRACKET
STACK(F+7) = INRECORD
  LABEL ADDADDR;%
%
  STARTSYM:=NUMSYMS+1;%
  IF THISLEVEL>CURLEVEL THEN % VARIABLE IN FIELD LIST OF
  BEGIN % RECORD USED IN WITH-STATEMENT,
    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);%

```

```

00001127 T 0375
00001128 T 0375
00001129 T 0375
00001130 T 0375
00001131 T 0375
00001132 T 0375
00001133 T 0375
00001134 T 0375
00001135 T 0375
00001136 T 0375
00001137 T 0375
00001138 T 0375
00001139 T 0375
00001140 T 0375
00001141 T 0375
00001142 T 0375
00001143 T 0375
00001144 T 0375
00001145 T 0375
00001146 T 0375
00001147 T 0375
00001148 T 0375
00001149 T 0375
00001150 T 0375
00001151 T 0375
00001152 T 0375
00001153 T 0375
00001154 T 0375
00001155 T 0375
00001156 T 0375
00001157 T 0375
00001158 T 0375
00001159 T 0375
00001160 T 0375
START OF SEGMENT ***** 10
00001161 T 0000
00001162 T 0000
00001163 T 0000
00001164 T 0000
00001165 T 0000
00001166 T 0001
00001167 T 0002
00001168 T 0002
00001169 T 0003
00001170 T 0006
00001171 T 0015
00001172 T 0016

```

```

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];%
LLIM:=TYPETAB2[CURTYPE]; ULIM:=TYPETAB3[CURTYPE];%
IF T,FORM#ARRAYS THEN ERROR(12);%
IF INRECORD THEN PUTTEXT("  +(");%
INSYMBOL;%
IF CHECKOPTION THEN CHECKEXPR(LLIM,ULIM) ELSE EXPRESSION;%
CHECKTYPES(T,INXTYPE,CURTYPE);%
CURTYPE:=T,ARRTYPE;%
IF INRECORD THEN%
BEGIN%
IF LLIM<0 THEN BEGIN PUTSYM("+"); PUTCONST(-LLIM) END ELSE%
IF LLIM>0 THEN BEGIN PUTSYM("-"); PUTCONST(LLIM) END;%
PUTSYM(")");%
IF TYPETAB1[CURTYPE],SIZE>1 THEN%
BEGIN PUTSYM("x"); PUTCONST(TYPETAB1[CURTYPE],SIZE) END;%
END ELSE IF TYPETAB1[CURTYPE],STRUCT>0 THEN PUTSYM(",");%
END UNTIL CURSY#COMMA;%
IF CURSY#RBRACKET THEN%
BEGIN ERROR(59); SKIP(RBRACKET);%
IF CURSY=RBRACKET THEN INSYMBOL;%
END ELSE INSYMBOL;%
END OF BRACKETS ELSE%
IF CURSY=DOT THEN%
BEGIN%
IF NOT(INBRACKET OR INRECORD) THEN%
BEGIN PUTSYM("."); INBRACKET:=TRUE END;%
T:=TYPETAB1[CURTYPE];%
IF T,FORM#RECORD THEN ERROR(12);%
INSYMBOL;%
IF CURSY=IDENTIFIER THEN%
BEGIN%
SEARCHTAB(T,RECTAB);%
IF FOUND THEN%
BEGIN%
THISID:=NAMETAB3[T,RECTAB,THISINDEX];%
PUTSYM("+");%
PUTCONST(THISID,INFO); CURTYPE:=THISID,TYPE;%
END ELSE BEGIN ERROR(1); CURTYPE:=0 END;%
END ELSE BEGIN ERROR(9); CURTYPE:=0 END;%

```

ADDADDR;

```

00001173 T 0017
00001174 T 0019
00001175 T 0019
00001176 T 0022
00001177 T 0022
00001178 T 0022
00001179 T 0025
00001180 T 0027
00001181 T 0043
00001182 T 0044
00001183 T 0047
00001184 T 0047
00001185 T 0048
00001186 T 0049
00001187 T 0049
00001188 T 0050
00001189 T 0051
00001190 T 0060
00001191 T 0060
00001192 T 0061
00001193 T 0063
00001194 T 0065
00001195 T 0072
00001196 T 0072
00001197 T 0151
00001198 T 0212
00001199 T 0213
00001200 T 0213
00001201 T 0214
00001202 T 0235
00001203 T 0257
00001204 T 0265
00001205 T 0266
00001206 T 0288
00001207 T 0298
00001208 T 0299
00001209 T 0300
00001210 T 0302
00001211 T 0304
00001212 T 0305
00001213 T 0305
00001214 T 0306
00001215 T 0306
00001216 T 0307
00001217 T 0316
00001218 T 0317
00001219 T 0320
00001220 T 0320
00001221 T 0321
00001222 T 0321
00001223 T 0342
00001224 T 0342
00001225 T 0342
00001226 T 0345
00001227 T 0352
00001228 T 0367
00001229 T 0369

```

```

INRECORD:=TRUE;%
INSYMBOL;%
END OF DOT ELSE%
BEGIN % CURSY=ARROW
T:=TYPETAB1[CURTYPE];%
IF T,FORM=FILES THEN%
BEGIN%
CURTYPE:=T,FILETYPE;%
IF TYPETAB1[CURTYPE],STRUCT=0 THEN PUTTEXT(" [0]");%
END ELSE%
IF T,FORM=TEXTFILE THEN%
BEGIN%
SYMTAB[NUMSYMS]:=SYMTAB[NUMSYMS] & "]" [351516];%
PUTSYM(","); PUTTEXT("LASTCH");%
CURTYPE:=CHARTYPE;%
END ELSE%
IF T,FORM=POINTERS THEN%
BEGIN%
IF INBRACKET THEN PUTSYM(")");%
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]:= " (T1=";%
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 TYPETAB1[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;%
END;%
INSIDEBRACKETS:=INBRACKET;%
CURMODE:=NUMBER;%
END OF VARIABLE;%

```

```

00001230 T 0371
00001231 T 0372
00001232 T 0372
00001233 T 0372
00001234 T 0373
00001235 T 0374
00001236 T 0375
00001237 T 0375
00001238 T 0377
00001239 T 0384
00001240 T 0384
00001241 T 0388
00001242 T 0388
00001243 T 0391
00001244 T 0404
00001245 T 0405
00001246 T 0405
00001247 T 0408
00001248 T 0408
00001249 T 0417
00001250 T 0417
00001251 T 0419
00001252 T 0419
00001253 T 0421
00001254 T 0425
00001255 T 0426
00001256 T 0428
00001257 T 0430
00001258 T 0431
00001259 T 0434
00001260 T 0436
00001261 T 0438
00001262 T 0438
00001263 T 0438
00001264 T 0441
00001265 T 0443
00001266 T 0443
00001267 T 0452
00001268 T 0453
00001269 T 0466
00001270 T 0480
00001271 T 0481
00001272 T 0484
00001273 T 0484
00001274 T 0484
00001275 T 0484
00001276 T 0485
10 IS 490 LONG, NEXT SEG 2

```

```

%
%
PROCEDURE PASSPARAMS;%
PRT(314) = PASSPARAMS
BEGIN%

```

```

00001277 T 0375
00001278 T 0375
00001279 T 0375
00001280 T 0375

```

INTEGER NPARS,PARAM,PTYPE,P,FIRSTSYM;%

00001281 T 0375  
START OF SEGMENT \*\*\*\*\* 11

STACK(F+2) = NPARS  
STACK(F+3) = PARAM  
STACK(F+4) = PTYPE  
STACK(F+5) = P  
STACK(F+6) = FIRSTSYM

BOOLEAN FORMALPROC,CHECK;%  
STACK(F+7) = FORMALPROC  
STACK(F+10) = CHECK

LABEL EXIT;%

%

PUTID("V",1000\*THISLEVEL+THISINDEX,5);%  
PI=THISID,INFO;%  
FORMALPROC=BOOLEAN(THISID,FORMAL);%  
NPARS=PARAMTAB[P]; PI=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]; PI=P+1;%  
PTYPE=PARAM,PARAMTYPE;%  
IF PARAM,PARAMKIND=CONST THEN%  
BEGIN%  
CHECK=CHECKOPTION AND TYPETAB1[PTYPE],FORM LEQ CHAR;%  
IF CHECK THEN PUTTEXT("CHECK(");%  
PUTDUMMY; FIRSTSYM=NUMSYMS;%  
EXPRLEVEL=EXPRLEVEL+1;%  
EXPRESSION; EXPRLEVEL=EXPRLEVEL-1;%  
IF CURMODE=BITPATTERN THEN%  
BEGIN SYMTAB[FIRSTSYM]= " REAL(" PUTSYM(")"); END;%  
IF CHECK THEN%  
BEGIN%  
PUTSYM(","); PUTCONST(TYPETAB2[PTYPE]);%  
PUTSYM(","); PUTCONST(TYPETAB3[PTYPE]);%  
PUTSYM(","); PUTCONST(CARDCNT); PUTSYM(")");%  
END;%  
END ELSE%  
IF PARAM,PARAMKIND=VAR THEN%  
BEGIN%  
IF CURSY=IDENTIFIER THEN%  
BEGIN%  
SEARCH;%  
IF FOUND THEN%  
BEGIN%  
IF THISID,IDCLASS=VAR OR%  
THISID,IDCLASS=CONST AND BOOLEAN(THISID,FORMAL) THEN%  
BEGIN%  
IF PARAM,PARAMFILE=1 THEN%  
BEGIN%  
CURTYPE=THISID,TYPE;%  
PUTID("V",1000\*THISLEVEL+THISINDEX,5); PUTSYM(",");%  
PUTID("F",1000\*THISLEVEL+THISINDEX,5); PUTSYM(",");%  
END;%  
END;%  
END;%  
END;%  
END;%

00001282 T 0000

00001283 T 0000

00001284 T 0000

00001285 T 0000

00001286 T 0016

00001287 T 0018

00001288 T 0019

00001289 T 0021

00001290 T 0023

00001291 T 0023

00001292 T 0024

00001293 T 0024

00001294 T 0033

00001295 T 0034

00001296 T 0034

00001297 T 0037

00001298 T 0040

00001299 T 0041

00001300 T 0042

00001301 T 0043

00001302 T 0045

00001303 T 0052

00001304 T 0060

00001305 T 0061

00001306 T 0063

00001307 T 0064

00001308 T 0075

00001309 T 0075

00001310 T 0076

00001311 T 0096

00001312 T 0117

00001313 T 0146

00001314 T 0146

00001315 T 0146

00001316 T 0147

00001317 T 0148

00001318 T 0149

00001319 T 0149

00001320 T 0178

00001321 T 0178

00001322 T 0178

00001323 T 0180

00001324 T 0182

00001325 T 0183

00001326 T 0184

00001327 T 0184

00001328 T 0186

00001329 T 0210

```

        PUTID("I",1000*THISLEVEL+THISINDEX,5);%
        INSYMBOL;%
    END ELSE%
    BEGIN%
        VARIABLE;%
        IF TYPETAB1[CURTYPE],STRUCT>0 THEN%
        IF NOT SIMPLEVARIABLE THEN ERROR(92);%
        END;%
    END ELSE BEGIN ERROR(8); CURTYPE!=0 END;%
    END ELSE BEGIN ERROR(1); CURTYPE!=0 END;%
    END ELSE BEGIN ERROR(9); CURTYPE!=0 END;%
END ELSE%
BEGIN%
    IF CURSY=IDENTIFIER THEN%
    BEGIN%
        SEARCH;%
        IF FOUND THEN%
        BEGIN%
            IF THISID.IDCLASS#PARAM,PARAMKIND THEN ERROR(91);%
            PUTID("V",1000*THISLEVEL+THISINDEX,5);%
            CURTYPE!=IF THISID.IDCLASS=FUNC THEN THISID,TYPE ELSE 0;%
            INSYMBOL;%
        END ELSE BEGIN ERROR(1); CURTYPE!=0 END;%
        END ELSE BEGIN ERROR(9); CURTYPE!=0 END;%
    END;%
    CHECKTYPES(PTYPE,CURTYPE);%
    NPARS:=NPARS+1;%
    IF CURSY=COMMA THEN PUTSYM(",");%
    END UNTIL CURSY#COMMA;%
    IF CURSY#RPAR THEN BEGIN ERROR(89); SKIP(RPAR) END;%
EXIT: PUTSYM(")");%
    IF CURSY=RPAR THEN INSYMBOL;%
    END;%
    IF NPARS>0 AND NOT FORMALPROC THEN ERROR(3);%
    CURMODE:=NUMBER;%
END OF PASSPARAMS;%

```

```

00001330 T 0234
00001331 T 0250
00001332 T 0251
00001333 T 0251
00001334 T 0251
00001335 T 0252
00001336 T 0253
00001337 T 0256
00001338 T 0256
00001339 T 0258
00001340 T 0260
00001341 T 0262
00001342 T 0262
00001343 T 0262
00001344 T 0263
00001345 T 0263
00001346 T 0292
00001347 T 0292
00001348 T 0292
00001349 T 0295
00001350 T 0312
00001351 T 0316
00001352 T 0317
00001353 T 0319
00001354 T 0321
00001355 T 0321
00001356 T 0380
00001357 T 0381
00001358 T 0390
00001359 T 0391
00001360 T 0394
00001361 T 0401
00001362 T 0403
00001363 T 0403
00001364 T 0406
00001365 T 0407

```

11 IS 414 LONG, NEXT SEG 2

```

%
%
PROCEDURE FACTOR;
PRT(315) = FACTOR
BEGIN
    INTEGER STARTSYM,STYPE,T;%

STACK(F+2) = STARTSYM
STACK(F+3) = STYPE
STACK(F+4) = T

STACK(F+5) = FIRST    BOOLEAN FIRST;%
STACK(F+6) = VAL      REAL VAL;%

%
DEFINE PARAMETER=
BEGIN%

```

```

%*** FACTOR ***
%*****

```

```

00001366 T 0375
00001367 T 0375
00001368 T 0375

00001369 T 0375
00001370 T 0375
START OF SEGMENT ***** 12

00001371 T 0000
00001372 T 0000

00001373 T 0000
00001374 T 0000
00001375 T 0000

```

%\*\*\* CHECK THAT THE FUNCTION HAS 1 PARAM.



```

INSYMBOL;%
IF CURSY=LPAR THEN%
BEGIN%
  PUTSYM("("); INSYMBOL; EXPRESSION;%
  IF TYPETAB1[CURTYPE],FORM=NUMERIC THEN CURTYPE:=INTTYPE;%
  IF CURSY#RPAR THEN BEGIN ERROR(3); SKIP(RPAR) END;%
  PUTSYM(")"); IF CURSY=RPAR THEN INSYMBOL;%
END ELSE ERROR(3);%
END OF PARAMETER#;%
%
CURMODE:=NUMBER;%
IF CURSY=IDENTIFIER THEN%
BEGIN%
  SEARCH;%
  IF FOUND THEN%
  BEGIN%
    IF THISID.IDCLASS=VAR OR%
      THISID.IDCLASS=CONST AND BOOLEAN(THISID.FORMAL)%
    THEN VARIABLE ELSE%
    IF THISID.IDCLASS=CONST THEN%
    BEGIN%
      IF THISID.INFO<=1023 THEN PUTCONST(THISID.INFO)%
        ELSE PUTCONST(CONSTTAB[THISID.INFO-1023]);
      CURTYPE:=THISID.TYPE; CURMODE:=NUMBER;%
    INSYMBOL;%
    END ELSE%
    IF THISID.IDCLASS=FUNC THEN%
    BEGIN%
      IF THISTAB=0 THEN          %*** INTRINSIC FUNCTION ***
      BEGIN%
        INTEGER DUMMY;%

```

```

00001376 T 0000
00001377 T 0000
00001378 T 0000
00001379 T 0000
00001380 T 0000
00001381 T 0000
00001382 T 0000
00001383 T 0000
00001384 T 0000
00001385 T 0000
00001386 T 0000
00001387 T 0000
00001388 T 0001
00001389 T 0002
00001390 T 0031
00001391 T 0031
00001392 T 0031
00001393 T 0033
00001394 T 0035
00001395 T 0036
00001396 T 0038
00001397 T 0038
00001398 T 0054
00001399 T 0068
00001400 T 0070
00001401 T 0071
00001402 T 0071
00001403 T 0073
00001404 T 0073
00001405 T 0074
00001406 T 0074

```

PRT(316) = \*SEGMENT DESCRIPTOR\*

STACK(F+7) = DUMMY

START OF SEGMENT \*\*\*\*\* 13

```

IF CURNAME1="3000ABS" THEN          % "ABS"
BEGIN%
  PUTTEXT(" ABS"); PARAMETER;%
  IF CURTYPE#REALTYPE AND CURTYPE#INTTYPE THEN ERROR(67);%
END ELSE%
IF CURNAME1="3000CHR" THEN          % "CHR"
BEGIN%
  INSYMBOL;%
  IF CURSY=LPAR THEN%
  BEGIN INSYMBOL; CHECKEXPR(0,63);%
    IF TYPETAB1[CURTYPE],FORM=NUMERIC THEN ERROR(67);%
    IF CURSY#RPAR THEN BEGIN ERROR(46); SKIP(RPAR) END;%
    IF CURSY=RPAR THEN INSYMBOL;%
  END ELSE ERROR(58);%
  CURTYPE:=CHARTYPE;%
END ELSE%
IF CURNAME1="3000EOF" OR          % "EOF"/"EOLN"
  CURNAME1="400EOLN" THEN%
BEGIN%
  FIRST:=CURNAME1="3000EOF";%
  FILEPARAM(INPUTFILE);%
  PUTID("I",FILENAME,5);%
  PUTTEXT( (IF FIRST THEN " ,EOF" ELSE " ,EOLN") );%

```

```

00001407 T 0000
00001408 T 0000
00001409 T 0001
00001410 T 0036
00001411 T 0039
00001412 T 0039
00001413 T 0040
00001414 T 0041
00001415 T 0041
00001416 T 0042
00001417 T 0120
00001418 T 0122
00001419 T 0125
00001420 T 0127
00001421 T 0128
00001422 T 0129
00001423 T 0129
00001424 T 0130
00001425 T 0131
00001426 T 0132
00001427 T 0133
00001428 T 0186
00001429 T 0201

```

```

IF LPARFOUND THEN%
BEGIN%
  IF CURSY#RPAR THEN BEGIN ERROR(46); SKIP(RPAR) END;%
  IF CURSY=RPAR THEN INSYMBOL;%
END;%
CURTYPE:=BOOLTYPE;%
END ELSE%
IF CURNAME1="3000ODD" THEN % "ODD"
BEGIN%
  PUTTEXT(" ODD"); PARAMETER;%
  IF CURTYPE#INTTYPE THEN ERROR(67);%
  CURTYPE:=BOOLTYPE; CURMODE:=BITPATTERN;%
END ELSE%
IF CURNAME1="3000ORD" THEN % "ORD"
BEGIN%
  PUTSYM("("); INSYMBOL;%
  IF CURSY#LPAR THEN%
  BEGIN%
    INSYMBOL; EXPRESSION;%
    IF TYPETAB1[CURTYPE],FORM>CHAR THEN ERROR(67);%
    IF CURSY#RPAR THEN BEGIN ERROR(46); SKIP(RPAR) END;%
    INSYMBOL;%
  END ELSE ERROR(58);%
  CURTYPE:=INTTYPE; PUTSYM(")");%
END ELSE%
IF CURNAME1="400PRED" OR % "PRED"/"SUCC"
CURNAME1="400SUCC" THEN%
BEGIN%
  FIRST:=CURNAME1="400PRED";%
  PUTTEXT("CHECK("); INSYMBOL;%
  IF CURSY#LPAR THEN%
  BEGIN%
    INSYMBOL; EXPRESSION;%
    PUTSYM(IF FIRST THEN "-" ELSE "+"); PUTSYM("1");%
    IF TYPETAB1[CURTYPE],FORM>CHAR THEN ERROR(67);%
    PUTSYM(","); PUTCONST(TYPETAB2[CURTYPE]);%
    PUTSYM(","); PUTCONST(TYPETAB3[CURTYPE]);%
    PUTSYM(","); PUTCONST(CARDCNT);%
    PUTSYM(")");%
    IF CURSY#RPAR THEN BEGIN ERROR(46); SKIP(RPAR) END;%
    IF CURSY=RPAR THEN INSYMBOL;%
  END ELSE BEGIN ERROR(58); CURTYPE:=0 END;%
END ELSE%
IF CURNAME1="50ROUND" THEN % "ROUND"
BEGIN%
  PUTTEXT(" ROUND"); PARAMETER;%
  IF CURTYPE#REALTYPE THEN ERROR(67);%
  NUMSYMS:=NUMSYMS-1; PUTSYM(",");%
  PUTCONST(CARDCNT); PUTSYM(")");%
  CURTYPE:=INTTYPE;%
END ELSE%
IF CURNAME1="3000SQR" THEN % "SQR"
BEGIN%
  PUTTEXT(" SQR"); PARAMETER;%
  NUMSYMS:=NUMSYMS-1; PUTSYM(",");%
  PUTCONST(CARDCNT); PUTSYM(")");%
  IF CURTYPE#REALTYPE AND CURTYPE#INTTYPE THEN ERROR(67);%

```

```

00001430 T 0209
00001431 T 0209
00001432 T 0209
00001433 T 0212
00001434 T 0214
00001435 T 0214
00001436 T 0215
00001437 T 0215
00001438 T 0218
00001439 T 0219
00001440 T 0254
00001441 T 0256
00001442 T 0258
00001443 T 0258
00001444 T 0259
00001445 T 0259
00001446 T 0269
00001447 T 0269
00001448 T 0270
00001449 T 0271
00001450 T 0274
00001451 T 0276
00001452 T 0277
00001453 T 0278
00001454 T 0286
00001455 T 0286
00001456 T 0288
00001457 T 0289
00001458 T 0289
00001459 T 0290
00001460 T 0299
00001461 T 0299
00001462 T 0300
00001463 T 0301
00001464 T 0319
00001465 T 0321
00001466 T 0342
00001467 T 0363
00001468 T 0384
00001469 T 0392
00001470 T 0394
00001471 T 0396
00001472 T 0398
00001473 T 0398
00001474 T 0399
00001475 T 0400
00001476 T 0435
00001477 T 0437
00001478 T 0446
00001479 T 0467
00001480 T 0467
00001481 T 0467
00001482 T 0469
00001483 T 0469
00001484 T 0504
00001485 T 0513
00001486 T 0534

```

```

END ELSE%
IF CURNAME1="50TRUNC" THEN % "TRUNC"
BEGIN%
  PUTTEXT(" TRUNC"); PARAMETER;%
  NUMSYMS1=NUMSYMS=1; PUTSYM(",");%
  PUTCONST(CARDCNT); PUTSYM(" ");%
  IF CURTYPE#REALTYPE THEN ERROR(67);%
  CURTYPE1=INTTYPE;%
END ELSE%
IF CURNAME1="6CONCAT" THEN % "CONCAT"
CONCAT ELSE%
IF CURNAME1="400TIME" THEN % "TIME"
BEGIN%
  PUTTEXT("(TIME("); PUTTEXT("1)/60)");%
  CURTYPE1=REALTYPE; INSYMBOL;%
END ELSE%
IF CURNAME1="400DATE" THEN % "DATE"
BEGIN%
  PUTTEXT("CURDAT");%
  CURTYPE1=ALFATYPE; INSYMBOL;%
END ELSE%
IF CURNAME1="7ELAPSE" AND CURNAME2="D" THEN % "ELAPSED"
BEGIN%
  PUTTEXT("(TIME("); PUTTEXT("2)/60)");%
  CURTYPE1=REALTYPE; INSYMBOL;%
END ELSE%
IF CURNAME1="6IOTIME" THEN % "IOTIME"
BEGIN%
  PUTTEXT("(TIME("); PUTTEXT("3)/60)");%
  CURTYPE1=REALTYPE; INSYMBOL;%
END ELSE%
IF CURNAME1="7WEEKDA" AND CURNAME2="Y" THEN % "WEEKDAY"
BEGIN%
  PUTTEXT("WEEKDA");%
  CURTYPE1=ALFATYPE; INSYMBOL;%
END ELSE IF CURNAME1="400USER" THEN % "USER"
BEGIN%
  PUTTEXT(" TIME"); PUTTEXT(" (-1)");%
  CURTYPE1=ALFATYPE; INSYMBOL;%
END ELSE % "SIN","COS" ETC,
BEGIN%
  PUTTEXT(IF CURNAME1="3000SIN" THEN " SIN" ELSE%
    IF CURNAME1="3000COS" THEN " COS" ELSE%
    IF CURNAME1="6ARCTAN" THEN "ARCTAN" ELSE%
    IF CURNAME1="400SQRT" THEN " SQRT" ELSE%
    IF CURNAME1="3000EXP" THEN " EXP" ELSE%
    " LN");%
  PARAMETER;%
  IF CURTYPE#REALTYPE AND CURTYPE#INTTYPE THEN ERROR(67);%
  CURTYPE1=REALTYPE;%
END;%
END OF INTRINSIC FUNCTIONS ELSE%

```

```

00001487 T 0537
00001488 T 0537
00001489 T 0538
00001490 T 0538
00001491 T 0573
00001492 T 0582
00001493 T 0603
00001494 T 0605
00001495 T 0605
00001496 T 0605
00001497 T 0607
00001498 T 0608
00001499 T 0610
00001500 T 0611
00001501 T 0625
00001502 T 0626
00001503 T 0626
00001504 T 0629
00001505 T 0630
00001506 T 0637
00001507 T 0638
00001508 T 0638
00001509 T 0642
00001510 T 0643
00001511 T 0657
00001512 T 0658
00001513 T 0658
00001514 T 0661
00001515 T 0662
00001516 T 0676
00001517 T 0677
00001518 T 0677
00001519 T 0681
00001520 T 0682
00001521 T 0689
00001522 T 0690
00001523 T 0693
00001524 T 0694
00001525 T 0708
00001526 T 0709
00001527 T 0709
00001528 T 0712
00001529 T 0712
00001530 T 0712
00001531 T 0712
00001532 T 0712
00001533 T 0712
00001534 T 0727
00001535 T 0765
00001536 T 0768
00001537 T 0769
00001538 T 0769

```

PRT(317) = \*SEGMENT DESCRIPTOR\*

```

BEGIN%
T1=THISID,TYPE;%
PASSPARAMS;%

```

```

13 IS 784 LONG, NEXT SEG 12
00001539 T 0076
00001540 T 0076
00001541 T 0077

```

```

        CURTYPE:=T;%
    END;%
END OF FUNCTIONS ELSE%
IF THISID, IDCLASS=PROC THEN%
BEGIN%
    ERROR(68); PASSPARAMS;%
    CURTYPE:=0;%
    END ELSE BEGIN ERROR(69); CURTYPE:=0; INSYMBOL END;%
END ELSE BEGIN ERROR(1); CURTYPE:=0; INSYMBOL END;%
END OF IDENTIFIER ELSE%
IF CURSYSCHARCONST THEN%
BEGIN%
    CONSTANT(VAL, CURTYPE); PUTCONST(VAL);%
END ELSE%
IF CURSY=NOTSY THEN%
BEGIN%
    PUTTEXT(" NOT "); PUTDUMMY; STARTSYM:=NUMSYMS;%
    INSYMBOL; FACTOR;%
    IF CURTYPE>0 THEN%
    IF CURTYPE#BOOLTYPE THEN BEGIN ERROR(17); CURTYPE:=0 END;%
    IF CURMODE=NUMBER THEN%
    BEGIN SYMTAB[STARTSYM]:=" B("; PUTSYM(")");%
        CURMODE:=BITPATTERN;%
    END;%
END ELSE%
IF CURSY=NILSY THEN%
BEGIN%
    PUTCONST(0); CURTYPE:=NILTYPE;%
    INSYMBOL;%
END ELSE%
IF CURSY=LPAR THEN%
BEGIN%
    PUTSYM("(");%
    INSYMBOL; EXPRESSION;%
    IF CURSY#RPAR THEN BEGIN ERROR(46); SKIP(RPAR) END;%
    PUTSYM(")");%
    INSYMBOL;%
END ELSE%
IF CURSY=LBRACKET THEN          %*** SET CONSTANT ***
BEGIN%
    INSYMBOL;%
    IF CURSY=RBRACKET THEN%
    BEGIN%
        PUTCONST(0); CURTYPE:=EMPTYSET; CURMODE:=NUMBER;%
        INSYMBOL;%
    END ELSE%
    BEGIN%
        FIRST:=TRUE;%
        DO BEGIN%
            IF FIRST THEN FIRST:=FALSE ELSE INSYMBOL;%
            PUTTEXT(" BIT("); STARTSYM:=NUMSYMS;%
            EXPRESSION;%
            IF STYPE=0 THEN%
            BEGIN STYPE:=CURTYPE;%
                IF TYPETAB[CURTYPE], FORM>CHAR THEN ERROR(72);%
            END ELSE CHECKTYPES(STYPE, CURTYPE);%
            IF CURSY=DOUBLEDOT THEN%

```

```

00001542 T 0078
00001543 T 0079
00001544 T 0079
00001545 T 0079
00001546 T 0080
00001547 T 0081
00001548 T 0082
00001549 T 0083
00001550 T 0085
00001551 T 0088
00001552 T 0088
00001553 T 0089
00001554 T 0090
00001555 T 0190
00001556 T 0190
00001557 T 0191
00001558 T 0192
00001559 T 0206
00001560 T 0207
00001561 T 0208
00001562 T 0211
00001563 T 0212
00001564 T 0223
00001565 T 0224
00001566 T 0224
00001567 T 0224
00001568 T 0225
00001569 T 0226
00001570 T 0239
00001571 T 0239
00001572 T 0239
00001573 T 0241
00001574 T 0241
00001575 T 0249
00001576 T 0250
00001577 T 0252
00001578 T 0260
00001579 T 0260
00001580 T 0260
00001581 T 0262
00001582 T 0262
00001583 T 0263
00001584 T 0263
00001585 T 0264
00001586 T 0279
00001587 T 0279
00001588 T 0279
00001589 T 0280
00001590 T 0280
00001591 T 0281
00001592 T 0283
00001593 T 0290
00001594 T 0290
00001595 T 0291
00001596 T 0292
00001597 T 0295
00001598 T 0356

```

```

BEGIN%
  PUTSYM(","); SYMTAB[STARTSYM]:= " BITS(";%
  INSYMBOL; EXPRESSION;%
  IF STYPE=0 THEN%
    BEGIN STYPE:=CURTYPE;%
      IF TYPETAB1[CURTYPE],FORM>CHAR THEN ERROR(72);%
    END ELSE CHECKTYPES(STYPE,CURTYPE);%
  END;%
  PUTSYM(","); PUTCONST(CARDCNT); PUTSYM(")");%
  IF CURSY=COMMA THEN PUTTEXT(" OR");%
  END UNTIL CURSY#COMMA;%
  IF CURSY#RBRACKET THEN%
    BEGIN ERROR(59); SKIP(RBRACKET);%
      IF CURSY=RBRACKET THEN INSYMBOL;%
    END ELSE INSYMBOL;%
  NEWTYPE; T1:=SET; T1,SIZE:=1; T1,STRUCT:=0;%
  T1,SETTYPE:=STYPE; TYPETAB1[TYPEINDEX]:=T1;%
  CURTYPE:=TYPEINDEX;%
  CURMODE:=BITPATTERN;%
  END;%
  END OF SET CONSTANT ELSE BEGIN ERROR(99); INSYMBOL END;%
  END OF FACTOR;%

```

```

00001599 T 0356
00001600 T 0357
00001601 T 0366
00001602 T 0367
00001603 T 0367
00001604 T 0369
00001605 T 0371
00001606 T 0433
00001607 T 0433
00001608 T 0461
00001609 T 0468
00001610 T 0469
00001611 T 0470
00001612 T 0472
00001613 T 0473
00001614 T 0476
00001615 T 0485
00001616 T 0488
00001617 T 0489
00001618 T 0490
00001619 T 0490
00001620 T 0492

```

12 IS 498 LONG, NEXT SEG 2

```

%
%
PROCEDURE TERM; %*** TERM ***
PRT(320) = TERM
BEGIN %*****
  INTEGER STARTSYM,MODE,TYPE1,MULOPTR,F;%

```

```

00001621 T 0375
00001622 T 0375
00001623 T 0375

```

```

STACK(F+2) = STARTSYM
STACK(F+3) = MODE
STACK(F+4) = TYPE1
STACK(F+5) = MULOPTR
STACK(F+6) = F

```

START OF SEGMENT \*\*\*\*\* 14

```

PUTDUMMY; STARTSYM:=NUMSYMS;%
FACTOR;%
MODE:=CURMODE;%
WHILE CURSY>ASTERISK AND CURSY<MODSY DO % "*" , "/" , "DIV" , "MOD" , "AND"
BEGIN%
  TYPE1:=CURTYPE; MULOPTR:=CURSY;%
  F:=TYPETAB1[TYPE1],FORM;%
  IF F=NUMERIC OR F=FLOATING THEN%
    BEGIN%
      MODE:=NUMBER;%
      IF CURSY=ASTERISK THEN PUTSYM("x") ELSE%
      IF CURSY=SLASH THEN PUTSYM("/") ELSE%
      IF CURSY=ANDSY THEN ERROR(64) ELSE%
      BEGIN%
        IF F=FLOATING THEN ERROR(64);%
        IF CURSY=DIVSY THEN PUTTEXT(" DIV") ELSE PUTTEXT(" MOD");%
      END END ELSE%
    IF CURTYPE=BOOLTYPE OR F=SET THEN%

```

```

00001626 T 0000
00001627 T 0006
00001628 T 0007
00001629 T 0007
00001630 T 0010
00001631 T 0010
00001632 T 0011
00001633 T 0013
00001634 T 0015
00001635 T 0015
00001636 T 0016
00001637 T 0026
00001638 T 0035
00001639 T 0038
00001640 T 0038
00001641 T 0040
00001642 T 0055
00001643 T 0055

```

```

BEGIN%
MODE:=BITPATTERN;%
IF CURMODE#MODE THEN%
BEGIN SYMTAB[STARTSYM]:= " B(") PUTSYM(")") END;%
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 CURTYPE1#REALTYPE;%
END;%
IF CURTYPE#REALTYPE AND MULOPTR#DIVSY THEN ERROR(65);%
END;%
IF MULOPTR#SLASH THEN CURTYPE:=REALTYPE;%
IF CURTYPE#0 THEN CURTYPE1:=TYPE1;%
END OF WHILE LOOP;%
IF MODE#BITPATTERN AND CURMODE#MODE THEN%
BEGIN SYMTAB[STARTSYM]:= " B(") PUTSYM(")") END;%
CURMODE:=MODE;%
END OF TERM;%

```

```

00001644 T 0059
00001645 T 0060
00001646 T 0061
00001647 T 0061
00001648 T 0072
00001649 T 0078
00001650 T 0082
00001651 T 0084
00001652 T 0091
00001653 T 0092
00001654 T 0094
00001655 T 0094
00001656 T 0095
00001657 T 0095
00001658 T 0098
00001659 T 0159
00001660 T 0161
00001661 T 0161
00001662 T 0164
00001663 T 0164
00001664 T 0166
00001665 T 0168
00001666 T 0168
00001667 T 0170
00001668 T 0180
00001669 T 0181
14 IS 185 LONG, NEXT SEG 2

```

```

%
%
PROCEDURE SIMPLEEXPRESSION; %*** SIMPLE EXPRESSION ***
PRT(321) = SIMPLEEXPRESSION %*****
BEGIN %*****
INTEGER STARTSYM,MODE,TYPE1,F;%
STACK(F+2) = STARTSYM
STACK(F+3) = MODE
STACK(F+4) = TYPE1
STACK(F+5) = F
STACK(F+6) = SIGNED %
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"

```

```

00001670 T 0375
00001671 T 0375
00001672 T 0375
00001673 T 0375
00001674 T 0375
START OF SEGMENT ***** 15
00001675 T 0000
00001676 T 0000
00001677 T 0000
00001678 T 0006
00001679 T 0008
00001680 T 0009
00001681 T 0020
00001682 T 0021
00001683 T 0021
00001684 T 0021
00001685 T 0022
00001686 T 0022
00001687 T 0024
00001688 T 0027
00001689 T 0027

```

```

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%
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;%

```

```

00001690 T 0030
00001691 T 0030
00001692 T 0032
00001693 T 0034
00001694 T 0035
00001695 T 0044
00001696 T 0054
00001697 T 0054
00001698 T 0056
00001699 T 0056
00001700 T 0057
00001701 T 0058
00001702 T 0068
00001703 T 0077
00001704 T 0077
00001705 T 0079
00001706 T 0079
00001707 T 0080
00001708 T 0081
00001709 T 0091
00001710 T 0098
00001711 T 0114
00001712 T 0116
00001713 T 0118
00001714 T 0118
00001715 T 0125
00001716 T 0125
00001717 T 0127
00001718 T 0127
00001719 T 0128
00001720 T 0129
00001721 T 0131
00001722 T 0192
00001723 T 0194
00001724 T 0194
00001725 T 0196
00001726 T 0196
00001727 T 0198
00001728 T 0208
00001729 T 0209

```

15 IS 213 LONG, NEXT SEG 2

```

%
%
PROCEDURE EXPRESSION; %*** EXPRESSION ***
BEGIN %*****
INTEGER STARTSYM,FIRSTSYM,TYPE1,RELOPTR,F;%

```

```

00001730 T 0375
00001731 T 0375
00001732 T 0375
00001733 T 0375
00001734 T 0375

```

START OF SEGMENT \*\*\*\*\* 16

```

STACK(F+2) = STARTSYM
STACK(F+3) = FIRSTSYM
STACK(F+4) = TYPE1
STACK(F+5) = RELOPTR
STACK(F+6) = F

```

BOOLEAN CALLGEN;%

00001735 T 0000

STACK(F+7) = CALLGEN  
%

```
EXPRLEVEL:=EXPRLEVEL+1;%  
IF EXPRLEVEL = 1 THEN%  
BEGIN%  
  PUTDUMMY;%  
  FIRSTSYM := NUMSYMS;%  
END;%  
PUTDUMMY; STARTSYM:=NUMSYMS;%  
PUTDUMMY;%  
SIMPLEEXPRESSION;%  
IF CURSY≥LSSSY AND CURSY≤INBY THEN % "<","≤","≥",">","=","≠","IN"  
BEGIN%  
  TYPE1:=CURTYPE; FI=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);%  
    SYMTAB[STARTSYM]:= "INTST("; PUTSYM(","); CALLGEN:=TRUE;%  
  END;%  
END ELSE%  
IF F$SET THEN%  
BEGIN%  
  IF CURMODE=BITPATTERN THEN%  
  BEGIN SYMTAB[STARTSYM+1]:= " REAL("; PUTSYM(")") END;%  
  IF CURSY=EQLSY OR CURSY=NEQSY THEN%  
  BEGIN PUTSYM(IF CURSY=EQLSY THEN "=" ELSE "≠");%  
  END ELSE%  
  BEGIN%  
    IF CURSY=LEQSY THEN SYMTAB[STARTSYM]:= "INCL1(" ELSE%  
    IF CURSY=GEQSY THEN SYMTAB[STARTSYM]:= "INCL2(" ELSE ERROR(64);%  
    PUTSYM(","); CALLGEN:=TRUE;%  
  END END ELSE%  
IF F$POINTERS THEN%  
BEGIN%  
  IF CURSY=EQLSY THEN PUTSYM("=") ELSE%  
  IF CURSY=NEQSY THEN PUTSYM("≠") ELSE ERROR(64);%  
END ELSE ERROR(64);%  
INSYMBOL;%  
PUTDUMMY; STARTSYM:=NUMSYMS;%  
SIMPLEEXPRESSION;%  
IF CURTYPE>0 AND TYPE1>0 THEN%  
IF CURTYPE≠TYPE1 THEN%  
IF RELOPTR≠INSY THEN%  
BEGIN%  
  IF TYPETAB1[TYPE1].FORM≠NUMERIC OR CURTYPE≠REALTYPE THEN%  
  CHECKTYPES(TYPE1,CURTYPE);%  
END ELSE%  
IF TYPETAB1[CURTYPE].FORM≠SET THEN ERROR(66)%
```

00001736	T	0000
00001737	T	0000
00001738	T	0001
00001739	T	0002
00001740	T	0002
00001741	T	0008
00001742	T	0009
00001743	T	0009
00001744	T	0017
00001745	T	0024
00001746	T	0025
00001747	T	0026
00001748	T	0027
00001749	T	0029
00001750	T	0030
00001751	T	0031
00001752	T	0031
00001753	T	0032
00001754	T	0043
00001755	T	0052
00001756	T	0061
00001757	T	0070
00001758	T	0080
00001759	T	0089
00001760	T	0098
00001761	T	0099
00001762	T	0101
00001763	T	0112
00001764	T	0112
00001765	T	0112
00001766	T	0113
00001767	T	0114
00001768	T	0114
00001769	T	0125
00001770	T	0127
00001771	T	0137
00001772	T	0137
00001773	T	0137
00001774	T	0140
00001775	T	0146
00001776	T	0155
00001777	T	0155
00001778	T	0156
00001779	T	0156
00001780	T	0165
00001781	T	0176
00001782	T	0177
00001783	T	0177
00001784	T	0184
00001785	T	0184
00001786	T	0186
00001787	T	0187
00001788	T	0189
00001789	T	0189
00001790	T	0192
00001791	T	0253
00001792	T	0253



```

ELSE CHECKTYPES(TYPE1,TYPETAB1[CURTYPE],SETTYPE);%
IF CURMODE=BITPATTERN THEN%
BEGIN SYMTAB[STARTSYM]= " REAL("; PUTSYM(")") END;%
IF CALLGEN THEN PUTSYM(")");%
CURTYPE=BOOLTYPE; CURMODE=BITPATTERN;%
END;%
EXPRLEVEL=EXPRLEVEL-1;%
IF EXPRLEVEL=0 THEN%
BEGIN%
IF CURMODE=BITPATTERN THEN%
BEGIN%
SYMTAB[FIRSTSYM]= " REAL(";%
PUTSYM(")");%
END;%
WRITEEXPR;%
END;%
PRT(322) = *DESTINATION POINTER*
END OF EXPRESSION;%

```

```

00001793 T 0256
00001794 T 0318
00001795 T 0318
00001796 T 0329
00001797 T 0337
00001798 T 0339
00001799 T 0339
00001800 T 0341
00001801 T 0342
00001802 T 0342
00001803 T 0343
00001804 T 0343
00001805 T 0345
00001806 T 0353
00001807 T 0353

```

```

00001808 T 0432
00001809 T 0432
16 IS 439 LONG: NEXT SEG 2

```

```

%
%
DEFINE BOOLEXP=#%
BEGIN%
PUTDUMMY; EXPRLEVEL=1; EXPRESSION;%
IF CURTYPE>0 THEN IF CURTYPE#BOOLTYPE THEN ERROR(17);%
IF CURMODE#BITPATTERN THEN%
BEGIN SYMTAB[1]= " B("; PUTSYM(")") END;%
EXPRLEVEL=#0; WRITEEXPR;%
END OF BOOLEXP#;%

```

```

00001810 T 0375
00001811 T 0375
00001812 T 0375
00001813 T 0375
00001814 T 0375
00001815 T 0375
00001816 T 0375
00001817 T 0375
00001818 T 0375
00001819 T 0375

```

```

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
%
%
%
%
%
%
%
%
%
%
%
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
%
%
PROCEDURE CONCAT;
BEGIN
  DEFINE INEXPR=%
  BEGIN INSYMBOL; EXPRESSION;%
    IF CURTYPE>0 THEN%
      IF TYPETAB1[CURTYPE].FORM#NUMERIC THEN ERROR(17);%
    END #;%
  %
  PUTTEXT("CONCAT"); PUTSYM("(");%
  INSYMBOL;%
  IF CURSY=LPAR THEN%
    BEGIN%
      INSYMBOL; EXPRESSION;%
      IF CURTYPE>0 THEN%
        IF TYPETAB1[CURTYPE].FORM#ALFA THEN ERROR(17);%
        IF CURSY=COMMA THEN%
          BEGIN%
            PUTSYM(","); INSYMBOL; EXPRESSION;%
            IF CURTYPE>0 THEN%
              IF TYPETAB1[CURTYPE].FORM#ALFA THEN ERROR(17);%
              IF CURSY=COMMA THEN%
                BEGIN%
                  PUTSYM(","); INEXPR;%
                  IF CURSY=COMMA THEN%
                    BEGIN%
                      PUTSYM(","); INEXPR;%
                      IF CURSY=COMMA THEN%
                        BEGIN%
                          PUTSYM(","); INEXPR;%
                          PUTSYM(","); PUTCONST(CARDCNT);%
                          PUTSYM(")");%
                          IF CURSY#RPAR THEN BEGIN ERROR(3); SKIP(RPAR) END;%
                        END ELSE BEGIN ERROR(3); SKIP(RPAR) END;%
                        END ELSE BEGIN ERROR(3); SKIP(RPAR) END;%
                        END ELSE BEGIN ERROR(3); SKIP(RPAR) END;%
                        END ELSE BEGIN ERROR(3); SKIP(RPAR) END;%
                        END ELSE BEGIN ERROR(3); SKIP(RPAR) END;%
                      CURTYPE:=REALTYPE;%
                      IF CURSY=RPAR THEN INSYMBOL;%
                    END OF CONCAT;%
          END
        END
      END
    END
  %

```

```

00001821 T 0375
00001822 T 0375
00001823 T 0375
00001824 T 0375
00001825 T 0375
00001826 T 0375
00001827 T 0375
00001828 T 0375
00001829 T 0375
00001830 T 0375
00001831 T 0375
00001832 T 0375
00001833 T 0375
00001834 T 0375
00001835 T 0375
START OF SEGMENT ***** 17
00001836 T 0000
00001837 T 0000
00001838 T 0000
00001839 T 0000
00001840 T 0000
00001841 T 0000
00001842 T 0014
00001843 T 0015
00001844 T 0015
00001845 T 0016
00001846 T 0017
00001847 T 0018
00001848 T 0021
00001849 T 0022
00001850 T 0022
00001851 T 0031
00001852 T 0031
00001853 T 0035
00001854 T 0035
00001855 T 0036
00001856 T 0048
00001857 T 0049
00001858 T 0050
00001859 T 0062
00001860 T 0063
00001861 T 0063
00001862 T 0076
00001863 T 0096
00001864 T 0104
00001865 T 0106
00001866 T 0108
00001867 T 0110
00001868 T 0112
00001869 T 0114
00001870 T 0116
00001871 T 0117
00001872 T 0119
17 IS 120 LONG, NEXT SEG 2

```

%

00001873 T 0375

```

%
PROCEDURE PREAD(CHANGELINE);%
PRT(323) = PREAD
VALUE CHANGELINE; BOOLEAN CHANGELINE;%
BEGIN%
    INTEGER FILEID,F;%

STACK(F+2) = FILEID
STACK(F+3) = F
    BOOLEAN CHECK;%
STACK(F+4) = CHECK
    GEN(" BEGIN",7,2);%
    FILEPARAM(INPUTFILE); 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=IDENTIFIER THEN%
            BEGIN%
                SEARCH;%
                IF FOUND THEN%
                BEGIN%
                    IF THISID.IDCLASS=VAR OR%
                    THISID.IDCLASS=CONST AND BOOLEAN(THISID.FORMAL) THEN%
                    BEGIN%
                        VARIABLE; F:=TYPETAB1[CURTYPE],FORM;%
                        IF F=NUMERIC OR F=FLOATING OR F=CHAR THEN%
                        BEGIN%
                            CHECK:=CHECKOPTION AND F#FLOATING;%
                            WRITEEXPR; GEN("=",2,6);%
PRT(324) = *DESTINATION POINTER*
                            IF CHECK THEN GEN("CHECK",6,2);%
                            GEN("PREAD",6,2); GENID("F",FILEID,5); GEN(", ",1,7);%
                            GENID("V",FILEID,5); GEN(", ",1,7);%
                            GENID("I",FILEID,5); GEN(", ",1,7);%
                            IF F=NUMERIC THEN GENINT(2) ELSE%
                            IF F=FLOATING THEN GENINT(3) ELSE GENINT(1);%
                            GEN(", ",1,7); GENINT(CARDCNT); GEN(", ",1,7);%
                            IF CHECK THEN%
                            BEGIN%
                                GEN(", ",1,7); GENINT(TYPETAB2[CURTYPE]); GEN(", ",1,7);%
                                GENINT(TYPETAB3[CURTYPE]); GEN(", ",1,7);%
                                GENINT(CARDCNT); GEN(", ",1,7);%
                            END;%
                            END ELSE BEGIN ERROR(82); INSYMBOL END;%
                            END ELSE BEGIN ERROR(8); INSYMBOL END;%
                            END ELSE BEGIN ERROR(1); INSYMBOL END;%
                            END ELSE ERROR(9);%
                            GEN(", ",1,7);%
                            END UNTIL CURSY#COMMA;%
                            IF CURSY#RPAR THEN BEGIN ERROR(46); SKIP(RPAR) END;%
                            IF CURSY=RPAR THEN INSYMBOL;%
                            END;%
                            IF CHANGELINE THEN%
                            BEGIN%

```

```

00001874 T 0375
00001875 T 0375

00001876 T 0375
00001877 T 0375
00001878 T 0375
START OF SEGMENT ***** 18

00001879 T 0000

00001880 T 0000
00001881 T 0009
00001882 T 0062
00001883 T 0065
00001884 T 0066
00001885 T 0066
00001886 T 0067
00001887 T 0068
00001888 T 0070
00001889 T 0071
00001890 T 0071
00001891 T 0100
00001892 T 0100
00001893 T 0100
00001894 T 0102
00001895 T 0104
00001896 T 0105
00001897 T 0107
00001898 T 0109
00001899 T 0110
00001900 T 0112

00001901 T 0199
00001902 T 0209
00001903 T 0239
00001904 T 0260
00001905 T 0281
00001906 T 0329
00001907 T 0418
00001908 T 0480
00001909 T 0480
00001910 T 0481
00001911 T 0543
00001912 T 0596
00001913 T 0649
00001914 T 0649
00001915 T 0652
00001916 T 0655
00001917 T 0658
00001918 T 0660
00001919 T 0669
00001920 T 0671
00001921 T 0673
00001922 T 0675
00001923 T 0675
00001924 T 0676

```

```

GEN("RLINE(",6,2); GENID("F",FILEID,5); GEN(" ",1,7);%
GENID("V",FILEID,5); GEN(" ",1,7);%
GENID("I",FILEID,5); GEN(")",1,7);%
END;%
GEN("END",4,5);%
END OF PREAD;%

```

```

00001925 T 0676
00001926 T 0706
00001927 T 0727
00001928 T 0748
00001929 T 0748
00001930 T 0757
18 IS 773 LONG, NEXT SEG 2

```

```

%
%
PROCEDURE PWRITE(LINEFEED);%
PRT(325) = PWRITE
VALUE LINEFEED; BOOLEAN LINEFEED;%
BEGIN%
INTEGER FILEID,F,I,LASTSY;%

```

```

00001931 T 0375
00001932 T 0375
00001933 T 0375
00001934 T 0375
00001935 T 0375
00001936 T 0375

```

```

STACK(F+2) = FILEID
STACK(F+3) = F
STACK(F+4) = I
STACK(F+5) = LASTSY

```

```

START OF SEGMENT ***** 19

```

```

STACK(F+6) = P
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);%
PRT(326) = *DESTINATION POINTER*
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 ISCURLLENGTH THEN%
BEGIN%
IF ALGOLCNT<10 THEN WRITEALGOL;%
REPLACE ALGOLPNT;ALGOLPNT BY " ", PIP FOR 7, " ", " ,";%
ALGOLCNT:=ALGOLCNT+1;%
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; FI=TYPETAB1[CURTYPE].FORM;%
GEN(" ",1,7);%

```

```

00001937 T 0000
00001938 T 0000
00001939 T 0009
00001940 T 0062
00001941 T 0065
00001942 T 0066
00001943 T 0066
00001944 T 0067
00001945 T 0068
00001946 T 0070
00001947 T 0072
00001948 T 0072
00001949 T 0102
00001950 T 0123
00001951 T 0144
00001952 T 0145
00001953 T 0147
00001954 T 0147
00001955 T 0148
00001956 T 0150
00001957 T 0161
00001958 T 0163
00001959 T 0174
00001960 T 0227
00001961 T 0280
00001962 T 0280
00001963 T 0280
00001964 T 0281
00001965 T 0311
00001966 T 0332
00001967 T 0353
00001968 T 0354
00001969 T 0356

```

```

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%
  BEGIN%
    IF F=FLOATING THEN GENINT(16) ELSE%
    IF F=ALFA AND LASTSY=ALFACONST 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;%
  PRT(327) = *SEGMENT DESCRIPTOR*
  STACK(F+7) = 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;%
  PRT(330) = *SEGMENT DESCRIPTOR*
  GEN("END",4,5);%
  END OF PWRITE;%

```

```

00001970 T 0365
00001971 T 0368
00001972 T 0369
00001973 T 0370
00001974 T 0416
00001975 T 0461
00001976 T 0506
00001977 T 0595
00001978 T 0604
00001979 T 0605
00001980 T 0605
00001981 T 0606
00001982 T 0609
00001983 T 0618
00001984 T 0619
00001985 T 0619
00001986 T 0621
00001987 T 0622
00001988 T 0625
00001989 T 0634
00001990 T 0643
00001991 T 0643
00001992 T 0646
00001993 T 0691
00001994 T 0737
00001995 T 0782
00001996 T 0871
00001997 T 0880
00001998 T 0880
00001999 T 0882
00002000 T 0935
00002001 T 0935
00002002 T 0945
00002003 T 0946
00002004 T 0949
00002005 T 0951
00002006 T 0951
00002007 T 0952
00002008 T 0953
00002009 T 0953

```

```

START OF SEGMENT ***** 20
00002010 T 0000
00002011 T 0030
00002012 T 0051
00002013 T 0072
20 IS 74 LONG, NEXT SEG 19
00002014 T 0954
00002015 T 0963
19 IS 975 LONG, NEXT SEG 2

```

%

00002016 T 0375

```

%
PROCEDURE FILEHANDLING(PROCNUM);
PRT(331) = FILEHANDLING
VALUE PROCNUM; INTEGER PROCNUM;
BEGIN
    INTEGER F;

```

```

STACK(F+2) = F

```

```

CASE PROCNUM OF
BEGIN ;
PRT(332) = *CASE STATEMENT DESCRIPTOR*
    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);%
PRT(333) = *DESTINATION POINTER*
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;%
PRT(334) = PACK
BEGIN%
    INTEGER IT,T;%

```

```

STACK(F+2) = IT
STACK(F+3) = T

```

```

GEN("PACK",5,3);%
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];%

```

```

%*** FILE HANDLING PROCEDURES:

```

```

%***
%*** 1) PUT
%*** 2) GET

```

```

%*** 3) RESET
%*** 4) REWRITE
%*** 5) PAGE

```

```

00002017 T 0375
00002018 T 0375

```

```

00002019 T 0375
00002020 T 0375
00002021 T 0375

```

```

START OF SEGMENT ***** 21

```

```

00002022 T 0000
00002023 T 0000

```

```

00002024 T 0000
00002025 T 0010
00002026 T 0019
00002027 T 0029
00002028 T 0038
00002029 T 0048

```

```

START OF SEGMENT ***** 22
22 IS 7 LONG, NEXT SEG 21

```

```

00002030 T 0048
00002031 T 0114
00002032 T 0116
00002033 T 0118
00002034 T 0121

```

```

00002035 T 0142
00002036 T 0163
00002037 T 0184
00002038 T 0236
00002039 T 0239
00002040 T 0240

```

```

21 IS 243 LONG, NEXT SEG 2

```

```

00002041 T 0375
00002042 T 0375
00002043 T 0375

```

```

00002044 T 0375
00002045 T 0375

```

```

START OF SEGMENT ***** 23

```

```

00002046 T 0000
00002047 T 0009
00002048 T 0009
00002049 T 0010
00002050 T 0010
00002051 T 0011
00002052 T 0012
00002053 T 0012
00002054 T 0042
00002055 T 0043
00002056 T 0043
00002057 T 0044
00002058 T 0045

```

```

IF T,FORM=ARRAYS THEN%
BEGIN%
  IT:=T,INXTYPE;%
  IF TYPETAB1[T,ARRTYPE],FORM#CHAR THEN ERROR(88);%
  GENID("V",1000*THISLEVEL+THISINDEX,5);%
PRT(335) = *DESTINATION POINTER*
  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;%
      END
    END
  END

```

```

00002059 T 0046
00002060 T 0048
00002061 T 0048
00002062 T 0049
00002063 T 0053
00002064 T 0066
00002065 T 0069
00002066 T 0123
00002067 T 0177
00002068 T 0178
00002069 T 0179
00002070 T 0181
00002071 T 0182
00002072 T 0182
00002073 T 0183
00002074 T 0184
00002075 T 0193
00002076 T 0253
00002077 T 0253
00002078 T 0254
00002079 T 0263
00002080 T 0263
00002081 T 0264
00002082 T 0265
00002083 T 0294
00002084 T 0294
00002085 T 0294
00002086 T 0296
00002087 T 0298
00002088 T 0299
00002089 T 0378
00002090 T 0379
00002091 T 0382
00002092 T 0383
00002093 T 0385
00002094 T 0386
00002095 T 0388
00002096 T 0390
00002097 T 0393
00002098 T 0394
00002099 T 0397
00002100 T 0459

```

23 IS 465 LONG, NEXT SEG 2

```

%
%
PROCEDURE UNPACK;%
PRT(336) = UNPACK
BEGIN%
  INTEGER IT,T;%

```

```

00002101 T 0375
00002102 T 0375
00002103 T 0375

```

```

STACK(F+2) = IT
STACK(F+3) = T

```

```

00002104 T 0375
00002105 T 0375
START OF SEGMENT ***** 24

```

```

GEN("UNPACK(",7,1); INSYMBOL;%
IF CURSY=LPAR THEN%
BEGIN%
INSYMBOL; EXPRESSION;%
IF CURTYPE>0 THEN IF TYPETAB1[CURTYPE],FORM#ALFA THEN ERROR(17);%
IF CURSY=COMMA THEN%
BEGIN%
GEN("",1,7); 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);%
IF THISLEVEL>1 AND THISLEVEL#CURLEVEL THEN ERROR(5);%
GENID("V",1000*THISLEVEL+THISINDEX,5);%
PRT(337) = *DESTINATION POINTER*
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);%
END ELSE BEGIN ERROR(89); SKIP(RPAR) END;%
END ELSE BEGIN ERROR(89); SKIP(RPAR) END;%
IF CURSY#RPAR THEN BEGIN ERROR(89); 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 UNPACK;%

```

```

00002106 T 0000
00002107 T 0009
00002108 T 0010
00002109 T 0010
00002110 T 0011
00002111 T 0015
00002112 T 0016
00002113 T 0017
00002114 T 0026
00002115 T 0027
00002116 T 0027
00002117 T 0057
00002118 T 0058
00002119 T 0058
00002120 T 0059
00002121 T 0060
00002122 T 0061
00002123 T 0063
00002124 T 0063
00002125 T 0064
00002126 T 0068
00002127 T 0071
00002128 T 0084
00002129 T 0138
00002130 T 0192
00002131 T 0193
00002132 T 0194
00002133 T 0196
00002134 T 0197
00002135 T 0197
00002136 T 0198
00002137 T 0199
00002138 T 0208
00002139 T 0268
00002140 T 0270
00002141 T 0272
00002142 T 0274
00002143 T 0276
00002144 T 0278
00002145 T 0340

```

24 IS 344 LONG, NEXT SEG 2

```

%
%
PROCEDURE NEWDISP; ***** "NEW", "DISPOSE"
PRT(340) = NEWDISP
BEGIN%
INTEGER T1;%
STACK(F+2) = T1
IF CURNAME1="3000NEW" THEN GEN("NEW(",4,4) ELSE%
BEGIN GEN("DISPOSE",7,1); GEN(")",1,7) END;%
INSYMBOL;%

```

```

00002146 T 0375
00002147 T 0375
00002148 T 0375

```

```

00002149 T 0375
00002150 T 0375

```

START OF SEGMENT \*\*\*\*\* 25

```

00002151 T 0000
00002152 T 0010
00002153 T 0031

```



```

IF CURSY=LPAR THEN%
BEGIN%
INSYMBOL;%
IF CURSY=IDENTIFIER THEN%
BEGIN%
SEARCH;%
IF FOUND THEN%
BEGIN%
VARIABLE;%
IF CURTYPE>0 THEN IF TYPETAB1[CURTYPE],FORM=POINTERS THEN%
BEGIN%
WRITEEXPR; GEN(", ",1,7);%
PRT(341) = *DESTINATION POINTER*
T1:=TYPETAB1[CURTYPE],POINTTYPE;%
T1:=TYPETAB1[T1],SIZE;%
IF T1>1023 THEN ERROR(86);%
GENINT(T1); GEN(", ",1,7);%
END ELSE ERROR(81);%
END ELSE BEGIN ERROR(1); INSYMBOL END;%
END ELSE ERROR(9);%
WHILE CURSY=COMMA DO%
BEGIN INSYMBOL;%
IF CURSY NEQ IDENTIFIER THEN ERROR(9);%
IF CURSY NEQ RPAR THEN INSYMBOL;%
END;%
END ELSE BEGIN ERROR(58); SKIP(RPAR) END;%
IF CURSY#RPAR THEN BEGIN ERROR(46); SKIP(RPAR) END;%
IF CURSY=RPAR THEN INSYMBOL;%
END OF NEWDISP;%

```

```

00002154 T 0031
00002155 T 0032
00002156 T 0032
00002157 T 0033
00002158 T 0034
00002159 T 0034
00002160 T 0064
00002161 T 0065
00002162 T 0065
00002163 T 0066
00002164 T 0068
00002165 T 0069
00002166 T 0156
00002167 T 0158
00002168 T 0159
00002169 T 0161
00002170 T 0214
00002171 T 0215
00002172 T 0217
00002173 T 0218
00002174 T 0220
00002175 T 0220
00002176 T 0222
00002177 T 0224
00002178 T 0225
00002179 T 0227
00002180 T 0229
00002181 T 0231
25 IS 234 LONG, NEXT SEG 2

```



EXIT;%  
END OF ASSIGNMENT;%

00002236 T 0345  
00002237 T 0345  
26 IS 352 LONG, NEXT SEG 2

%  
%  
PROCEDURE COMPSTAT;%  
PRT(345) = COMPSTAT  
BEGIN%  
INTEGER BEGINNUM;%  
  
STACK(F+2) = BEGINNUM  
LABEL STATM;%  
%  
BEGINNUM:=NUMBEGINS:=NUMBEGINS+1; MARGIN(" B",BEGINNUM);%  
PRT(346) = \*DESTINATION POINTER\*  
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;%

00002238 T 0375  
00002239 T 0375  
00002240 T 0375  
  
00002241 T 0375  
00002242 T 0375  
START OF SEGMENT \*\*\*\*\* 27

00002243 T 0000  
00002244 T 0000  
00002245 T 0000  
  
00002246 T 0020  
00002247 T 0029  
00002248 T 0029  
00002249 T 0031  
00002250 T 0032  
00002251 T 0041  
00002252 T 0047  
00002253 T 0049  
00002254 T 0051  
00002255 T 0051  
00002256 T 0053  
00002257 T 0056  
00002258 T 0056  
00002259 T 0083  
00002260 T 0083  
27 IS 88 LONG, NEXT SEG 2

%  
%  
PROCEDURE IFSTAT;%  
PRT(347) = IFSTAT  
BEGIN%  
LABEL EXIT;%  
  
GEN("IF",3,6);%  
INSYMBOL; BOOLEXPRESS;%  
PRT(350) = \*DESTINATION POINTER\*  
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;%  
GEN(" THEN",6,3);%

00002261 T 0375  
00002262 T 0375  
00002263 T 0375  
  
00002264 T 0375  
00002265 T 0375  
START OF SEGMENT \*\*\*\*\* 28  
00002266 T 0000  
00002267 T 0009  
  
00002268 T 0112  
00002269 T 0113  
00002270 T 0115  
00002271 T 0116  
00002272 T 0117  
00002273 T 0119  
00002274 T 0121  
00002275 T 0121

```

INSYMBOL; STATEMENT;%
IF CURSY=ELSESY THEN%
BEGIN GEN(" ELSE",6,3); INSYMBOL; STATEMENT END;%
EXIT;%
END OF IFSTAT;%

```

```

00002276 T 0130
00002277 T 0131
00002278 T 0132
00002279 T 0142
00002280 T 0143
28 IS 146 LONG, NEXT SEG 2

```

```

%
%
PROCEDURE CASESTAT;%
PRT(351) = CASESTAT
BEGIN%
DEFINE CASEHASH(N)=(N),[38]9] MOD MAXCASES;%
INTEGER ARRAY CASETAB[0:MAXCASES];%
INTEGER CASENUM,CASETYPE,NCASELABS,TEMPVARNUM,CONVAL,CONTYPE,C,T;%
STACK(F+2) = CASETAB
STACK(F+3) = CASENUM
STACK(F+4) = CASETYPE
STACK(F+5) = NCASELABS
STACK(F+6) = TEMPVARNUM
STACK(F+7) = CONVAL
STACK(F+10) = CONTYPE
STACK(F+11) = C
STACK(F+12) = T
BOOLEAN ZEROLAB,FIRST;%
STACK(F+13) = ZEROLAB
STACK(F+14) = FIRST
%

```

```

00002281 T 0375
00002282 T 0375
00002283 T 0375
00002284 T 0375
00002285 T 0375
START OF SEGMENT ***** 29
00002286 T 0000
00002287 T 0001
00002288 T 0001

```

```

CASENUM:=NUMCASES:=NUMCASES+1; MARGIN("CB",CASENUM);%
PRT(352) = *DESTINATION POINTER*
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 TYPETAB[CASETYPE],FORMZFLOATING 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%

```

```

00002289 T 0001
00002290 T 0001
00002291 T 0021
00002292 T 0023
00002293 T 0025
00002294 T 0055
00002295 T 0056
00002296 T 0066
00002297 T 0067
00002298 T 0069
00002299 T 0070
00002300 T 0073
00002301 T 0073
00002302 T 0075
00002303 T 0080
00002304 T 0081
00002305 T 0081
00002306 T 0083
00002307 T 0084
00002308 T 0084
00002309 T 0085
00002310 T 0094
00002311 T 0095
00002312 T 0098
00002313 T 0186

```

```

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(" OR",4,5);%
  END;%
  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;%

```

```

00002314 T 0186
00002315 T 0187
00002316 T 0189
00002317 T 0249
00002318 T 0313
00002319 T 0314
00002320 T 0315
00002321 T 0315
00002322 T 0316
00002323 T 0319
00002324 T 0320
00002325 T 0322
00002326 T 0326
00002327 T 0329
00002328 T 0333
00002329 T 0333
00002330 T 0336
00002331 T 0346
00002332 T 0346
00002333 T 0347
00002334 T 0356
00002335 T 0359
00002336 T 0361
00002337 T 0361
00002338 T 0363
00002339 T 0365
00002340 T 0365
00002341 T 0375
00002342 T 0376
00002343 T 0379
00002344 T 0406
00002345 T 0408
00002346 T 0408

```

29 IS 423 LONG, NEXT SEG 2

```

%
%
PROCEDURE WHILESTAT;%
PRT(353) = WHILESTAT
BEGIN%
  LABEL STATM,EXIT;%

  GEN("WHILE",6,3);%
  INSYMBOL; BOOLEXPRESS;%
PRT(354) = *DESTINATION POINTER*
  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;%
PRT(355) = STATM
PRT(356) = EXIT

END; END;%

```

```

00002347 T 0375
00002348 T 0375
00002349 T 0375

00002350 T 0375
00002351 T 0375
START OF SEGMENT ***** 30
00002352 T 0000
00002353 T 0009

00002354 T 0112
00002355 T 0113
00002356 T 0115
00002357 T 0116
00002358 T 0117
00002359 T 0119

00002360 T 0125

```

```

GEN(" DO",4,5);%
INSYMBOL;%
STATM: STATEMENT;%
EXIT;%
END OF WHILESTAT;%

```

```

00002361 T 0125
00002362 T 0134
00002363 T 0134
00002364 T 0135
00002365 T 0136
30 IS 142 LONG, NEXT SEG 2

```

```

%
%
PROCEDURE REPEATSTAT;%
PRT(357) = REPEATSTAT
BEGIN%
INTEGER REPNUM;%

```

```

00002366 T 0375
00002367 T 0375
00002368 T 0375

```

```

STACK(F+2) = REPNUM

```

```

00002369 T 0375
00002370 T 0375
START OF SEGMENT ***** 31

```

```

LABEL NEWTRY;%
%
REPNUM:=NUMREPS:=NUMREPS+1;%
MARGIN(" R",REPNUM);%
PRT(360) = *DESTINATION POINTER*
GEN("DO",3,6); GEN("BEGIN",6,3);%
DO BEGIN%

```

```

00002371 T 0000
00002372 T 0000
00002373 T 0000
00002374 T 0001

```

```

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;%
END;%
GEN(" END",5,4); GEN("UNTIL",6,3); MARGIN(" U",REPNUM);%
INSYMBOL; BOOLEXP;
END OF REPEATSTAT;%

```

```

00002375 T 0020
00002376 T 0038
00002377 T 0038
00002378 T 0038
00002379 T 0039
00002380 T 0048
00002381 T 0055
00002382 T 0057
00002383 T 0059
00002384 T 0059
00002385 T 0060
00002386 T 0061
00002387 T 0063
00002388 T 0065
00002389 T 0066
00002390 T 0066
00002391 T 0102
00002392 T 0208

```

```

31 IS 211 LONG, NEXT SEG 2

```

```

%
%
PROCEDURE FORSTAT;%
PRT(361) = FORSTAT
BEGIN%
INTEGER VARTYPE,VARNUM,LLIM,ULIM;%

```

```

00002393 T 0375
00002394 T 0375
00002395 T 0375

```

```

STACK(F+2) = VARTYPE
STACK(F+3) = VARNUM
STACK(F+4) = LLIM
STACK(F+5) = ULIM

```

```

00002396 T 0375
00002397 T 0375
START OF SEGMENT ***** 32

```

STACK(F+6) = DOWN

```
BOOLEAN DOWN;%  
LABEL STATM;%  
%  
GEN("BEGIN",6,3);%  
INSYMBOL;%  
IF CURSY=IDENTIFIER THEN%  
BEGIN%  
  SEARCH;%  
  IF FOUND THEN%  
  BEGIN%  
    VARNUM:=1000*THISLEVEL+THISINDEX;%  
    IF THISID.IDCLASS=VAR OR%  
      THISID.IDCLASS=CONST AND BOOLEAN(THISID.FORMAL) THEN%  
    BEGIN%  
      IF THISLEVEL>1 AND THISLEVEL<CURLEVEL THEN ERROR(5);%  
      IF THISLEVEL>CURLEVEL THEN ERROR(83);%  
      VARTYPE:=THISID.TYPE;%  
      IF TYPETAB1[VARTYPE].FORMSCHAR THEN%  
      BEGIN%  
        LLIM:=TYPETAB2[VARTYPE]; ULIM:=TYPETAB3[VARTYPE];%  
        END ELSE BEGIN ERROR(12); VARTYPE:=0 END;%  
      END ELSE ERROR(8);%  
    END ELSE ERROR(1);%  
  END ELSE ERROR(9);%  
  INSYMBOL;%  
  IF CURSY=ASSIGNSY THEN%  
  BEGIN ERROR(28);%  
    SKIP(ASSIGNSY);%  
    IF CURSY=ASSIGNSY THEN INSYMBOL ELSE%  
    IF SYMKIND[CURSY]=INITIAL THEN GO TO STATM;%  
  END ELSE INSYMBOL;%  
  GENID("V",VARNUM,5); GEN(" ",1,7);%  
PRT(362) = *DESTINATION POINTER*  
  IF CHECKOPTION THEN CHECKEXPR(LLIM,ULIM) ELSE EXPRESSION;%  
  WRITEEXPR;%  
  GEN(" ",1,7);%  
  IF VARTYPE=0 THEN VARTYPE:=CURTYPE ELSE CHECKTYPES(VARTYPE,CURTYPE);  
  NUMTEMPS:=NUMTEMPS+1; IF NUMTEMPS>MAXTEMPS THEN ERROR(16);%  
  IF CURSY=TOSY THEN INSYMBOL ELSE%  
  IF CURSY=DOWNTOSY THEN BEGIN DOWN:=TRUE; INSYMBOL END ELSE%  
  BEGIN IF CURTYPE>0 THEN ERROR(23);%  
    SKIP(TOSY);%  
    IF CURSY=TOSY THEN INSYMBOL ELSE%  
    BEGIN IF CURTYPE=0 THEN ERROR(23);%  
      IF SYMKIND[CURSY]=INITIAL THEN GO TO STATM;%  
    END; END;%  
  GENID("T",NUMTEMPS,2); GEN(" ",1,7);%  
  IF CHECKOPTION THEN CHECKEXPR(LLIM,ULIM) ELSE EXPRESSION;%  
  WRITEEXPR;%  
  GEN(" ",1,7);%  
  IF VARTYPE=0 THEN VARTYPE:=CURTYPE ELSE CHECKTYPES(VARTYPE,CURTYPE);  
  IF CURSY=DOSY THEN%  
  BEGIN IF CURTYPE>0 THEN ERROR(19);%  
    SKIP(DOSY);%  
    IF CURSY=DOSY THEN INSYMBOL ELSE%  
    IF CURTYPE=0 THEN ERROR(19);%
```

00002398	T	0000
00002399	T	0000
00002400	T	0000
00002401	T	0000
00002402	T	0009
00002403	T	0009
00002404	T	0010
00002405	T	0010
00002406	T	0040
00002407	T	0041
00002408	T	0041
00002409	T	0043
00002410	T	0044
00002411	T	0047
00002412	T	0047
00002413	T	0050
00002414	T	0052
00002415	T	0053
00002416	T	0055
00002417	T	0055
00002418	T	0057
00002419	T	0059
00002420	T	0061
00002421	T	0062
00002422	T	0063
00002423	T	0064
00002424	T	0064
00002425	T	0066
00002426	T	0066
00002427	T	0068
00002428	T	0070
00002429	T	0071
00002430	T	0092
00002431	T	0170
00002432	T	0248
00002433	T	0257
00002434	T	0319
00002435	T	0322
00002436	T	0324
00002437	T	0327
00002438	T	0329
00002439	T	0330
00002440	T	0332
00002441	T	0334
00002442	T	0336
00002443	T	0336
00002444	T	0357
00002445	T	0434
00002446	T	0512
00002447	T	0521
00002448	T	0583
00002449	T	0583
00002450	T	0586
00002451	T	0587
00002452	T	0588

```

END ELSE INSYMBOL;%
GEN("FOR",4,5); GENID("V",VARNUM,5); GEN("=",1,7);%
GENID("V",VARNUM,5); GEN(" ",1,7);%
IF DOWN THEN GEN("DOWNT0",7,2) ELSE GEN("UPT0",5,4);%
GENID("T",NUMTEMPS,2); GEN(" DO",4,5);%
STAT% STATEMENT;%
GEN(" END",5,4);%
NUMTEMPS:=NUMTEMPS-1;%
END OF FORSTAT;%

```

```

00002453 T 0591
00002454 T 0592
00002455 T 0622
00002456 T 0643
00002457 T 0665
00002458 T 0686
00002459 T 0686
00002460 T 0695
00002461 T 0696
32 IS 703 LONG, NEXT SEG 2

```

```

%
%
PROCEDURE GOTOSTAT;%
PRT(363) = GOTOSTAT
BEGIN%
INTEGER I;%

STACK(F+2) = I

INSYMBOL;%
IF CURSY=INTCONST THEN%
BEGIN I:=NUMLABS;%
WHILE I>1 AND LABTAB[I],LABVAL#CURVAL DO I:=I-1;%
IF I=0 THEN ERROR(15);%
GEN("GO",3,6); GENID("L",CURVAL,4);%
PRT(364) = *DESTINATION POINTER*
INSYMBOL;%
END ELSE ERROR(10);%
END OF GOTOSTAT;%

```

```

00002462 T 0375
00002463 T 0375
00002464 T 0375

00002465 T 0375
00002466 T 0375
START OF SEGMENT ***** 33

00002467 T 0000
00002468 T 0000
00002469 T 0001
00002470 T 0002
00002471 T 0007
00002472 T 0009

00002473 T 0030
00002474 T 0031
00002475 T 0033
33 IS 36 LONG, NEXT SEG 2

```

```

%
%
PROCEDURE WITHSTAT;%
PRT(365) = WITHSTAT
BEGIN%
INTEGER STARTLEVEL,VERYFIRSTWITHSYM,I;%

STACK(F+2) = STARTLEVEL
STACK(F+3) = VERYFIRSTWITHSYM
STACK(F+4) = I

REAL D;%

STACK(F+5) = D

STARTLEVEL:=TOPLEVEL; VERYFIRSTWITHSYM:=NWITHSYMS;%
DO BEGIN%
INSYMBOL;%
IF CURSY=IDENTIFIER THEN%
BEGIN%
SEARCH;%
IF FOUND THEN%
BEGIN%

```

```

00002476 T 0375
00002477 T 0375
00002478 T 0375

00002479 T 0375
00002480 T 0375
START OF SEGMENT ***** 34

00002481 T 0000
00002482 T 0000
00002483 T 0001
00002484 T 0002
00002485 T 0002
00002486 T 0003
00002487 T 0003
00002488 T 0032
00002489 T 0032

```



```

IF THISID.IDCLASS=VAR THEN%
BEGIN%
  VARIABLE;%
  IF CURTYPE>0 THEN%
  IF TYPETAB1[CURTYPE],FORM#RECORD THEN ERROR(98);%
  IF SIMPLEVARIABLE THEN%
  BEGIN PUTSYM("["); INSIDEBRACKETS:=TRUE END;%
  IF TOPLEVEL<MAXLEVEL THEN%
  BEGIN%
    TOPLEVEL:=TOPLEVEL+1;%
    D,NAMETAB:=TYPETAB1[CURTYPE],RECTAB;%
    D,RECTYPE:=CURTYPE;%
    D,NUMPNTRSINWITH:=NUMPOINTERS;%
    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+I]:=SYMTAB[I];%
      NWITHSYMS:=NWITHSYMS+1;%
    END;%
    D,LASTWITHSYM:=NWITHSYMS-1;%
    DISPLAY[TOPLEVEL,I]=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;%

```

```

00002490 T 0032
00002491 T 0034
00002492 T 0034
00002493 T 0035
00002494 T 0035
00002495 T 0039
00002496 T 0039
00002497 T 0048
00002498 T 0048
00002499 T 0049
00002500 T 0050
00002501 T 0053
00002502 T 0054
00002503 T 0056
00002504 T 0058
00002505 T 0060
00002506 T 0062
00002507 T 0064
00002508 T 0064
00002509 T 0065
00002510 T 0066
00002511 T 0069
00002512 T 0071
00002513 T 0072
00002514 T 0073
00002515 T 0075
00002516 T 0077
00002517 T 0079
00002518 T 0079
00002519 T 0080
00002520 T 0081
00002521 T 0082
00002522 T 0084
00002523 T 0086
00002524 T 0087
00002525 T 0087
00002526 T 0089

```

34 IS 93 LONG, NEXT SEG 2

```

%
%
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;%

```

```

00002527 T 0375
00002528 T 0375
00002529 T 0375
00002530 T 0375
00002531 T 0375
00002532 T 0000
00002533 T 0000
00002534 T 0000
00002535 T 0000
00002536 T 0001
00002537 T 0002
00002538 T 0003
00002539 T 0006

```

START OF SEGMENT \*\*\*\*\* 35

STACK(F+2) = 1

```

        GO TO LABFOUND;%
    END;%
    ERROR(15);%
    LABFOUND: GENID("L",CURVAL,4); GEN(":",1,7);%
PRT(366) = *DESTINATION POINTER*
    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%
        THISID.IDCLASS#FUNC THEN ASSIGNMENT ELSE%
        IF THISID.IDCLASS#PROC THEN%
        BEGIN%
            IF THISLEVEL=0 THEN          % *** INTRINSIC PROCEDURE ***
            BEGIN%
                IF CURNAME1#"5WRITE" THEN PWRITE(FALSE) ELSE%
                IF CURNAME1#"7WRITEL" AND%
                CURNAME2#"000000N" THEN PWRITE(TRUE) ELSE%
                IF CURNAME1#"400READ" THEN PREAD(FALSE) ELSE%
                IF CURNAME1#"6READLN" THEN PREAD(TRUE) ELSE%
                IF CURNAME1#"400PAGE" THEN FILEHANDLING(5) ELSE%
                IF CURNAME1#"300GET" THEN FILEHANDLING(2) ELSE%
                IF CURNAME1#"300PUT" THEN FILEHANDLING(1) ELSE%
                IF CURNAME1#"50RESET" THEN FILEHANDLING(3) ELSE%
                IF CURNAME1#"7REWRI" AND%
                CURNAME2#"000000E" THEN FILEHANDLING(4) ELSE%
                IF CURNAME1#"300NEW" THEN NEWDISP ELSE%
                IF CURNAME1#"7DISPOS" AND%
                CURNAME2#"000000E" THEN NEWDISP ELSE%
                IF CURNAME1#"400PACK" THEN PACK ELSE%
                IF CURNAME1#"6UNPACK" THEN UNPACK ELSE ERROR(0);%
            END ELSE PASSPARAMS;%
            WRITEEXPR;%
        END ELSE BEGIN ERROR(13); SKIP(99) END;%
        END ELSE BEGIN ERROR(1); ASSIGNMENT END;%
    END OF IDENTIFIER ELSE%
    IF CURSY#BEGINSY THEN COMPSTAT ELSE%
    IF CURSY#IFSY THEN IFSTAT ELSE%
    IF CURSY#CASESY THEN CASESTAT ELSE%
    IF CURSY#WHILESY THEN WHILESTAT ELSE%
    IF CURSY#REPEATSY THEN REPEATSTAT ELSE%
    IF CURSY#FORSY THEN FORSTAT ELSE%
    IF CURSY#WITHSY THEN WITHSTAT ELSE%
    IF CURSY#GOTOSY THEN GOTOSTAT ELSE%
    IF SYMKIND[CURSY]#TERMINAL THEN%
    BEGIN ERROR(13); INSYMBOL; SKIP(SEMICOLON) END;%

```

```

00002540 T 0009
00002541 T 0009
00002542 T 0012
00002543 T 0012
00002544 T 0034
00002545 T 0034
00002546 T 0035
00002547 T 0036
00002548 T 0039
00002549 T 0040
00002550 T 0040
00002551 T 0040
00002552 T 0040
00002553 T 0040
00002554 T 0040
00002555 T 0041
00002556 T 0070
00002557 T 0070
00002558 T 0070
00002559 T 0072
00002560 T 0074
00002561 T 0077
00002562 T 0078
00002563 T 0079
00002564 T 0080
00002565 T 0080
00002566 T 0082
00002567 T 0084
00002568 T 0087
00002569 T 0091
00002570 T 0095
00002571 T 0099
00002572 T 0103
00002573 T 0107
00002574 T 0111
00002575 T 0113
00002576 T 0116
00002577 T 0119
00002578 T 0122
00002579 T 0124
00002580 T 0128
00002581 T 0135
00002582 T 0136
00002583 T 0215
00002584 T 0217
00002585 T 0219
00002586 T 0219
00002587 T 0221
00002588 T 0223
00002589 T 0226
00002590 T 0228
00002591 T 0230
00002592 T 0232
00002593 T 0235
00002594 T 0237
00002595 T 0238

```

END OF STATEMENT:8

00002596 T 0241  
35 IS 244 LONG, NEXT SEG 2

```

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
%
%
%
%
%
%
%
%
%
%
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
%
%
REAL VALX1,VALX2;%
PRT(367) = VALX1
PRT(370) = VALX2
INTEGER TYPEX1,TYPEX2;%
PRT(371) = TYPEX1
PRT(372) = TYPEX2
BOOLEAN PACKED;%
PRT(373) = PACKED
%
PROCEDURE FIELDLIST(RECTAB,FIRSTADDR,LASTADDR);%
PRT(374) = FIELDLIST
VALUE RECTAB,FIRSTADDR;%
INTEGER RECTAB,FIRSTADDR,LASTADDR;%
FORWARD;%
%
DEFINE SUBRANGE=
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;%
NEWTYPET 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);%
PRT(375) = TYPEDECL
INTEGER TTYPE,TSIZE;%
BEGIN%
PROCEDURE TYPERR(ERRNUM,TTYPE,TSIZE);%
PRT(376) = TYPERR
VALUE ERRNUM;%
INTEGER ERRNUM,TTYPE,TSIZE;%
BEGIN ERROR(ERRNUM);%
TTYPE:=TSIZE:=0;%
END;%

```

```

00002598 T 0375
00002599 T 0375
00002600 T 0375
00002601 T 0375
00002602 T 0375
00002603 T 0375
00002604 T 0375
00002605 T 0375
00002606 T 0375
00002607 T 0375
00002608 T 0375
00002609 T 0375
00002610 T 0375
00002611 T 0375
00002612 T 0375
00002613 T 0375
00002614 T 0375
00002615 T 0375
00002616 T 0375
00002617 T 0375
00002618 T 0375
00002619 T 0375
00002620 T 0375
00002621 T 0375
00002622 T 0375
00002623 T 0375
00002624 T 0375
00002625 T 0375
00002626 T 0375
00002627 T 0375
00002628 T 0375
00002629 T 0375
00002630 T 0375
00002631 T 0375
00002632 T 0375
00002633 T 0375
00002634 T 0375
00002635 T 0375
00002636 T 0375
00002637 T 0375
00002638 T 0375
00002639 T 0375
00002640 T 0375
00002641 T 0000
00002642 T 0000
00002643 T 0000
00002644 T 0000
00002645 T 0002

```

START OF SEGMENT \*\*\*\*\* 36

```

%
INTEGER RECINX,ARRSTRUCT,TX,SX,T1,T2,T3,T,N)%
STACK(F+2) = RECINX
STACK(F+3) = ARRSTRUCT
STACK(F+4) = TX
STACK(F+5) = SX
STACK(F+6) = T1
STACK(F+7) = T2
STACK(F+10) = T3
STACK(F+11) = T
STACK(F+12) = N
%
STACK(F+13) = FIRST
%
BOOLEAN FIRST)%
PACKED:=FALSE)%
IF CURSY=IDENTIFIER THEN
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))%
STACK(F+14) = *DISPLAY POINTER*
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))%
T3,INFO:=N; NAMETAB3[CURLEVEL,THISINDEX]:=T3)%
N:=N+1; INSYMBOL)%
END ELSE ERROR(9))%
END UNTIL CURSY<=COMMA)%
IF CURSY<=RPAR THEN BEGIN ERROR(46); SKIP(RPAR) END)%
T1:=SYMBOLIC; T1,STRUCT:=0)%
T1,SIZE:=TSIZE:=1; TTYPE:=TYPEINDEX)%
TYPETAB1[TYPEINDEX]:=T1)%
TYPETAB2[TYPEINDEX]:=0; TYPETAB3[TYPEINDEX]:=N-1)%
IF CURSY<=RPAR THEN INSYMBOL)%
END ELSE%
%
IF CURSY=ARROW THEN
BEGIN
%*** POINTER DECLARATION ***
%*****
00002646 T 0002
00002647 T 0002
00002648 T 0002
00002649 T 0002
00002650 T 0002
00002651 T 0003
00002652 T 0004
00002653 T 0005
00002654 T 0034
00002655 T 0034
00002656 T 0034
00002657 T 0036
00002658 T 0036
00002659 T 0039
00002660 T 0040
00002661 T 0249
00002662 T 0251
00002663 T 0255
00002664 T 0255
00002665 T 0466
00002666 T 0467
00002667 T 0467
00002668 T 0468
00002669 T 0477
00002670 T 0477
00002671 T 0477
00002672 T 0478
00002673 T 0478
00002674 T 0501
00002675 T 0505
00002676 T 0507
00002677 T 0508
00002678 T 0509
00002679 T 0512
00002680 T 0514
00002681 T 0518
00002682 T 0519
00002683 T 0522
00002684 T 0524
00002685 T 0524
00002686 T 0524
00002687 T 0525

```

```

INSYMBOL;%
IF CURSY=IDENTIFIER THEN%
BEGIN%
  NEWTYPE; TTYPE:=TYPEINDEX; T1:=POINTERS;%
  T1,SIZE:=TSIZE:=1; T1,STRUCT:=0;%
  TYPETAB1[TYPEINDEX]:=T1;%
  SEARCH;%
  IF FOUND THEN%
  BEGIN%
    IF THISID.IDCLASS=TYPES THEN%
    TYPETAB1[TYPEINDEX].POINTTYPE:=THISID.TYPE ELSE%
    TYPERR(7,TTYPE,TSIZE);%
  END ELSE%
  BEGIN%
    IF NumpNtrs<MAXPNTRS THEN NumpNtrs:=NumpNtrs+1 ELSE ERROR(52);
    PNTRTAB1[NumpNtrs]:=CURNAME1; PNTRTAB2[NumpNtrs]:=CURNAME2;%
    PNTRTAB3[NumpNtrs]:=TYPEINDEX;%
  END;%
  INSYMBOL;%
END ELSE TYPERR(9,TTYPE,TSIZE);%
END OF POINTER DECLARATION ELSE%
BEGIN%
  IF CURSY=PACKEDSY THEN BEGIN PACKED:=TRUE; INSYMBOL END;%
  IF CURSY=ARRAYSY THEN
  BEGIN
    INSYMBOL;%
    IF CURSY#LBRACKET THEN ERROR(47) ELSE INSYMBOL;%
    T:=0; FIRST:=TRUE;%
    DO BEGIN%
      IF FIRST THEN FIRST:=FALSE ELSE INSYMBOL;%
      TYPEDECL(TX,SX);%
      IF TX>0 THEN%
      BEGIN%
        IF TYPETAB1[TX].FORM>CHAR THEN ERROR(48);%
        T1:=ARRAYS; T1,INXTYPE:=TX; T1,ARRTYPE:=T;%
        T2:=TYPETAB2[TX]; T3:=TYPETAB3[TX];%
        IF T3-T2>1022 THEN ERROR(61);%
        T1,SIZE:=MIN(1023,T3-T2+1);%
        NEWTYPE;%
        TYPETAB1[TYPEINDEX]:=T1;%
        TYPETAB2[TYPEINDEX]:=T2; TYPETAB3[TYPEINDEX]:=T3;%
        T:=TYPEINDEX;%
      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
  END
  IF CURSY#ARRAYSY THEN
  BEGIN
    INSYMBOL;%
    IF CURSY#LBRACKET THEN ERROR(47) ELSE INSYMBOL;%
    T:=0; FIRST:=TRUE;%
    DO BEGIN%
      IF FIRST THEN FIRST:=FALSE ELSE INSYMBOL;%
      TYPEDECL(TX,SX);%
      IF TX>0 THEN%
      BEGIN%
        IF TYPETAB1[TX].FORM>CHAR THEN ERROR(48);%
        T1:=ARRAYS; T1,INXTYPE:=TX; T1,ARRTYPE:=T;%
        T2:=TYPETAB2[TX]; T3:=TYPETAB3[TX];%
        IF T3-T2>1022 THEN ERROR(61);%
        T1,SIZE:=MIN(1023,T3-T2+1);%
        NEWTYPE;%
        TYPETAB1[TYPEINDEX]:=T1;%
        TYPETAB2[TYPEINDEX]:=T2; TYPETAB3[TYPEINDEX]:=T3;%
        T:=TYPEINDEX;%
      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
  END

```

```

00002688 T 0526
00002689 T 0526
00002690 T 0527
00002691 T 0527
00002692 T 0534
00002693 T 0538
00002694 T 0540
00002695 T 0569
00002696 T 0569
00002697 T 0569
00002698 T 0571
00002699 T 0574
00002700 T 0577
00002701 T 0577
00002702 T 0577
00002703 T 0581
00002704 T 0584
00002705 T 0585
00002706 T 0585
00002707 T 0585
00002708 T 0588
00002709 T 0588
00002710 T 0589
00002711 T 0591
00002712 T 0591
00002713 T 0592
00002714 T 0592
00002715 T 0593
00002716 T 0596
00002717 T 0597
00002718 T 0598
00002719 T 0600
00002720 T 0601
00002721 T 0602
00002722 T 0602
00002723 T 0605
00002724 T 0609
00002725 T 0611
00002726 T 0614
00002727 T 0619
00002728 T 0624
00002729 T 0625
00002730 T 0628
00002731 T 0628
00002732 T 0628
00002733 T 0630
00002734 T 0633
00002735 T 0635
00002736 T 0636
00002737 T 0637
00002738 T 0640
00002739 T 0641
00002740 T 0643
00002741 T 0643
00002742 T 0645
00002743 T 0650
00002744 T 0655

```

```

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 T1=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%
%
IF CURSY=RECORDSY THEN                                  %*** RECORD 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;%

```

```

00002745 T 0658
00002746 T 0660
00002747 T 0662
00002748 T 0662
00002749 T 0662
00002750 T 0663
00002751 T 0663
00002752 T 0664
00002753 T 0665
00002754 T 0666
00002755 T 0668
00002756 T 0669
00002757 T 0670
00002758 T 0670
00002759 T 0672
00002760 T 0674
00002761 T 0677
00002762 T 0677
00002763 T 0683
00002764 T 0687
00002765 T 0691
00002766 T 0693
00002767 T 0694
00002768 T 0694
00002769 T 0694
00002770 T 0695
00002771 T 0696
00002772 T 0696
00002773 T 0697
00002774 T 0698
00002775 T 0700
00002776 T 0701
00002777 T 0702
00002778 T 0703
00002779 T 0703
00002780 T 0706
00002781 T 0710
00002782 T 0710
00002783 T 0716
00002784 T 0720
00002785 T 0724
00002786 T 0725
00002787 T 0727
00002788 T 0727
00002789 T 0727
00002790 T 0728
00002791 T 0729
00002792 T 0733
00002793 T 0734
00002794 T 0736
00002795 T 0736
00002796 T 0738
00002797 T 0740
00002798 T 0746
00002799 T 0751
00002800 T 0754
00002801 T 0757

```

```

        IF CURSY=ENDSY THEN INSYMBOL;%
        END ELSE BEGIN ERROR(4); SKIP(99) END;%
    END;%
END OF TYPEDECL;%

```

```

00002802 T 0760
00002803 T 0762
00002804 T 0764
00002805 T 0764
36 IS 777 LONG, NEXT SEG 2

```

```

%
%
PROCEDURE FIELDLIST(RECTAB,FIRSTADDR, LASTADDR);%
VALUE RECTAB,FIRSTADDR;%
INTEGER RECTAB,FIRSTADDR, LASTADDR;%
BEGIN%
    INTEGER ARRAY ILIST[0:LISTLENGTH];%

```

```

00002806 T 0375
00002807 T 0375
00002808 T 0375
00002809 T 0375
00002810 T 0375
00002811 T 0375
00002812 T 0375
START OF SEGMENT ***** 37

```

```

STACK(F+2) = ILIST
                INTEGER LISTINX;%
STACK(F+3) = LISTINX
                INTEGER CASETYPE, ADDR, MAXADDR, INDEX, CTYPE, TX, SX, T1, T3, LLIM, ULIM, I;%
STACK(F+4) = CASETYPE
STACK(F+5) = ADDR
STACK(F+6) = MAXADDR
STACK(F+7) = INDEX
STACK(F+10) = CTYPE
STACK(F+11) = TX
STACK(F+12) = SX
STACK(F+13) = T1
STACK(F+14) = T3
STACK(F+15) = LLIM
STACK(F+16) = ULIM
STACK(F+17) = I

```

```

00002813 T 0001
00002814 T 0001

```

```

                BOOLEAN FIRST;%
STACK(F+20) = FIRST
                REAL CVAL;%
STACK(F+21) = CVAL
                LABEL CASETYPEID, CASEPART, EXIT;%

```

```

00002815 T 0001
00002816 T 0001

```

```

%
    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);%
                    END ELSE%
                        BEGIN ERROR(9);%
                END ELSE%
                    BEGIN ERROR(9);%
            END ELSE%
                BEGIN ERROR(9);%
        END ELSE%
            BEGIN ERROR(9);%
    END ELSE%
        BEGIN ERROR(9);%

```

```

00002817 T 0001
00002818 T 0001
00002819 T 0001
00002820 T 0002
00002821 T 0003
00002822 T 0005
00002823 T 0006
00002824 T 0007
00002825 T 0007
00002826 T 0009
00002827 T 0010
00002828 T 0012
00002829 T 0013
00002830 T 0013
00002831 T 0016
00002832 T 0017
00002833 T 0040
00002834 T 0041
00002835 T 0042
00002836 T 0042

```



```

                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:=TX;%
        FOR I:=1 STEP 1 UNTIL LISTINX DO%
        BEGIN%
            T3.INFO:=ADDR; ADDR:=MIN(ADDR+SX,1024);%
            NAMETAB3[RECTAB,I,LIST[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];%
                LLIMI:=TYPETAB2[CASETYPE]; ULIMI:=TYPETAB3[CASETYPE];%
                IF T1,FORM>CHAR THEN ERROR(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 LLIMI:=-MAXINT; ULIMI:=MAXINT END;%
            DO BEGIN%
                WHILE CURSY#SEMICOLON DO INSYMBOL;%
                IF CURSY<=CHARCONST OR CURSY=PLUS OR CURSY=MINUS THEN%
                BEGIN%
                    FIRST:=TRUE;%

```

```

00002837 T 0043
00002838 T 0045
00002839 T 0045
00002840 T 0046
00002841 T 0049
00002842 T 0049
00002843 T 0050
00002844 T 0054
00002845 T 0058
00002846 T 0059
00002847 T 0059
00002848 T 0064
00002849 T 0066
00002850 T 0068
00002851 T 0068
00002852 T 0070
00002853 T 0071
00002854 T 0073
00002855 T 0073
00002856 T 0073
00002857 T 0075
00002858 T 0076
00002859 T 0077
00002860 T 0077
00002861 T 0106
00002862 T 0108
00002863 T 0131
00002864 T 0131
00002865 T 0133
00002866 T 0134
00002867 T 0135
00002868 T 0135
00002869 T 0164
00002870 T 0164
00002871 T 0164
00002872 T 0166
00002873 T 0166
00002874 T 0169
00002875 T 0171
00002876 T 0173
00002877 T 0174
00002878 T 0175
00002879 T 0180
00002880 T 0184
00002881 T 0184
00002882 T 0184
00002883 T 0186
00002884 T 0188
00002885 T 0190
00002886 T 0192
00002887 T 0195
00002888 T 0197
00002889 T 0200
00002890 T 0201
00002891 T 0205
00002892 T 0207
00002893 T 0208

```

```

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);%
      IF LISTINX<LISTLENGTH THEN BEGIN ERROR(30); LISTINX:=0 END;%
      LISTINX:=LISTINX+1;%
      ILIST[LISTINX]:=CVAL; I:=1;%
      WHILE ILIST[I]#CVAL DO I:=I+1;%
      IF I<LISTINX THEN ERROR(31);%
    END;%
  END UNTIL CURSY#COMMA;%
  IF CURSY#COLON THEN BEGIN ERROR(26); SKIP(LPAR) END;%
  IF CURSY=COLON THEN INSYMBOL;%
  IF CURSY=LPAR THEN%
    BEGIN%
      INSYMBOL; FIELDLIST(RECTAB,ADDR,MAXADDR);%
      IF MAXADDR>LASTADDR THEN LASTADDR:=MAXADDR;%
      IF CURSY#RPAR THEN BEGIN ERROR(46); SKIP(RPAR) END;%
      INSYMBOL;%
    END ELSE ERROR(58);%
  END;%
END UNTIL CURSY NEQ SEMICOLON; %
EXIT;%
END OF FIELDLIST;%

```

```

00002894 T 0209
00002895 T 0209
00002896 T 0211
00002897 T 0299
00002898 T 0299
00002899 T 0300
00002900 T 0302
00002901 T 0305
00002902 T 0365
00002903 T 0367
00002904 T 0369
00002905 T 0371
00002906 T 0374
00002907 T 0376
00002908 T 0376
00002909 T 0377
00002910 T 0380
00002911 T 0382
00002912 T 0382
00002913 T 0383
00002914 T 0385
00002915 T 0387
00002916 T 0390
00002917 T 0390
00002918 T 0391
00002919 T 0391
00002920 T 0393
00002921 T 0393

```

37 IS 401 LONG, NEXT SEG 2

```

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
%
%
%
%
%
%
%
%
%
%
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
%
%
%

```

PART 8: THE PROCEDURE BLOCK,  
-----

```

00002923 T 0375
00002924 T 0375
00002925 T 0375
00002926 T 0375
00002927 T 0375
00002928 T 0375
00002929 T 0375
00002930 T 0375
00002931 T 0375
00002932 T 0375
00002933 T 0375
00002934 T 0375
00002935 T 0375
00002936 T 0375
00002937 T 0375
00002938 T 0375
00002939 T 0375
00002940 T 0375
00002941 T 0375
00002942 T 0375

```

```

PROCEDURE DECLAREVARS(PARAM,TAB,FIRST, LAST,LEVEL);%
PRT(377) = DECLAREVARS
VALUE PARAM,FIRST, LAST,LEVEL;%
INTEGER ARRAY TAB[0];%
INTEGER FIRST, LAST,LEVEL;%
BOOLEAN PARAM;%
BEGIN%
    INTEGER LEVEL1000,TYP,NAM,NAMTAB,T1,I,J,RECSIZE;%

```

```

STACK(F+2) = LEVEL1000
STACK(F+3) = TYP
STACK(F+4) = NAM
STACK(F+5) = NAMTAB
STACK(F+6) = T1
STACK(F+7) = I
STACK(F+10) = J
STACK(F+11) = RECSIZE
STACK(F+12) = REALVAR
STACK(F+13) = ARRAYVAR
STACK(F+14) = FIRSTDIM
STACK(F+15) = EXTFILE
STACK(F+16) = FNAME
STACK(F+17) = FNLENGTH
STACK(F+20) = FNSTART
%

```

```

    BOOLEAN REALVAR,ARRAYVAR,FIRSTDIM,EXTFILE;%
    ALPHA FNAME;%
    INTEGER FNLENGTH,FNSTART;%

```

```

    LEVEL1000:=LEVEL*1000;%
    FOR I:=FIRST STEP 1 UNTIL LAST DO%
    BEGIN%
        NAMI=TAB[I],[9:10]; NAMTABI=NAMETAB3[LEVEL,NAM];%
        TYPi=NAMTAB,TYPE; T1i=TYPETAB1[TYP];%
        IF NAMTAB,IDCLASS GEQ FUNC THEN%
        BEGIN%
            IF REALVAR OR ARRAYVAR THEN%
            BEGIN%
                GEN(" ",1,7);%
                REALVARI=ARRAYVARI=FALSE;%
            END;%
            IF NAMTAB,IDCLASS=FUNC THEN GEN("REAL",5,4);%
            GEN("PROCEDU",8,1);%
            GENID("V",LEVEL1000+NAM,5); GEN(" ",1,7);%
        END;%
    END;%

```

PRT(400) = \*DESTINATION POINTER\*

START OF SEGMENT \*\*\*\*\* 38

```

00002943 T 0000
00002944 T 0000
00002945 T 0000
00002946 T 0000
00002947 T 0000
00002948 T 0001
00002949 T 0002
00002950 T 0002
00002951 T 0005
00002952 T 0007
00002953 T 0008
00002954 T 0009
00002955 T 0010
00002956 T 0010
00002957 T 0019
00002958 T 0020
00002959 T 0020
00002960 T 0031
00002961 T 0040

```

```

END ELSE%
IF T1,STRUCT=0 THEN                                     %*** SIMPLE TYPE ***
BEGIN%
  IF ARRAYVAR THEN BEGIN GEN(")",1,7); ARRAYVARI=FALSE END;%
  IF REALVAR THEN GEN(")",1,7) ELSE%
    BEGIN GEN("REAL",5,4); REALVARI=TRUE END;%
  GENID("V",LEVEL1000+NAM,5);%
END ELSE%
BEGIN%
  IF REALVAR THEN BEGIN GEN(")",1,7); REALVARI=FALSE END;%
  IF T1,FORM<FILES THEN                                     %*** ARRAY/RECORD ***
  BEGIN%
    IF ARRAYVAR THEN GEN(")",1,7) ELSE%
      BEGIN GEN("ARRAY",6,3); ARRAYVARI=TRUE END;%
    GENID("V",LEVEL1000+NAM,5); GEN("["),1,7);%
    FIRSTDIM:=TRUE;%
    DO BEGIN%
      IF FIRSTDIM THEN FIRSTDIM:=FALSE ELSE GEN(")",1,7);%
      GENINT(TYPETAB2[TYP]);%
      IF NOT PARAM THEN%
        BEGIN GEN(")",1,7); GENINT(TYPETAB3[TYP]) END;%
      TYP:=IF T1,FORM=ARRAYS THEN T1,ARRTYPE ELSE REALTYPE;%
      T1:=TYPETAB1[TYP];%
    END UNTIL T1,STRUCT=0;%
    GEN(")",1,7);%
  END ELSE%
  BEGIN                                                     %*** FILE ***
    IF REALVAR OR ARRAYVAR THEN%
      BEGIN GEN(")",1,7); REALVARI=ARRAYVARI=FALSE END;%
    IF T1,FORM=TEXTFILE AND NOT PARAM THEN%
    BEGIN%
      IF NUMFILES>MAXFILES THEN ERROR(97);%
      ELSE NUMFILES:=NUMFILES+1;%
      FILETAB[ NUMFILES ]:=NAM;%
    END;%
    EXTFILE:=FALSE;%
    FNAME:=NAMETAB1[LEVEL,NAM];%
    FNLENGTH := FNAME,NAMELENGTH; FNSTART := 8-FNLENGTH; %
    IF FNLENGTH LEQ 6 THEN %
    BEGIN%
      FOR J:=1 STEP 1 UNTIL NUMEXTFILES DO%
        IF FNAME=EXTFILETAB[J] THEN EXTFILE:=TRUE;%
    END;%
    IF EXTFILE AND NOT PARAM THEN%
    BEGIN%
      IF NUMFILES GEQ MAXFILES THEN ERROR(97);%
      ELSE%
        NUMFILES := NUMFILES + 1;%
        FILETAB[ NUMFILES ] := "NAM - 1";%
        GEN("DEFINE",7,2); GENID("F",LEVEL1000+NAM,5);%
        GEN(")",1,7);%
        GEN(FNAME,FNLENGTH,FNSTART); %
        GEN("#)",2,6); GEN("SAVE",5,4); GEN("FILE",5,4);%
        GEN(FNAME,FNLENGTH,FNSTART); %
    END ELSE%
    BEGIN%
      GEN("FILE",5,4); GENID("F",LEVEL1000+NAM,5);%

```

```

00002962 T 0062
00002963 T 0062
00002964 T 0066
00002965 T 0066
00002966 T 0077
00002967 T 0087
00002968 T 0097
00002969 T 0109
00002970 T 0109
00002971 T 0112
00002972 T 0122
00002973 T 0123
00002974 T 0124
00002975 T 0134
00002976 T 0144
00002977 T 0165
00002978 T 0166
00002979 T 0167
00002980 T 0179
00002981 T 0223
00002982 T 0223
00002983 T 0277
00002984 T 0281
00002985 T 0282
00002986 T 0283
00002987 T 0292
00002988 T 0292
00002989 T 0293
00002990 T 0294
00002991 T 0304
00002992 T 0306
00002993 T 0307
00002994 T 0309
00002995 T 0311
00002996 T 0312
00002997 T 0312
00002998 T 0313
00002999 T 0314
00003000 T 0317
00003001 T 0318
00003002 T 0318
00003003 T 0320
00003004 T 0324
00003005 T 0324
00003006 T 0325
00003007 T 0326
00003008 T 0327
00003009 T 0328
00003010 T 0329
00003011 T 0331
00003012 T 0353
00003013 T 0362
00003014 T 0371
00003015 T 0398
00003016 T 0407
00003017 T 0407
00003018 T 0411

```

```

END;%
IF NOT PARAM THEN%
BEGIN%
  GEN(" DISK",6,3); GEN("SERIAL",7,2);%
  IF EXTFILE THEN%
  BEGIN%
    IF ALGOLCNT LSS 13 THEN WRITEALGOL;%
    GEN("[010]",5,3);%
    GEN(" ",1,7);%
    GEN(FNAME,FNLENGTH,FNSTART); %
    GEN(" ",1,7); GEN("/ ",1,7);%
    IF ALGOLCNT<9 THEN WRITEALGOL;%
    GEN(" ",1,7); GEN(USER,7,1); GEN(" ",1,7);%
  END ELSE%
  BEGIN%
    GEN("[20]",4,4); GEN("300",4,4);%
  END;%
  GEN("(1",3,5);%
  RECSIZE:=IF T1,FORM=TEXTFILE THEN 10 ELSE%
    IF TYPETAB1[T1,FILETYPE],STRUCT=0 THEN 1 ELSE%
    TYPETAB3[T1,FILETYPE]-TYPETAB2[T1,FILETYPE]+1;%
  GENINT(RECSIZE); GEN(" ",1,7);%
  IF RECSIZE=1 OR RECSIZE=10 THEN GENINT(150)%
    ELSE GENINT(RECSIZE);%
  IF ALGOLCNT LSS 10 THEN WRITEALGOL;%
  GEN(",SAVE",6,3); GEN("30",2,6);%
  GEN(")",2,6);%
  END ELSE GEN(")",1,7);%
  GEN("ARRAY",6,3); GENID("V",LEVEL1000+NAM,5);%
  GEN(" ",1,7);%
  IF TYPETAB1[T1,FILETYPE],STRUCT=0 THEN%
  BEGIN%
    IF PARAM THEN GEN("0",1,7) ELSE GEN("0:0",3,5);%
  END ELSE%
  BEGIN%
    GENINT(TYPETAB2[T1,FILETYPE]);%
    IF NOT PARAM THEN%
    BEGIN GEN("I",1,7); GENINT(TYPETAB3[T1,FILETYPE]) END;%
  END;%
  GEN(")",2,6);%
  GEN("INTEGER",8,1); GENID("I",LEVEL1000+NAM,5);%
  GEN(" ",1,7);%
  END;%
  END;%
  END OF LOOP;%
  IF REALVAR OR ARRAYVAR THEN GEN(")",1,7);%
  END OF DECLAREVARS;%

```

```

00003019 T 0432
00003020 T 0432
00003021 T 0433
00003022 T 0433
00003023 T 0451
00003024 T 0451
00003025 T 0452
00003026 T 0454
00003027 T 0463
00003028 T 0472
00003029 T 0481
00003030 T 0499
00003031 T 0500
00003032 T 0532
00003033 T 0532
00003034 T 0532
00003035 T 0550
00003036 T 0550
00003037 T 0559
00003038 T 0562
00003039 T 0565
00003040 T 0568
00003041 T 0624
00003042 T 0670
00003043 T 0714
00003044 T 0716
00003045 T 0734
00003046 T 0743
00003047 T 0755
00003048 T 0776
00003049 T 0785
00003050 T 0787
00003051 T 0788
00003052 T 0809
00003053 T 0809
00003054 T 0809
00003055 T 0854
00003056 T 0854
00003057 T 0909
00003058 T 0909
00003059 T 0918
00003060 T 0939
00003061 T 0948
00003062 T 0948
00003063 T 0948
00003064 T 0951
00003065 T 0961
38 IS 978 LONG, NEXT SEG 2

```

```

%
%
PROCEDURE PARAMETERLIST;%
PRT(401) = PARAMETERLIST
BEGIN%

```

```

00003066 T 0375
00003067 T 0375
00003068 T 0375
00003069 T 0375

```

INTEGER FIRSTPARAM,CURKIND,P1,PX,I,T;%

00003070 T 0375  
START OF SEGMENT \*\*\*\*\*

39

STACK(F+2) = FIRSTPARAM  
STACK(F+3) = CURKIND  
STACK(F+4) = P1  
STACK(F+5) = PX  
STACK(F+6) = I  
STACK(F+7) = T

STACK(F+10) = FIRST  
%  
BOOLEAN FIRST;%

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],PARAMTYPEI:=T3;%  
IF CURKIND=CONST OR CURKIND=VAR THEN%  
BEGIN%

00003071 T 0000  
00003072 T 0000  
00003073 T 0000  
00003074 T 0000  
00003075 T 0000  
00003076 T 0000  
00003077 T 0000  
00003078 T 0000  
00003079 T 0000  
00003080 T 0000  
00003081 T 0005  
00003082 T 0006  
00003083 T 0006  
00003084 T 0007  
00003085 T 0007  
00003086 T 0010  
00003087 T 0010  
00003088 T 0012  
00003089 T 0015  
00003090 T 0016  
00003091 T 0017  
00003092 T 0019  
00003093 T 0020  
00003094 T 0022  
00003095 T 0023  
00003096 T 0023  
00003097 T 0049  
00003098 T 0051  
00003099 T 0053  
00003100 T 0059  
00003101 T 0060  
00003102 T 0061  
00003103 T 0062  
00003104 T 0063  
00003105 T 0063  
00003106 T 0065  
00003107 T 0066  
00003108 T 0067  
00003109 T 0067  
00003110 T 0096  
00003111 T 0096  
00003112 T 0096  
00003113 T 0098  
00003114 T 0098  
00003115 T 0099  
00003116 T 0101  
00003117 T 0105  
00003118 T 0107

```

T1=TYPETAB1[T3];%
IF T.FORMZFILES THEN%
  FOR I1=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(2); T3:=0 END;%
  INSYMBOL;%
END ELSE%
BEGIN%
  IF CURKIND#PROC THEN ERROR(7);%
  T3:=0;%
END;%
  T3.IDCLASS:=CURKIND; T3.FORMAL:=1;%
  FOR I1=P1 STEP 1 UNTIL NUMPARAMS DO%
    NAMETAB3[CURLEVEL+1,PARAMTAB[I],PARAMNAME]:=T3;%
  END UNTIL CURSY#SEMICOLON;%
  IF CURSY#RPAR THEN%
  BEGIN ERROR(46); SKIP(RPAR);%
  IF CURSY=RPAR THEN INSYMBOL;%
  END ELSE INSYMBOL;%
END;%
PARAMTAB[FIRSTPARAM]:=NUMPARAMS-FIRSTPARAM;%
END OF PARAMETERLIST;%

```

```

00003119 T 0108
00003120 T 0109
00003121 T 0110
00003122 T 0112
00003123 T 0116
00003124 T 0120
00003125 T 0123
00003126 T 0125
00003127 T 0127
00003128 T 0129
00003129 T 0129
00003130 T 0129
00003131 T 0130
00003132 T 0132
00003133 T 0133
00003134 T 0133
00003135 T 0136
00003136 T 0138
00003137 T 0143
00003138 T 0144
00003139 T 0145
00003140 T 0147
00003141 T 0149
00003142 T 0150
00003143 T 0150
00003144 T 0152

```

39 IS 157 LONG, NEXT SEG 2

```

%
%
PROCEDURE BLOCK;%
PRT(402) = BLOCK
  BEGIN%
    INTEGER INDEX,CTYPE,NUMFORWARDS,T,T3,TX,I;%

```

```

STACK(F+2) = INDEX
STACK(F+3) = CTYPE
STACK(F+4) = NUMFORWARDS
STACK(F+5) = T
STACK(F+6) = T3
STACK(F+7) = TX
STACK(F+10) = I

```

```

  REAL CVAL;%
  STACK(F+11) = CVAL
  ALPHA C1,C2;%
  STACK(F+12) = C1
  STACK(F+13) = C2
  BOOLEAN VALUEPARAMS,FUN;%
  STACK(F+14) = VALUEPARAMS
  STACK(F+15) = FUN

```

LABEL START;%

```

%
INTEGER LABTABTOP,CONSTTABTOP,TYPETABTOP,PARAMTABTOP,TOPREC,%

```

```

00003145 T 0375
00003146 T 0375
00003147 T 0375

```

```

00003148 T 0375
00003149 T 0375

```

START OF SEGMENT \*\*\*\*\* 40

```

00003150 T 0000

```

```

00003151 T 0000

```

```

00003152 T 0000

```

```

00003153 T 0000

```

```

00003154 T 0000

```

```

00003155 T 0000

```

```

STACK(F+16) = LABTABTOP
STACK(F+17) = CONSTTABTOP
STACK(F+20) = TYPETABTOP
STACK(F+21) = PARAMTABTOP
STACK(F+22) = TOPREC
FORMERFIRSTLAB,FIRSTFILE;%
STACK(F+23) = FORMERFIRSTLAB
STACK(F+24) = FIRSTFILE
%
FORMERFIRSTLABI=FIRSTLAB;%
LABTABTOPI=NUMLABS; FIRSTLABI=LABTABTOP+1;%
CONSTTABTOPI=NUMCONSTS;%
TYPETABTOPI=NUMTYPES;%
PARAMTABTOPI=NUMPARAMS;%
TOPRECI=LASTREC;%
FIRSTFILEI=NUMFILES+1;%
%
TOPLEVELI=CURLEVEL;%
IF CURLEVEL>1 THEN GEN("BEGIN",6,3);%
START;%
IF CURSY=LABELSY THEN %*** LABEL DECLARATION ***
BEGIN %*****
GEN("LABEL",6,3);%
DO BEGIN%
INSYMBOL;%
IF CURSY=INTCONST THEN%
BEGIN%
GENID("L",CURVAL,4);%
PRT(403) = *DESTINATION POINTER*
IF CURVAL>9999 THEN ERROR(33);%
FOR I=FIRSTLAB STEP 1 UNTIL NUMLABS DO%
IF LABTAB[I],LABVAL=CURVAL THEN ERROR(31);%
IF NUMLABS<=MAXLABS THEN BEGIN ERROR(34); NUMLABSI=0 END;%
NUMLABSI=NUMLABS+1;%
LABTAB[ NUMLABS ]I=CURVAL;%
INSYMBOL;%
END ELSE BEGIN ERROR(10); SKIP(COMMA) END;%
IF CURSY=COMMA THEN GEN(", ",1,7);%
END UNTIL CURSY<=COMMA;%
IF CURSY=SEMICOLON THEN BEGIN ERROR(25); SKIP(SEMICOLON) END;%
GEN("; ",1,7);%
IF SYMKIND[CURSY]≠INITIAL THEN INSYMBOL;%
END OF LABEL DECLARATION;%
%
IF CURSY=CONSTSY THEN %*** CONSTANT DECLARATION ***
BEGIN %*****
INSYMBOL;%
DO BEGIN%
IF CURSY=IDENTIFIER THEN%
BEGIN%
NEWNAME(CURNAME1,CURNAME2,CURLEVEL); INDEXI=THISINDEX;%
INSYMBOL;%
IF CURSY=EQLSY THEN%
BEGIN%
INSYMBOL; CONSTANT(CVAL,CTYPE);%
T3I=CTYPE; T3,1DCLASSI=CONST;%
IF CVAL,[4618]≠0 OR CVAL>1023 THEN%
BEGIN%
00003156 T 0000
00003157 T 0000
00003158 T 0000
00003159 T 0000
00003160 T 0002
00003161 T 0003
00003162 T 0004
00003163 T 0005
00003164 T 0005
00003165 T 0007
00003166 T 0007
00003167 T 0007
00003168 T 0018
00003169 T 0018
00003170 T 0018
00003171 T 0019
00003172 T 0028
00003173 T 0029
00003174 T 0029
00003175 T 0030
00003176 T 0030
00003177 T 0042
00003178 T 0044
00003179 T 0049
00003180 T 0054
00003181 T 0056
00003182 T 0058
00003183 T 0059
00003184 T 0059
00003185 T 0061
00003186 T 0072
00003187 T 0073
00003188 T 0076
00003189 T 0085
00003190 T 0087
00003191 T 0087
00003192 T 0087
00003193 T 0087
00003194 T 0088
00003195 T 0088
00003196 T 0089
00003197 T 0089
00003198 T 0090
00003199 T 0113
00003200 T 0113
00003201 T 0114
00003202 T 0115
00003203 T 0203
00003204 T 0205
00003205 T 0207

```



```

        IF NUMCONSTS>=MAXCONSTS THEN%
        BEGIN ERROR(35); NUMCONSTS:=0 END;%
        NUMCONSTS:=NUMCONSTS+1;%
        CONSTTAB[NUMCONSTS]:=CVAL;%
        T3,INFO:=1023+NUMCONSTS;%
        END ELSE T3,INFO:=CVAL;%
        NAMETAB3[CURLEVEL,INDEX]:=T3;%
        END ELSE BEGIN ERROR(36); SKIP(SEMICOLON) END;%
        END ELSE BEGIN ERROR(2); SKIP(SEMICOLON) END;%
        IF CURSY#SEMICOLON THEN BEGIN ERROR(25); SKIP(SEMICOLON) END;%
        IF SYMKIND[CURSY]#INITIAL THEN INSYMBOL;%
        END UNTIL CURSY#IDENTIFIER;%
    END OF CONSTANT DECLARATION;%

```

%

```

    IF CURSY=TYPE$ THEN
        BEGIN
            INSYMBOL;%
            DO BEGIN%
                IF CURSY=IDENTIFIER THEN%
                    BEGIN%
                        NEWNAME(CURNAME1,CURNAME2,CURLEVEL); INDEX:=THISINDEX;%
                        INSYMBOL;%
                        IF CURSY=EQLSY THEN%
                            BEGIN%
                                INSYMBOL;%
                                TYPEDECL(CTYPE, TX);%
                                T3:=CTYPE; T3.IDCLASS:=TYPE$;%
                                NAMETAB3[CURLEVEL,INDEX]:=T3;%
                            END ELSE BEGIN ERROR(36); SKIP(SEMICOLON) END;%
                            END ELSE BEGIN ERROR(2); SKIP(SEMICOLON) END;%
                            IF CURSY#SEMICOLON THEN BEGIN ERROR(25); SKIP(SEMICOLON) END;%
                            IF SYMKIND[CURSY]#INITIAL THEN INSYMBOL;%
                            END UNTIL CURSY#IDENTIFIER;%
                        END OF TYPE DECLARATION;%
                    END
                END
            END
        END
    END

```

%

```

    IF CURSY=VAR$ THEN
        BEGIN
            VARINDEX:=0;%
            DO BEGIN%
                FIRSTVARI:=VARINDEX+1;%
                DO BEGIN%
                    IF CURSY=VAR$ OR CURSY=COMMA THEN INSYMBOL;%
                    IF CURSY=IDENTIFIER THEN%
                        BEGIN%
                            IF VARINDEX>=LISTLENGTH THEN%
                                BEGIN ERROR(37); VARINDEX:=0 END;%
                                VARINDEX:=VARINDEX+1;%
                                NEWNAME(CURNAME1,CURNAME2,CURLEVEL);%
                                VARLIST[VARINDEX]:=THISINDEX;%
                                INSYMBOL;%
                            END ELSE BEGIN ERROR(9); SKIP(COLON) END;%
                            END UNTIL CURSY#COMMA;%
                            IF CURSY#COLON THEN BEGIN ERROR(26); SKIP(COLON) END;%
                            IF CURSY=COLON THEN%
                                BEGIN%
                                    INSYMBOL;%
                                    TYPEDECL(CTYPE, TX);%
                                END
                            END
                        END
                    END
                END
            END
        END
    END

```

```

00003206 T 0208
00003207 T 0209
00003208 T 0211
00003209 T 0212
00003210 T 0213
00003211 T 0215
00003212 T 0218
00003213 T 0220
00003214 T 0222
00003215 T 0224
00003216 T 0226
00003217 T 0228
00003218 T 0230
00003219 T 0230
00003220 T 0230
00003221 T 0230
00003222 T 0231
00003223 T 0231
00003224 T 0232
00003225 T 0232
00003226 T 0233
00003227 T 0256
00003228 T 0256
00003229 T 0257
00003230 T 0258
00003231 T 0258
00003232 T 0259
00003233 T 0262
00003234 T 0264
00003235 T 0266
00003236 T 0268
00003237 T 0270
00003238 T 0272
00003239 T 0274
00003240 T 0274
00003241 T 0274
00003242 T 0274
00003243 T 0275
00003244 T 0276
00003245 T 0276
00003246 T 0277
00003247 T 0278
00003248 T 0280
00003249 T 0281
00003250 T 0282
00003251 T 0282
00003252 T 0284
00003253 T 0286
00003254 T 0308
00003255 T 0309
00003256 T 0310
00003257 T 0312
00003258 T 0313
00003259 T 0316
00003260 T 0317
00003261 T 0317
00003262 T 0318

```

```

T3:=CTYPE; T3.IDCLASS:=VAR;%
FOR I:=FIRSTVAR STEP 1 UNTIL VARINDEX DO%
  NAMETAB3(CURLEVEL,VARLIST[I])=T3;%
END ELSE BEGIN ERROR(26); SKIP(SEMICOLON) END;%
IF CURSY#SEMICOLON THEN BEGIN ERROR(25); SKIP(SEMICOLON) END;%
IF SYMKIND(CURSY)#INITIAL THEN INSYMBOL;%
END UNTIL CURSY#IDENTIFIER;%
DECLAREVARS(FALSE,VARLIST,1,VARINDEX,CURLEVEL);%
END OF VARIABLE DECLARATION;%
%
IF NumpNtrs>0 THEN%
BEGIN%
  C1:=CURNAME1; C2:=CURNAME2;%
  FOR I:=1 STEP 1 UNTIL NumpNtrs DO%
  BEGIN%
    CURNAME1:=PNTRTAB1[I]; CURNAME2:=PNTRTAB2[I];%
    SEARCHTAB(CURLEVEL);%
    THISID:=NAMETAB3(CURLEVEL,THISINDEX);%
    IF FOUND AND THISID.IDCLASS=TYPES THEN%
      TYPETAB1[PNTRTAB3[I]],POINTTYPE1=THISID,TYPE ELSE ERROR(62);%
    END;%
    CURNAME1=C1; CURNAME2=C2; NumpNtrs:=0;%
  END;%
%
WHILE CURSY=FUNCSY OR CURSY=PROCSY DO %*** PROC/FUNC DECLARATION ***
BEGIN %*****%
  FUN:=CURSY=FUNCSY; INSYMBOL;%
  IF CURSY=IDENTIFIER THEN%
  BEGIN%
    SEARCHTAB(CURLEVEL);%
    THISID:=NAMETAB3(CURLEVEL,THISINDEX);%
    IF FOUND AND THISID.IDCLASS<=PROC THEN%
    BEGIN%
      INDEX:=THISINDEX;%
      IF THISID.FORWARDDEF=1 THEN%
      BEGIN%
        NAMETAB3[THISLEVEL,THISINDEX],FORWARDDEF:=0;%
        NUMFORWARDS:=NUMFORWARDS+1;%
        IF(THISID.IDCLASS=PROC AND FUN)OR%
          (THISID.IDCLASS=FUNC AND NOT FUN) THEN ERROR(43);%
        INSYMBOL;%
      END ELSE BEGIN ERROR(2); SKIP(SEMICOLON) END;%
    END ELSE%
    BEGIN%
      NEWNAME(CURNAME1,CURNAME2,CURLEVEL); INDEX:=THISINDEX;%
      T3:=0; T3.INFO1=NUMPARAMS+1;%
      T3.IDCLASS:=IF FUN THEN FUNC ELSE PROC;%
      NAMETAB3(CURLEVEL,INDEX)=T3;%
      INSYMBOL; PARAMETERLIST;%
      IF CURSY=COLON THEN%
      BEGIN%
        IF NOT FUN THEN ERROR(48);%
        INSYMBOL;%
        IF CURSY=IDENTIFIER THEN%
        BEGIN%
          SEARCH;%
          IF FOUND THEN%
          BEGIN%

```

```

00003263 T 0319
00003264 T 0321
00003265 T 0323
00003266 T 0327
00003267 T 0329
00003268 T 0332
00003269 T 0334
00003270 T 0335
00003271 T 0337
00003272 T 0337
00003273 T 0337
00003274 T 0338
00003275 T 0339
00003276 T 0340
00003277 T 0342
00003278 T 0342
00003279 T 0344
00003280 T 0362
00003281 T 0364
00003282 T 0366
00003283 T 0371
00003284 T 0373
00003285 T 0375
00003286 T 0375
00003287 T 0375
00003288 T 0378
00003289 T 0378
00003290 T 0380
00003291 T 0380
00003292 T 0381
00003293 T 0399
00003294 T 0401
00003295 T 0403
00003296 T 0403
00003297 T 0404
00003298 T 0405
00003299 T 0406
00003300 T 0409
00003301 T 0410
00003302 T 0412
00003303 T 0415
00003304 T 0416
00003305 T 0418
00003306 T 0418
00003307 T 0418
00003308 T 0442
00003309 T 0445
00003310 T 0448
00003311 T 0450
00003312 T 0451
00003313 T 0452
00003314 T 0452
00003315 T 0454
00003316 T 0455
00003317 T 0455
00003318 T 0456
00003319 T 0485
00003320 T 0485

```

```

IF THISID.IDCLASS=TYPES THEN%
BEGIN%
  T:=TYPETAB1[THISID,TYPE];%
  IF T.FORM≤ALFA OR T.FORM=POINTERS THEN%
  BEGIN%
    NAMETAB3[CURLEVEL,INDEX],TYPE:=THISID,TYPE;%
  END ELSE ERROR(38);%
  END ELSE ERROR(7);%
  END ELSE ERROR(8);%
  END ELSE ERROR(9);%
  INSYMBOL;%
  END ELSE IF FUN THEN%
  BEGIN ERROR(26); SKIP(SEMICOLON) END;%
END;%
END ELSE BEGIN ERROR(9); SKIP(SEMICOLON) END;%
IF CURSY#SEMICOLON THEN BEGIN ERROR(25); SKIP(SEMICOLON) END;%
IF FUN THEN GEN("FUNCTN",7,2) ELSE%
GEN("PROCEDU",8,1); GENID("V",1000×CURLEVEL+INDEX,5);%
T:=NAMETAB3[CURLEVEL,INDEX],INFO; TX:=T+PARAMTAB[T];%
IF TX>T THEN%
BEGIN%
  GEN(",",1,7);%
  FOR I:=T+1 STEP 1 UNTIL TX DO%
  BEGIN GENID("V",1000×(CURLEVEL+1)+PARAMTAB[I],PARAMNAME,5);%
  IF BOOLEAN(PARAMTAB[I],PARAMFILE) THEN%
  BEGIN%
    GEN(",",1,7);%
    GENID("F",1000×(CURLEVEL+1)+PARAMTAB[I],PARAMNAME,5);%
    GEN(",",1,7);%
    GENID("I",1000×(CURLEVEL+1)+PARAMTAB[I],PARAMNAME,5);%
  END;%
  IF I LSS TX THEN GEN(",",1,7);%
  END;%
  GEN(")",2,6);%
  VALUEPARAMS:=FALSE;%
  FOR I:=T+1 STEP 1 UNTIL TX DO%
  IF PARAMTAB[I],PARAMKIND=CONST THEN%
  BEGIN%
    IF NOT VALUEPARAMS THEN%
    BEGIN GEN("VALUE",6,3);%
    VALUEPARAMS:=TRUE;%
    END ELSE GEN(",",1,7);%
    GENID("V",1000×(CURLEVEL+1)+PARAMTAB[I],PARAMNAME,5);%
  END;%
  IF VALUEPARAMS THEN GEN(",",1,7);%
  DECLAREVARS(TRUE,PARAMTAB,T+1,TX,CURLEVEL+1);%
  END ELSE GEN(",",1,7);%
  INSYMBOL;%
  IF CURNAME1="7FORWAR" AND CURNAME2="D" THEN%
  BEGIN%
    NAMETAB3[CURLEVEL,INDEX],FORWARDDEFI:=1;%
    NUMFORWARDSI:=NUMFORWARDS+1;%
    GEN("FORWARD",8,1);%
    INSYMBOL;%
  END ELSE%
  BEGIN%

```

```

00003321 T 0485
00003322 T 0487
00003323 T 0487
00003324 T 0489
00003325 T 0491
00003326 T 0492
00003327 T 0496
00003328 T 0497
00003329 T 0498
00003330 T 0499
00003331 T 0501
00003332 T 0501
00003333 T 0502
00003334 T 0504
00003335 T 0504
00003336 T 0506
00003337 T 0509
00003338 T 0518
00003339 T 0543
00003340 T 0546
00003341 T 0547
00003342 T 0548
00003343 T 0557
00003344 T 0561
00003345 T 0575
00003346 T 0576
00003347 T 0577
00003348 T 0586
00003349 T 0600
00003350 T 0609
00003351 T 0625
00003352 T 0625
00003353 T 0635
00003354 T 0635
00003355 T 0644
00003356 T 0645
00003357 T 0649
00003358 T 0651
00003359 T 0651
00003360 T 0652
00003361 T 0661
00003362 T 0662
00003363 T 0674
00003364 T 0688
00003365 T 0688
00003366 T 0698
00003367 T 0701
00003368 T 0711
00003369 T 0711
00003370 T 0711
00003371 T 0713
00003372 T 0714
00003373 T 0717
00003374 T 0718
00003375 T 0727
00003376 T 0728
00003377 T 0728

```

```

CURLEVEL:=CURLEVEL+1;%
IF CURLEVEL>=LASTREC THEN ERROR(55);%
BLOCKTAB[CURLEVEL]:=NUMBLOCKS:=NUMBLOCKS+1;%
T:=CURFUNC; CURFUNC:=IF FUN THEN INDEX ELSE -1;%
BLOCK;  %*** COMPILE PROCEDURE BODY ***
REPLACE POINTER(NAMETAB1[CURLEVEL,*]) BY 0%
      FOR MAXNAMES+1 WORDS;%
CURLEVEL:=CURLEVEL-1; CURFUNC:=T;%
TOPLEVEL:=CURLEVEL;%
END;%
IF CURSY#SEMICOLON THEN BEGIN ERROR(25); SKIP(SEMICOLON) END;%
GEN(";",1,7);%
IF SYMKIND[CURSY]#INITIAL THEN INSYMBOL;%
END OF PROCEDURE DECLARATION;%

%
%
IF NUMFORWARDS>0 THEN ERROR(44);%
GEN("INTEGER",8,1);%
FOR I:=1 STEP 1 UNTIL MAXTEMPS DO%
BEGIN GENID("T",1,2);%
      IF I<MAXTEMPS THEN GEN(";",1,7) ELSE GEN(";",1,7);%
END;%
IF CURSY#BEGINSY THEN%
BEGIN ERROR(39);%
      WHILE SYMKIND[CURSY]#INITIAL DO%
      BEGIN INSYMBOL; SKIP(SEMICOLON) END;%
      IF(CURSY=TYPESY)OR(CONSTSYS<CURSY AND CURSY<PROCSY)THEN%
      GO TO START;%
END;%
IF CURLEVEL=1 THEN%
BEGIN%
      GEN("INIT(",5,3);%
      IF INPUTDECL THEN GEN("TRUE",4,4) ELSE GEN("FALSE",5,3);%
      GEN(")",2,6);%
END;%
FOR I:=FIRSTFILE STEP 1 UNTIL NUMFILES DO%
IF FILETAB[I] LSS 0 THEN%
BEGIN%
      GEN("CHFIL(",6,2);%
      GENID("F",1000*CURLEVEL-FILETAB[I]-1,5);%
      GEN(")",2,6);%
END%
ELSE%
BEGIN%
      GENID("I",1000*CURLEVEL+FILETAB[I],5);%
      GEN(";",1,7); GEN("BUFSIZE",7,1); GEN(":=80",5,3);%
END;%
NUMFILES:=FIRSTFILE-1;%

%
%
COMPSTAT;  %*** COMPILE STATEMENT PART ***

FOR I:=LASTREC STEP 1 UNTIL TOPREC-1 DO%
REPLACE POINTER(NAMETAB1[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;%

```

```

00003378 T 0731
00003379 T 0732
00003380 T 0734
00003381 T 0736
00003382 T 0739
00003383 T 0740
00003384 T 0742
00003385 T 0746
00003386 T 0748
00003387 T 0749
00003388 T 0749
00003389 T 0751
00003390 T 0760
00003391 T 0762
00003392 T 0763
00003393 T 0763
00003394 T 0763
00003395 T 0766
00003396 T 0775
00003397 T 0777
00003398 T 0789
00003399 T 0808
00003400 T 0811
00003401 T 0811
00003402 T 0813
00003403 T 0814
00003404 T 0816
00003405 T 0819
00003406 T 0819
00003407 T 0819
00003408 T 0820
00003409 T 0821
00003410 T 0830
00003411 T 0852
00003412 T 0861
00003413 T 0861
00003414 T 0864
00003415 T 0865
00003416 T 0865
00003417 T 0874
00003418 T 0888
00003419 T 0897
00003420 T 0897
00003421 T 0897
00003422 T 0900
00003423 T 0913
00003424 T 0940
00003425 T 0942
00003426 T 0943
00003427 T 0943
00003428 T 0944
00003429 T 0944
00003430 T 0948
00003431 T 0957
00003432 T 0958
00003433 T 0963
00003434 T 0963

```

```
FIRSTLAB:=FORMERFIRSTLAB;%  
NUMCONSTS:=CONSTITABTOP;%  
NUMTYPES:=TYPETABTOP;%  
NUMPARAMS:=PARAMTABTOP;%  
IF CURLEVEL>1 THEN GEN("END",4,5);%  
END OF BLOCK;%
```

```
00003435 T 0964  
00003436 T 0965  
00003437 T 0966  
00003438 T 0966  
00003439 T 0967  
00003440 T 0977  
40 IS 988 LONG, NEXT SEG 2
```

```

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
%
%
%
%           PART 9:  THE MAIN PROGRAM,
%           -----
%
%
%
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
%
%
INTEGER PROGLNAMELENGTH;%
PRT(404) = PROGLNAMELENGTH
ALPHA PROGLNAME,ALGLNAME;%
PRT(405) = PROGLNAME
PRT(406) = ALGLNAME
%
ALGLNAME:="PASC000"&ENTIER(TIME(4)MOD 10)[17:5:6];%
ALGLNAME:=ALGLNAME&ENTIER(TIME(4)DIV 7)[11:5:6];%
ALGLNAME:=ALGLNAME&ENTIER(TIME(4)MOD 9)[5:5:6];%
USER:=TIME(*1);%
FILL PASCALGOL WITH ALGLNAME,USER;%
BEGIN%
FILE PASCRUN DISK SERIAL "PASCRUN"/"DISK" (2,10,150);%
PRT(407) = *SEGMENT DESCRIPTOR*
PRT(410) = PASCRUN
ARRAY BUF[0:9];%
PRT(411) = BUF
LABEL EOF;%
%
WHILE TRUE DO%
BEGIN%
READ(PASCRUN,9,BUF[*]) [EOF];%
PRT(412) = EOF
WRITE(PASCALGOL,10,BUF[*]);%
END;%
EOF;%
END OF TRANSFER OF RUN TIME SYSTEM;%
PRT(413) = *SEGMENT DESCRIPTOR*
CARDLNGTH:=72;%
INITIALIZE; NEWCARD;%
LISTOPTION:=CHECKOPTION:=TRUE;%
C:=" "; INSYMBOL;%
IF CURSY=PROGRAMSY THEN%
BEGIN%
INSYMBOL;%
IF CURSY=IDENTIFIER THEN%
BEGIN%
PROGLNAME:=CURNAME1,[35:36]; PROGLNAMELENGTH:=MIN(6,CURLNGTH);%
INSYMBOL;%
IF CURSY=LPAR THEN%
BEGIN%
DO BEGIN%
INSYMBOL;%
00003442 T 0375
00003443 T 0375
00003444 T 0375
00003445 T 0375
00003446 T 0375
00003447 T 0375
00003448 T 0375
00003449 T 0375
00003450 T 0375
00003451 T 0375
00003452 T 0375
00003453 T 0375
00003454 T 0375
00003455 T 0375
00003456 T 0375
00003457 T 0375
00003458 T 0378
00003459 T 0382
00003460 T 0386
00003461 T 0387
00003462 T 0391
00003463 T 0391
START OF SEGMENT ***** 41
00003464 T 0004
00003465 T 0006
00003466 T 0006
00003467 T 0006
00003468 T 0007
00003469 T 0007
00003470 T 0013
00003471 T 0017
00003472 T 0017
00003473 T 0018
41 IS 21 LONG, NEXT SEG 2
00003474 T 0394
00003475 T 0394
00003476 T 0460
00003477 T 0462
00003478 T 0463
00003479 T 0464
00003480 T 0464
00003481 T 0465
00003482 T 0465
00003483 T 0466
00003484 T 0470
00003485 T 0471
00003486 T 0471
00003487 T 0472
00003488 T 0473

```

```

IF CURSY=IDENTIFIER THEN%
BEGIN%
  IF CURNAME1="50INPUT" THEN INPUTDECL:=TRUE ELSE%
  IF CURNAME1="60OUTPUT" THEN OUTPUTDECL:=TRUE ELSE%
  BEGIN%
    IF CURLLENGTH>6 THEN ERROR(77);%
    NUMEXTFILES:=NUMEXTFILES+1;%
    IF NUMEXTFILES<=MAXEXTFILES THEN%
    EXTFILETAB[NUMEXTFILES]:=CURNAME1 ELSE%
    IF NUMEXTFILES=MAXEXTFILES+1 THEN ERROR(73);%
  END;%
  END ELSE ERROR(9);%
  INSYMBOL;%
  END UNTIL CURSY#COMMA;%
  IF CURSY#RPAR THEN BEGIN ERROR(46); SKIP(SEMICOLON) END;%
  IF CURSY#RPAR THEN INSYMBOL;%
  IF CURSY#SEMICOLON THEN BEGIN ERROR(25); SKIP(SEMICOLON) END;%
  END ELSE BEGIN ERROR(58); SKIP(SEMICOLON) END;%
  END ELSE BEGIN ERROR(9); SKIP(SEMICOLON) END;%
  END ELSE BEGIN ERROR(75); SKIP(SEMICOLON) END;%
  INSYMBOL;%
  CURLEVEL:=1;%
  LASTREC:=MAXTABLES+1;%
  %%%%%%%%%%%%%%%
  %
  BLOCK;          %  COMPILER USER PROGRAM,          %
  %
  %%%%%%%%%%%%%%%
  IF CURSY#DOT THEN%
  BEGIN%
    ERROR(76);%
    DO BLOCK UNTIL CURSY=DOT;%
  END;%
  IF FALSE THEN%
  BEGIN%
  ENDOFINPUT; ERROR(87);  CHARCNT:=1;%
    WRITE(LINES,TERMMESS);%
  END;%
  IF LISTOPTION AND CHARCNT>0 THEN PRINTLINE;%
  IF ERRINX>0 THEN PRINTERRORS;%
  WRITE(LINES[DBL]);%
  WRITE(LINES[DBL]);%
  IF NUMERRS=0 THEN%
  BEGIN%
    ARRAY ZIPARRAY[0:19], Z[0:0];%

```

```

00003489 T 0473
00003490 T 0474
00003491 T 0474
00003492 T 0476
00003493 T 0481
00003494 T 0483
00003495 T 0485
00003496 T 0486
00003497 T 0487
00003498 T 0488
00003499 T 0491
00003500 T 0491
00003501 T 0493
00003502 T 0493
00003503 T 0494
00003504 T 0497
00003505 T 0499
00003506 T 0502
00003507 T 0504
00003508 T 0506
00003509 T 0508
00003510 T 0508
00003511 T 0509
00003512 T 0510
00003513 T 0510
00003514 T 0510
00003515 T 0511
00003516 T 0511
00003517 T 0511
00003518 T 0511
00003519 T 0512
00003520 T 0513
00003521 T 0514
00003522 T 0514
00003523 T 0515
00003524 T 0515
00003525 T 0517
00003526 T 0520
00003527 T 0520
00003528 T 0565
00003529 T 0567
00003530 T 0571
00003531 T 0575
00003532 T 0576
00003533 T 0576

```

PRT(414) = \*SEGMENT DESCRIPTOR\*

PRT(410) = ZIPARRAY  
PRT(411) = Z

PRT(415) = ZIPPNT  
%

```

  POINTER ZIPPNT;%
  DEFINE ZIPTTEXT(TEXT,L)=%
  BEGIN%
    Z[0]:=TEXT;%
    REPLACE ZIPPNT:ZIPPNT BY POINTER(Z[*])+(8=L) FOR L;%
  END;%

```

START OF SEGMENT \*\*\*\*\* 42

```

00003534 T 0005
00003535 T 0005
00003536 T 0005
00003537 T 0005
00003538 T 0005
00003539 T 0005
00003540 T 0005

```

```

%
PRT(416) = ZIPNUM
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;%

```

```

00003541 T 0005
00003542 T 0005
00003543 T 0005
00003544 T 0005
00003545 T 0014

```

```

%
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(PROGNAME, PROGLNAMELENGTH);%
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%
PRT(417) = *SEGMENT DESCRIPTOR*

```

```

00003546 T 0027
00003547 T 0027
00003548 T 0027
00003549 T 0030
00003550 T 0032
00003551 T 0033
00003552 T 0037
00003553 T 0040
00003554 T 0055
00003555 T 0069
00003556 T 0084
00003557 T 0098
00003558 T 0107
00003559 T 0115
00003560 T 0123
00003561 T 0137
00003562 T 0161
00003563 T 0175
00003564 T 0190
00003565 T 0191
00003566 T 0191
00003567 T 0199
00003568 T 0206
00003569 T 0206
00003570 T 0214
00003571 T 0215

```

```

BEGIN%
INTEGER I;%
PRT(420) = *SEGMENT DESCRIPTOR*
PRT(410) = I
SWITCH FORMAT ERRORMESS1 I=%
PRT(421) = ERRORMESS1

```

```

42 IS 222 LONG, NEXT SEG 2
00003572 T 0577
00003573 T 0577
START OF SEGMENT ***** 43
00003574 T 0000
START OF SEGMENT ***** 44

```

```

(" 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."),%

```

```

00003575 T 0000
00003576 T 0000
00003577 T 0000
00003578 T 0000
00003579 T 0000
00003580 T 0000
00003581 T 0000
00003582 T 0000
00003583 T 0000
00003584 T 0000
00003585 T 0000

```



(" 11	CONSTANT OF OTHER TYPE THAN EXPECTED,")%&	00003586	T	0000
(" 12	VARIABLE OF ILLEGAL TYPE,")%&	00003587	T	0000
(" 13	UNRECOGNIZABLE STATEMENT,")%&	00003588	T	0000
(" 14	CONSTANT TOO BIG OR TOO SMALL,")%&	00003589	T	0000
(" 15	UNDEFINED LABEL,")%&	00003590	T	0000
(" 16	FOR= AND CASE=STATEMENTS NESTED TOO DEEP,")%&	00003591	T	0000
(" 17	EXPRESSION IS OF WRONG TYPE,")%&	00003592	T	0000
(" 18	""OF"" EXPECTED,")%&	00003593	T	0000
(" 19	""DO"" EXPECTED,")%&	00003594	T	0000
(" 20	""ELSE"" WITHOUT CORRESPONDING ""THEN"",")%&	00003595	T	0000
(" 21	ILLEGAL TERMINATION OF STATEMENT,")%&	00003596	T	0000
(" 22	""UNTIL"" EXPECTED,")%&	00003597	T	0000
(" 23	""TO""/""DOWNTO"" EXPECTED,")%&	00003598	T	0000
(" 24	""END"" EXPECTED,")%&	00003599	T	0000
(" 25	"";"" EXPECTED,")%&	00003600	T	0000
(" 26	"";"" EXPECTED,")%&	00003601	T	0000
(" 27	""THEN"" EXPECTED,")%&	00003602	T	0000
(" 28	""I="" EXPECTED,")%&	00003603	T	0000
(" 29	ONLY NUMBERS MAY BE SIGNED,")%&	00003604	T	0000
(" 30	TOO MANY CASES,")%&	00003605	T	0000
(" 31	LABEL USED MORE THAN ONCE,")%&	00003606	T	0000
(" 32	CONSTANT EXPECTED,")%&	00003607	T	0000
(" 33	LABEL NOT IN RANGE 0,,9999,")%&	00003608	T	0000
(" 34	TOO MANY LABELS DECLARED,")%&	00003609	T	0000
(" 35	TOO MANY CONSTANTS DECLARED,")%&	00003610	T	0000
(" 36	""="" EXPECTED,")%&	00003611	T	0000
(" 37	THE LIST IS TOO LONG,")%&	00003612	T	0000
(" 38	INVALID TYPE FOR A FUNCTION,")%&	00003613	T	0000
(" 39	""BEGIN"" EXPECTED,")%&	00003614	T	0000
(" 40	TOO MANY IDENTIFIERS DECLARED,")%&	00003615	T	0000
(" 41	ALFA CONSTANS MAY NOT BE LONGER THAN 7 CHARACTERS,")%&	00003616	T	0000
(" 42	EXPRESSION IS NOT OF TYPE BOOLEAN,")%&	00003617	T	0000
(" 43	NOT PROPER FORWARD DECLARATION,")%&	00003618	T	0000
(" 44	UNSATISFIED FORWARD DECLARATION,")%&	00003619	T	0000
(" 45	TOO MANY DIFFERENT TYPES DECLARED,")%&	00003620	T	0000
(" 46	"""" EXPECTED,")%&	00003621	T	0000
(" 47	""["" EXPECTED,")%&	00003622	T	0000
(" 48	A SIMPLE TYPE EXPECTED,")%&	00003623	T	0000
(" 49	""ARRAY OF ARRAY"" AND ""ARRAY OF RECORD"" ILLEGAL,")%& " AS FILE TYPE,")%&	00003624	T	0000
(" 50	""FILE OF FILE"" IS ILLEGAL,")%&	00003625	T	0000
(" 51	SET BOUNDRY IS TOO BIG OR TOO SMALL,")%&	00003626	T	0000
(" 52	TOO MANY UNDECLARED POINTERS,")%&	00003627	T	0000
(" 53	"",, "" EXPECTED,")%&	00003628	T	0000
(" 54	FIRST VALUE IS GREATER THAN SECOND VALUE,")%&	00003629	T	0000
(" 55	TOO MANY RECORDS DECLARED AT ONE TIME,")%&	00003630	T	0000
(" 56	THE RECORD CONTAINS MORE THAN 1023 WORDS,")%&	00003631	T	0000
(" 57	FILES NOT ALLOWED IN RECORDS,")%&	00003632	T	0000
(" 58	""( "" EXPECTED,")%&	00003633	T	0000
(" 59	""] "" EXPECTED,")%&	00003634	T	0000
		00003635	T	0000
		44 IS	590 LONG,	NEXT SEG 43
			00003636	T 0000
			00003637	T 0000
		START OF SEGMENT	*****	45
			00003638	T 0000
			00003639	T 0000

%

SWITCH FORMAT ERRORMESS2 I=%

PRT(422) = ERRORMESS2

(" 60 ""ARRAY OF FILE"" NOT ALLOWED,")%&  
 (" 61 RANGE OF INDEX IS GREATER THAN 1023,")%&

```

(" 62 UNSATISFIED POINTER DECLARATION,")%
(" 63 EXPRESSION IS TOO LONG,")%
(" 64 ILLEGAL OPERATOR FOR THIS TYPE OF EXPRESSION,")%
(" 65 INTEGER EXPRESSION EXPECTED,")%
(" 66 A SET EXPECTED,")%
(" 67 PARAMETER OF ILLEGAL TYPE,")%
(" 68 PROCEDURES NOT ALLOWED IN THIS CONTEXT,")%
(" 69 ILLEGAL USE OF THIS TYPE OF IDENTIFIER,")%
(" 70 TOO MANY PARAMETERS DECLARED IN THE PROGRAM,")%
(" 71 ""ARRAY OF CHAR"" EXPECTED,")%
(" 72 WRONG TYPE OF SET EXPRESSION,")%
(" 73 TOO MANY EXTERNAL FILES,")%
(" 74 ILLEGAL IDENTIFIER FOR EXTERNAL FILE,")%
(" 75 ""PROGRAM"" EXPECTED,")%
(" 76 """, "" EXPECTED AT END OF PROGRAM,")%
(" 77 EXTERNAL FILE IDENTIFIER MAY NOT EXCEED 6 CHARACTERS,")%
(" 78 ILLEGAL FILE PARAMETER,")%
(" 79 ILLEGAL USE OF FILE HANDLING PROCEDURE,")%
(" 80 TEXT=FILE EXPECTED,")%
(" 81 POINTER VARIABLE EXPECTED,")%
(" 82 ONLY VALUES OF TYPE REAL, INTEGER OR CHAR MAY BE READ,")%
(" 83 VARIABLES IN RECORDS ILLEGAL IN THIS CONTEXT,")%
(" 84 DISPLAY OVERFLOW,")%
(" 85 READ AND WRITE MAY ONLY BE USED ON TEXT=FILES,")%
(" 86 REFERENCED OBJECT IS TOO BIG,")%
(" 87 END=OF=INPUT DISCOVERED,")%
(" 88 CHARACTER ARRAY EXPECTED,")%
(" 89 """, "" EXPECTED,")%
(" 90 PROCEDURES MAY NOT HAVE ANY TYPE,")%
(" 91 PARAMETER OF WRONG KIND,")%
(" 92 ONLY COMPLETE ARRAYS AND RECORDS MAY BE TRANSMITTED,")%
(" 93 DECLARED LABEL NOT USED,")%
(" 94 PARAMETERS OF THIS TYPE SHOULD NOT BE VALUE PARAMETERS,")%
(" 95 ASSIGNMENT OF STRUCTURED VARIABLES NOT IMPLEMENTED,")%
(" 96 INPUT/OUTPUT NOT DECLARED,")%
(" 97 TOO MANY FILES IN USE,")%
(" 98 RECORD IDENTIFIER EXPECTED,")%
(" 99 UNRECOGNIZABLE ITEM,")%
())%

```

```

00003640 T 0000
00003641 T 0000
00003642 T 0000
00003643 T 0000
00003644 T 0000
00003645 T 0000
00003646 T 0000
00003647 T 0000
00003648 T 0000
00003649 T 0000
00003650 T 0000
00003651 T 0000
00003652 T 0000
00003653 T 0000
00003654 T 0000
00003655 T 0000
00003656 T 0000
00003657 T 0000
00003658 T 0000
00003659 T 0000
00003660 T 0000
00003661 T 0000
00003662 T 0000
00003663 T 0000
00003664 T 0000
00003665 T 0000
00003666 T 0000
00003667 T 0000
00003668 T 0000
00003669 T 0000
00003670 T 0000
00003671 T 0000
00003672 T 0000
00003673 T 0000
00003674 T 0000
00003675 T 0000
00003676 T 0000
00003677 T 0000
00003678 T 0000

```

45 IS 430 LONG, NEXT SEG 43

```

%
%
WRITE(LINES, ERRORS, NUMERRS);%
PRT(423) = *LIST, LABEL, OR SEGMENT DESCRIPTOR*
FOR I=0 STEP 1 UNTIL 59 DO IF ERR[I] THEN%
WRITE(LINES, ERRORMESS1[I]);%
FOR I=60 STEP 1 UNTIL 119 DO IF ERR[I] THEN%
WRITE(LINES, ERRORMESS2[I=60]);%
END OF ERROR MESSAGES;%
PRT(424) = *SEGMENT DESCRIPTOR*

```

```

00003679 T 0000
00003680 T 0000
00003681 T 0000
00003682 T 0007
00003683 T 0008
00003684 T 0015
00003685 T 0016
00003686 T 0023

```

43 IS 28 LONG, NEXT SEG 2

```

IF XREFOPTION THEN%
BEGIN%
REPLACE POINTER(XREFLINE[*]) BY " " FOR 17 WORDS;%
HEADING;%
SORT(PRINTXREF, XREFFILE, 0, XREFMAX, XREFCOMPARE, 3, 1000, 6000);%
PRT(425) = 0000369100000000H03+66Q017EDIT10N000000304
PRT(426) = MERGE

```

```

00003687 T 0578
00003688 T 0578
00003689 T 0578
00003690 T 0583
00003691 T 0604

```

PRT(427) = SORT

END;%  
END OF COMPILER,%

00003692 T 0624  
00003693 T 0624  
2 IS 630 LONG, NEXT SEG 1  
START OF SEGMENT \*\*\*\*\* 46  
46 IS 4 LONG, NEXT SEG 1

PRT(300) = EXP INTRINSIC, SEGMENT NUMBER = 47,  
PRT(277) = LN INTRINSIC, SEGMENT NUMBER = 48,  
PRT(242) = OUTPUT(W) INTRINSIC, SEGMENT NUMBER = 49,  
PRT(5) = BLOCK CONTROL INTRINSIC, SEGMENT NUMBER = 50,  
PRT(427) = SORT INTRINSIC, SEGMENT NUMBER = 51,  
PRT(301) = X TO THE I INTRINSIC, SEGMENT NUMBER = 52,  
PRT(273) = GO TO SOLVER INTRINSIC, SEGMENT NUMBER = 53,  
PRT(14) = ALGOL WRITE INTRINSIC, SEGMENT NUMBER = 54,  
PRT(15) = ALGOL READ INTRINSIC, SEGMENT NUMBER = 55,  
PRT(16) = ALGOL SELECT INTRINSIC, SEGMENT NUMBER = 56,  
PRT(426) = MERGE INTRINSIC, SEGMENT NUMBER = 57,  
PRT(34) = FILE ATTRIBS INTRINSIC, SEGMENT NUMBER = 58,  
PRT(240) = DYNAMIC DIALS INTRINSIC, SEGMENT NUMBER = 59,  
PRT(271) = STRNG SCAN INTRINSIC, SEGMENT NUMBER = 60,  
PRT(206) = STRNG REPLACE INTRINSIC, SEGMENT NUMBER = 61,

1 IS 2 LONG, NEXT SEG 0  
62 IS 69 LONG, NEXT SEG 0

FPB

00010200	00000000	00006246	64512325	60042321	51240002	01000000	00000000	43314525	62606005	43314525	62000314
00000000	00000047	21622321	43271147	21622321	43274643	00041400	00000000	00006751	25262631	43106751	25262631
43250005	14472162	23516445	24316242	60606007	47216223	51644500	06140000	00000000	00246251	63016060	05246251
63010007	14000000	00000000	24625163	02606005	24625163	02001002	00000000	00000024	62516303	60600524	62516303
00110200	00000000	00002462	51630460	60052462	51630400	12020000	00000000	00246251	63056060	05246251	63050013
02000000	00000000	24625163	06606005	24625163	06001402	00000000	00000024	62516307	60600524	62516307	00000000
10636411	00000000	10546113	20000000	10457536	04000000	10373465	45000000	10111240	27620000	00013510	35564000
00116432	45121000	00222141	16345200	00326571	42036440	00434327	72446150	00543415	71157602	00654321	27413543
00767405	55316473	01110530	71060221	01212657	07274266	01315432	71153343	01420741	47406234	01525132	01307703
01632360	41571663	01741054	52130240	02051267	64556310	02163545	61711772	02310047	71627437	02412061	70175347

SEGDICT

00000000	00000000	00002200	00201126	00002401	16601100	20017600	05600001	00020500	15100003	00022301	43500015
00000000	07600007	00000000	10100012	00020001	36200051	20027500	01600050	00031300	75200103	00031400	63600124
00031500	76200175	00031601	42000142	00032000	27100216	00032100	32500225	00030200	66700235	00030300	17000254
00032301	40500244	00032501	71700321	00032700	11200316	00033100	36300363	20033200	00700362	00033400	72100374
00033600	53000474	00034000	35200430	00034300	54000440	00034500	13000454	00034700	22200457	00035100	64700464
00035300	21600503	00035700	32300510	00036101	27700550	00036300	04400600	00036500	13500602	00034200	36400606
00037501	41100617	00037400	62100651	00037701	72200667	00040100	23500730	00040201	73400736	00040700	02500777
00041400	33601000	00042000	03401077	20042101	11601034	20042200	65601060	20022400	00401125	10030000	00100020
10027700	00100017	10024200	00100001	10000500	00100002	10042700	00100004	10030100	00100006	10027300	00100021
10001400	00100022	10001500	00100023	10001600	00100024	10042600	00100027	10003400	00100150	10024000	00100040
10027100	00100041	10020600	00100042	20003300	10501127	00000000	00000000	43314525	62606005	43314525	62000314
00000000	00000047	21622321	43271147	21622321	43274643	00041400	00000000	00006751	25262631	43106751	25262631



00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000

SEGZERO

00000000	00001134	00000000	00000077	00000000	00001137	00000000	00000430	00000000	00001132	00000000	00000044
00000000	00000001	00000000	65200014	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000

NUMBER OF ERRORS DETECTED = 0, COMPILATION TIME = 477 SECONDS,

PRT SIZE = 280; TOTAL SEGMENT SIZE = 15836 WORDS; DISK SIZE = 720 SEGS; NO, PGM, SEGS = 62

ESTIMATED CORE STORAGE REQUIRED = 27326 WORDS,

ESTIMATED AUXILIARY MEMORY REQUIRED = 0 WORDS,

NUMBER OF CARD-IMAGES PROCESSED = 3685,

PASCAL /JUNK  
\*\*\*\*\*

SOURCE FILE: ELPASO /U819005

```
A -- ALPHA -- DECLARED IN SEGMENT 5 AT 00000469
*00000537* 00000540 *00000547* 00000553
A -- REAL ARRAY -- NAME PARAMETER -- DECLARED IN SEGMENT 2 AT 00000594
A -- REAL ARRAY -- NAME PARAMETER -- DECLARED IN SEGMENT 2 AT 00000595
*00000597*
A -- REAL ARRAY -- NAME PARAMETER -- DECLARED IN SEGMENT 2 AT 00000601
A -- REAL ARRAY -- NAME PARAMETER -- DECLARED IN SEGMENT 2 AT 00000602
00000604 00000609
A -- REAL ARRAY -- NAME PARAMETER -- DECLARED IN SEGMENT 2 AT 00000613
A -- REAL ARRAY -- NAME PARAMETER -- DECLARED IN SEGMENT 2 AT 00000615
00000625
ABSX -- REAL -- DECLARED IN SEGMENT 4 AT 00000316
*00000329* 00000330 *00000332**00000333* 00000334 00000335
ADDADDR -- LABEL -- DECLARED IN SEGMENT 10 AT 00001163 -- OCCURS AT 00001226
00001176
ADDR -- INTEGER -- DECLARED IN SEGMENT 37 AT 00002814
*00002819**00002847* 00002852 00002856 00002878 *00002879* 00002913
ALFA -- DEFINE -- DECLARED IN SEGMENT 2 AT 00000089
00000513 00001554 00001750 00001847 00001852 00001970 00001975 00001993 00001994 00002090 00002110
00002312 00002660 00002664 00002896 00003202 00003324
ALFACNST -- DEFINE -- DECLARED IN SEGMENT 2 AT 00000194
00001002 00001554 00001946 00001962 00001993 00002312 00002660 00002664 00002896 00003202
ALFANUM -- DEFINE -- DECLARED IN SEGMENT 2 AT 00000858
00000891 00000897 00000901
ALFATYPE -- INTEGER -- DECLARED IN SEGMENT 2 AT 00000161
*00000512* 00001506 00001521 00001525 00001554 00002312 00002660 00002664 00002896 00003202
ALGOLCARD -- REAL ARRAY -- DECLARED IN SEGMENT 2 AT 00000128
00000360 00000361 00000362 00000363 00000485
ALGOLCNT -- INTEGER -- DECLARED IN SEGMENT 2 AT 00000130
00000321 *00000324* 00000327 *00000328**00000338**00000339**00000342**00000343**00000344**00000363**00000485*
*00001807**00001880**00001900**00001901**00001902**00001903**00001904**00001905**00001906**00001907**00001910*
*00001911**00001912**00001918**00001925**00001926**00001927**00001929**00001938**00001948**00001949**00001950*
00001955 *00001957**00001958**00001959**00001960**00001964**00001965**00001966**00001969**00001973**00001974*
*00001975**00001976**00001977**00001982**00001988**00001989**00001992**00001993**00001994**00001995**00001996*
*00001999**00002001**00002010**00002011**00002012**00002014**00002024**00002025**00002026**00002027**00002028*
*00002030**00002034**00002035**00002036**00002037**00002046**00002063**00002065**00002066**00002074**00002078*
*00002088**00002099**00002106**00002113**00002127**00002128**00002129**00002137**00002144**00002151**00002152*
*00002165**00002169**00002218**00002222**00002228**00002246**00002250**00002258**00002266**00002267**00002275*
*00002278**00002293**00002295**00002309**00002317**00002330**00002333**00002340**00002343**00002352**00002353*
*00002361**00002375**00002379**00002390**00002391**00002401**00002429**00002431**00002432**00002443**00002445*
*00002446**00002454**00002455**00002456**00002457**00002459**00002472**00002543**00002582**00002956**00002959*
*00002960**00002961**00002965**00002966**00002967**00002968**00002971**00002974**00002975**00002976**00002979*
*00002980**00002982**00002986**00002990**00003011**00003012**00003013**00003014**00003015**00003018**00003022*
00003025 *00003026**00003027**00003028**00003029* 00003030 *00003031**00003034**00003036**00003040**00003041*
*00003042* 00003043 *00003044**00003045**00003046**00003047**00003048**00003051**00003054**00003056**00003058*
*00003059**00003060**00003064**00003167**00003171**00003176**00003185**00003188**00003337**00003338**00003342*
*00003344**00003347**00003348**00003349**00003350**00003352**00003354**00003360**00003362**00003363**00003365*
```

```

*00003367**00003374**00003389**00003395**00003397**00003398**00003409**00003410**00003411**00003416**00003417*
*00003418**00003422**00003423**00003439*
ALGOLNAME  -- ALPHA  -- DECLARED IN SEGMENT 2 AT 00003455
*00003457**00003458**00003459* 00003461 00003561
ALGOLPNT  -- POINTER -- DECLARED IN SEGMENT 2 AT 00000129
*00000323**00000328**00000336**00000339**00000341**00000343**00000344**00000363* 00000364 *00000485* 00000486
*00001807**00001880**00001900**00001901**00001902**00001903**00001904**00001905**00001906**00001907**00001910*
*00001911**00001912**00001918**00001925**00001926**00001927**00001929**00001938**00001948**00001949**00001950*
*00001956**00001958**00001959**00001960**00001964**00001965**00001966**00001969**00001973**00001974**00001975*
*00001976**00001977**00001982**00001988**00001989**00001992**00001993**00001994**00001995**00001996**00001999*
*00002001**00002010**00002011**00002012**00002014**00002024**00002025**00002026**00002027**00002028**00002030*
*00002034**00002035**00002036**00002037**00002046**00002063**00002065**00002066**00002074**00002078**00002088*
*00002099**00002106**00002113**00002127**00002128**00002129**00002137**00002144**00002151**00002152**00002165*
*00002169**00002218**00002222**00002228**00002246**00002250**00002258**00002266**00002267**00002275**00002278*
*00002293**00002295**00002309**00002317**00002330**00002333**00002340**00002343**00002352**00002353**00002361*
*00002375**00002379**00002390**00002391**00002401**00002429**00002431**00002432**00002443**00002445**00002446*
*00002454**00002455**00002456**00002457**00002459**00002472**00002543**00002582**00002956**00002959**00002960*
*00002961**00002965**00002966**00002967**00002968**00002971**00002974**00002975**00002976**00002979**00002980*
*00002982**00002986**00002990**00003011**00003012**00003013**00003014**00003015**00003018**00003022**00003026*
*00003027**00003028**00003029**00003031**00003034**00003036**00003040**00003041**00003042**00003044**00003045*
*00003046**00003047**00003048**00003051**00003054**00003056**00003058**00003059**00003060**00003064**00003167*
*00003171**00003176**00003185**00003188**00003337**00003338**00003342**00003344**00003347**00003348**00003349*
*00003350**00003352**00003354**00003360**00003362**00003363**00003365**00003367**00003374**00003389**00003395*
*00003397**00003398**00003409**00003410**00003411**00003416**00003417**00003418**00003422**00003423**00003439*
ALIST  -- FORMAT  -- DECLARED IN SEGMENT 3 AT 00000187
00001049
ANDSY  -- DEFINE  -- DECLARED IN SEGMENT 2 AT 00000196
00000476 00000918 00001638 00001649
ARRAYS -- DEFINE  -- DECLARED IN SEGMENT 2 AT 00000094
00001193 00002059 00002122 00002723 00002983
ARRAYSY -- DEFINE  -- DECLARED IN SEGMENT 2 AT 00000206
00000938 00002712
ARRAYVAR -- BOOLEAN -- DECLARED IN SEGMENT 38 AT 00002943
00002954 *00002957**00002965* 00002974 *00002975* 00002989 *00002990* 00003064
ARROW  -- DEFINE  -- DECLARED IN SEGMENT 2 AT 00000201
00000475 00000476 00001182 00001262 00001263 00002686
ARRSTRUCT -- INTEGER -- DECLARED IN SEGMENT 36 AT 00002647
*00002738**00002742*
ARRTYPE -- DEFINE  -- DECLARED IN SEGMENT 2 AT 00000096
00001198 00002062 00002125 00002723 00002741 00002742 00002983
ASSIGN -- LABEL  -- DECLARED IN SEGMENT 26 AT 00002201 -- OCCURS AT 00002208
00002229 00002234
ASSIGNMENT -- PROCEDURE -- DECLARED IN SEGMENT 2 AT 00002198
00002237 00002560 00002584
ASSIGNSY -- DEFINE  -- DECLARED IN SEGMENT 2 AT 00000201
00001027 00002208 00002209 00002233 00002234 00002423 00002425 00002426
ASTERISK -- DEFINE  -- DECLARED IN SEGMENT 2 AT 00000195
00000476 00001629 00001636 00001649
AX  -- INTEGER  -- DECLARED IN SEGMENT 2 AT 00000578
*00000589**00000590* 00000591
A0  -- REAL  -- DECLARED IN SEGMENT 2 AT 00000577
*00000604* 00000606 00000608 *00000625* 00000626 00000645 00000646 00000648 00000649
A1  -- REAL  -- DECLARED IN SEGMENT 2 AT 00000577
*00000604* 00000607 *00000625* 00000626 00000647
A2  -- INTEGER  -- DECLARED IN SEGMENT 2 AT 00000578
*00000625* 00000626 00000635 00000638 *00000640* 00000650 00000651
B  -- REAL ARRAY -- NAME PARAMETER -- DECLARED IN SEGMENT 2 AT 00000601

```

```

B -- REAL ARRAY -- NAME PARAMETER -- DECLARED IN SEGMENT 2 AT 00000602
00000604 00000609
BEGINNUM -- INTEGER -- DECLARED IN SEGMENT 27 AT 00002242
*00002245* 00002258
BEGINSY -- DEFINE -- DECLARED IN SEGMENT 2 AT 00000202
00000935 00002248 00002586 00003400
BITPATTERN -- DEFINE -- DECLARED IN SEGMENT 2 AT 00000212
00001306 00001441 00001564 00001617 00001645 00001666 00001699 00001706 00001726 00001752 00001767
00001794 00001797 00001802 00002267 00002353 00002391
BLANK -- DEFINE -- DECLARED IN SEGMENT 2 AT 00000856
00000882 00000885 00000890 00000893 00000899 00000901 00000965 00000966 00000969 00000973 00000980
00000981 00000983 00000984 00001002 00001005 00001011 00001025 00001027 00001029 00001031 00001033
00001035 00001038 00001041 00001054 00001059 00001066 00001067 00001070
BLOCK -- PROCEDURE -- DECLARED IN SEGMENT 2 AT 00003147
00003382 00003440 00003514 00003520
BLOCKTAB -- REAL ARRAY -- DECLARED IN SEGMENT 2 AT 00000136
00000589 *00002793**00003380*
BOOLEXP -- DEFINE -- DECLARED IN SEGMENT 2 AT 00001812
00002267 00002353 00002391
BOOLTYPE -- INTEGER -- DECLARED IN SEGMENT 2 AT 00000161
*00000516* 00000524 00001435 00001441 00001561 00001643 00001697 00001797 00001971 00002267 00002353
00002391
BRACKETSINWITH -- DEFINE -- DECLARED IN SEGMENT 2 AT 00000077
00001172 00002504
BUF -- REAL ARRAY -- DECLARED IN SEGMENT 41 AT 00003464
00003462 00003470
B0 -- REAL -- DECLARED IN SEGMENT 2 AT 00000577
*00000604* 00000606 00000608
B1 -- REAL -- DECLARED IN SEGMENT 2 AT 00000577
*00000604* 00000607
C -- ALPHA -- DECLARED IN SEGMENT 2 AT 00000861
00000882 *00000885* 00000887 00000889 *00000890* 00000891 00000892 *00000893* 00000897 00000898 *00000899*
*00000901* 00000962 00000964 *00000965**00000966* 00000967 *00000969* 00000970 00000972 *00000973* 00000974
*00000975* 00000978 *00000980**00000981* 00000982 *00000983**00000984* 00001000 *00001002**00001005**00001011*
*00001025* 00001026 *00001027* 00001028 *00001029* 00001030 *00001031* 00001032 *00001033* 00001034 *00001035*
00001036 *00001038* 00001039 *00001041* 00001042 00001043 00001044 00001045 00001046 00001047 00001049
00001053 *00001054**00001059* 00001060 00001063 *00001066**00001067* 00001068 *00001070**00003477*
C -- INTEGER -- DECLARED IN SEGMENT 29 AT 00002287
*00002325* 00002327
CALLGEN -- BOOLEAN -- DECLARED IN SEGMENT 16 AT 00001735
*00001762**00001775* 00001796
CARD -- FILE -- DECLARED IN SEGMENT 2 AT 00000033
00000884 00001065 00003475
CARDCNT -- INTEGER -- DECLARED IN SEGMENT 2 AT 00000032
00000360 00000407 00000589 *00000884**00001065* 00001196 00001312 00001416 00001467 00001478 00001485
00001492 00001607 00001862 00001907 00001912 00001960 00001999 00002037 00002099 00002144 00002220
00002430 00002444 *00003475* 00003527
CARDLENGTH -- INTEGER -- DECLARED IN SEGMENT 2 AT 00000158
00000398 00000884 00000956 00001052 *00001053**00001054* 00001055 *00001056* 00001057 00001065 *00003474*
00003475
CARDPNT -- POINTER -- DECLARED IN SEGMENT 2 AT 00000129
*00000883**00000884**00000885**00000890**00000893**00000899**00000901**00000957* 00000958 *00000965**00000966*
*00000969**00000973**00000980**00000981**00000983**00000984**00001002**00001005**00001011**00001025**00001027*
*00001029**00001031**00001033**00001035**00001038**00001041**00001054**00001059**00001064**00001065**00001066*
*00001067**00001070**00003475*
CASEHASH -- DEFINE -- DECLARED IN SEGMENT 29 AT 00002285
00002324
CASENUM -- INTEGER -- DECLARED IN SEGMENT 29 AT 00002287

```



```

*00002290* 00002343
CASEPART -- LABEL -- DECLARED IN SEGMENT 37 AT 00002817 -- OCCURS AT 00002855
00002822
CASESTAT -- PROCEDURE -- DECLARED IN SEGMENT 2 AT 00002283
00002348 00002588
CASESY -- DEFINE -- DECLARED IN SEGMENT 2 AT 00000203
00000386 00000931 00002588 00002822
CASETAB -- INTEGER ARRAY -- DECLARED IN SEGMENT 29 AT 00002286
00002325 *00002327*
CASETYPE -- INTEGER -- DECLARED IN SEGMENT 29 AT 00002287
*00002295* 00002296 *00002297* 00002299 00002302 *00002315* 00002316
CASETYPE -- INTEGER -- DECLARED IN SEGMENT 37 AT 00002814
*00002873* 00002874 00002878 00002888 *00002899* 00002901
CASETYPEID -- LABEL -- DECLARED IN SEGMENT 37 AT 00002817 -- OCCURS AT 00002873
00002861
CH -- REAL ARRAY -- DECLARED IN SEGMENT 2 AT 00000126
00000487 00000885 00000890 00000893 00000899 00000901 00000965 00000966 00000969 00000973 00000980
00000981 00000983 00000984 00001002 00001005 00001011 00001016 00001025 00001027 00001029 00001031
00001033 00001035 00001038 00001041 00001054 00001059 00001066 00001067 00001070 *00001902**00001903*
*00001904**00001925**00001926**00001927**00001948**00001949**00001950**00001964**00001965**00001966**00002010*
*00002011**00002012**00002034**00002035**00002036**00002063**00002127**00002228**00002293**00002317**00002429*
*00002443**00002454**00002455**00002457**00002472**00002543**00002961**00002968**00002976**00003011**00003018*
*00003047**00003059**00003176**00003338**00003344**00003348**00003350**00003363**00003397**00003417**00003422*
CHANGFLINE -- BOOLEAN -- VALUE PARAMETER -- DECLARED IN SEGMENT 2 AT 00001875
CHANGELINE -- BOOLEAN -- VALUE PARAMETER -- DECLARED IN SEGMENT 2 AT 00001876
00001923
CHAR -- DEFINE -- DECLARED IN SEGMENT 2 AT 00000087
00000521 00001197 00001301 00001355 00001449 00001464 00001596 00001597 00001604 00001605 00001658
00001721 00001790 00001793 00001897 00001970 00001976 00001995 00002062 00002075 00002125 00002138
00002219 00002223 00002316 00002415 00002433 00002447 00002660 00002664 00002722 00002763 00002779
00002875 00002901
CHARCNT -- INTEGER -- DECLARED IN SEGMENT 2 AT 00000130
00000396 *00000883**00000884**00000885**00000890**00000893**00000899**00000901* 00000956 00000957 *00000965*
*00000966**00000969**00000973**00000980**00000981**00000983**00000984**00001002**00001005* 00001006 *00001011*
*00001025**00001027**00001029**00001031**00001033**00001035**00001038**00001041* 00001052 *00001054**00001057*
*00001059**00001064**00001065**00001066**00001067**00001070**00003475**00003524* 00003527
CHARCONST -- DEFINE -- DECLARED IN SEGMENT 2 AT 00000195
00001015 00001552 00001554 00002312 00002660 00002664 00002774 00002891 00002896 00003202
CHARPNT -- POINTER -- DECLARED IN SEGMENT 2 AT 00000127
*00000487* 00000885 00000890 00000893 00000899 00000901 00000965 00000966 00000969 00000973 00000980
00000981 00000983 00000984 00001002 00001005 00001009 00001011 00001016 00001025 00001027 00001029
00001031 00001033 00001035 00001038 00001041 00001054 00001059 00001066 00001067 00001070 00001902
00001903 00001904 00001925 00001926 00001927 00001948 00001949 00001950 00001964 00001965 00001966
00002010 00002011 00002012 00002034 00002035 00002036 00002063 00002127 00002228 00002293 00002317
00002429 00002443 00002454 00002455 00002457 00002472 00002543 00002961 00002968 00002976 00003011
00003018 00003047 00003059 00003176 00003338 00003344 00003348 00003350 00003363 00003397 00003417
00003422
CHARTYPE -- INTEGER -- DECLARED IN SEGMENT 2 AT 00000161
*00000520* 00001244 00001421 00001554 00002312 00002660 00002664 00002896 00003202
CHECK -- BOOLEAN -- DECLARED IN SEGMENT 11 AT 00001282
*00001301* 00001302 00001308
CHECK -- BOOLEAN -- DECLARED IN SEGMENT 18 AT 00001879
*00001899* 00001901 00001908
CHECKEXPR -- DEFINE -- DECLARED IN SEGMENT 2 AT 00001144
00001198 00001416 00002220 00002430 00002444
CHECKOPTION -- BOOLEAN -- DECLARED IN SEGMENT 2 AT 00000157
*00001045* 00001196 00001301 00001899 00002219 00002430 00002444 *00003476*

```

```

CHECKTYPES  -- DEFINE  -- DECLARED IN SEGMENT 2 AT 00000660
00001197 00001355 00001597 00001605 00001658 00001721 00001790 00001793 00002075 00002138 00002223
00002316 00002433 00002447 00002901
COLON  -- DEFINE  -- DECLARED IN SEGMENT 2 AT 00000201
00000475 00001026 00001978 00001983 00002334 00002335 00002545 00002547 00002840 00002864 00002909
00002910 00003103 00003256 00003258 00003259 00003312
COMMA  -- DEFINE  -- DECLARED IN SEGMENT 2 AT 00000200
00000477 00001207 00001357 00001358 00001608 00001609 00001848 00001853 00001856 00001859 00001887
00001919 00001945 00002002 00002072 00002076 00002111 00002135 00002173 00002330 00002332 00002519
00002677 00002732 00002837 00002839 00002908 00003102 00003184 00003185 00003186 00003247 00003257
00003502
COMPSTAT  -- PROCEDURE  -- DECLARED IN SEGMENT 2 AT 00002240
00002260 00002586 00003427
CONCAT  -- PROCEDURE  -- DECLARED IN SEGMENT 2 AT 00001833  -- FORWARD AT 00001089
00001497 00001872
CONST  -- DEFINE  -- DECLARED IN SEGMENT 2 AT 00000062
00000524 00000532 00001299 00001323 00001393 00001395 00001554 00001894 00002086 00002205 00002312
00002410 00002559 00002660 00002664 00002668 00002896 00003090 00003117 00003123 00003202 00003203
00003357
CONSTANT  -- DEFINE  -- DECLARED IN SEGMENT 2 AT 00000731
00001554 00001619 00002312 00002660 00002664 00002896 00003202 00003218
CONSTSY  -- DEFINE  -- DECLARED IN SEGMENT 2 AT 00000207
00000939 00003192 00003404
CONSTTAB  -- REAL ARRAY  -- DECLARED IN SEGMENT 2 AT 00000116
*00000534* 00001398 00001554 00002312 00002660 00002664 00002896 00003202 *00003209*
CONSTTABTOP  -- INTEGER  -- DECLARED IN SEGMENT 40 AT 00003155
*00003160* 00003436
CONTYPE  -- INTEGER  -- DECLARED IN SEGMENT 29 AT 00002287
*00002312* 00002313 00002315 00002316
CONVAL  -- INTEGER  -- DECLARED IN SEGMENT 29 AT 00002287
*00002312* 00002317 00002321 00002324 00002325 00002327
CTYPE  -- INTEGER  -- DECLARED IN SEGMENT 37 AT 00002814
*00002896* 00002897 00002899 00002901
CTYPE  -- INTEGER  -- DECLARED IN SEGMENT 40 AT 00003149
*00003202* 00003203 00003231 00003232 00003262 00003263
CURFUNC  -- INTEGER  -- DECLARED IN SEGMENT 2 AT 00000177
00002227 *00003381* *00003385*
CURKIND  -- INTEGER  -- DECLARED IN SEGMENT 39 AT 00003070
*00003087* *00003090* 00003097 00003105 00003117 00003123 00003131 00003134
CURLLENGTH  -- INTEGER  -- DECLARED IN SEGMENT 2 AT 00000862
*00000889* 00000891 *00000893* 00000895 00000897 *00000899* 00000903 00000904 00000956 00000958 00000959
*00001002* 00001009 *00001010* 00001013 00001014 00001018 00001020 00001554 00001946 00001953 00001959
00001999 00002312 00002660 00002664 00002896 00003202 00003483 00003494
CURLEVEL  -- INTEGER  -- DECLARED IN SEGMENT 2 AT 00000171
00001168 00001178 00001319 00001345 00001389 00001427 00001554 00001881 00001890 00001939 00002030
00002058 00002064 00002082 00002116 00002126 00002159 00002227 00002312 00002405 00002412 00002413
00002487 00002555 00002653 00002660 00002664 00002673 00002674 00002694 00002791 00002860 00002868
00002896 00003096 00003098 00003109 00003136 00003166 00003167 00003198 00003202 00003212 00003226
00003233 00003253 00003265 00003270 00003279 00003280 00003292 00003293 00003307 00003310 00003318
00003326 00003338 00003339 00003344 00003348 00003350 00003363 00003366 00003372 *00003378* 00003379
00003380 00003383 *00003385* 00003386 00003407 00003417 00003422 00003439 *00003510*
CURMODE  -- INTEGER  -- DECLARED IN SEGMENT 2 AT 00000180
*00001275* 00001306 *00001364* *00001386* *00001399* *00001441* 00001562 *00001564* *00001585* *00001617* 00001628
00001646 00001666 *00001668* 00001684 00001700 00001707 00001726 *00001728* 00001752 00001767 00001794
*00001797* 00001802 00002267 00002353 00002391
CURNAME?  -- ALPHA  -- DECLARED IN SEGMENT 2 AT 00000861
*00000889* *00000892* *00000903* 00000907 00000908 00000909 00000910 00000911 00000912 00000913 00000914

```

00000915 00000916 00000917 00000918 00000919 00000920 00000921 00000922 00000923 00000924 00000925  
00000926 00000927 00000928 00000929 00000930 00000931 00000932 00000933 00000934 00000935 00000936  
00000937 00000938 00000939 00000940 00000941 00000942 00000943 00000944 00000945 00000947 00000949  
00001222 00001319 00001345 00001389 00001407 00001412 00001423 00001424 00001426 00001427 00001437  
00001443 00001455 00001456 00001458 00001473 00001481 00001488 00001496 00001498 00001503 00001508  
00001513 00001518 00001522 00001533 00001554 00001881 00001890 00001939 00002030 00002053 00002082  
00002116 00002151 00002159 00002312 00002405 00002487 00002555 00002565 00002566 00002568 00002569  
00002570 00002571 00002572 00002573 00002574 00002576 00002577 00002579 00002580 00002653 00002660  
00002664 00002673 00002694 00002703 00002832 00002860 00002862 00002868 00002896 00003096 00003109  
00003198 00003202 00003226 00003253 00003275 \*00003278\* 00003279 \*00003284\* 00003292 00003307 00003318  
00003370 00003483 00003491 00003492 00003497

CURNAME2 == ALPHA == DECLARED IN SEGMENT 2 AT 00000861

\*00000889\*00000898\* 00000945 00000947 00000949 00001222 00001319 00001345 00001389 00001427 00001508  
00001518 00001554 00001881 00001890 00001939 00002030 00002053 00002082 00002116 00002159 00002312  
00002405 00002487 00002555 00002567 00002575 00002578 00002653 00002660 00002664 00002673 00002694  
00002703 00002832 00002860 00002862 00002868 00002896 00003096 00003109 00003198 00003202 00003226  
00003253 00003275 \*00003278\* 00003279 \*00003284\* 00003292 00003307 00003318 00003370

CURSY == INTEGER == DECLARED IN SEGMENT 2 AT 00000178

00000382 00000383 00000386 \*00000906\*00000907\*\*00000913\*\*00000928\*\*00000935\*\*00000941\*\*00000945\*\*00000947\*  
\*00000949\*\*00000951\*\*00000952\*\*00000953\* 00000955 \*00000964\*\*00000971\*\*00000980\* 00000989 \*00001002\*\*00001015\*  
\*00001025\* 00001026 \*00001027\* 00001028 \*00001029\* 00001030 \*00001031\* 00001032 \*00001033\* 00001034 \*00001035\*  
00001036 00001182 00001186 00001207 00001208 00001210 00001213 00001220 00001263 00001291 00001317  
00001343 00001357 00001358 00001359 00001361 00001387 00001409 00001415 00001418 00001419 00001427  
00001432 00001433 00001439 00001446 00001450 00001460 00001469 00001470 00001475 00001483 00001490  
00001534 00001552 00001554 00001556 00001567 00001572 00001576 00001580 00001583 00001598 00001608  
00001609 00001610 00001612 00001629 00001631 00001636 00001637 00001638 00001641 00001649 00001678  
00001680 00001689 00001694 00001695 00001702 00001709 00001710 00001746 00001749 00001754 00001755  
00001756 00001757 00001758 00001759 00001769 00001770 00001773 00001774 00001779 00001780 00001843  
00001848 00001853 00001856 00001859 00001864 00001871 00001881 00001883 00001885 00001887 00001888  
00001919 00001920 00001921 00001939 00001941 00001943 00001945 00001946 00001967 00001978 00001983  
00002002 00002003 00002004 00002030 00002038 00002039 00002048 00002051 00002072 00002076 00002080  
00002096 00002097 00002107 00002111 00002114 00002135 00002141 00002142 00002154 00002157 00002173  
00002175 00002176 00002179 00002180 00002208 00002210 00002234 00002248 00002251 00002252 00002253  
00002254 00002256 00002268 00002271 00002273 00002277 00002298 00002301 00002305 00002307 00002312  
00002330 00002332 00002334 00002335 00002337 00002340 00002341 00002342 00002354 00002357 00002359  
00002380 00002381 00002382 00002383 00002386 00002388 00002403 00002423 00002426 00002427 00002435  
00002436 00002439 00002441 00002448 00002451 00002468 00002485 00002519 00002520 00002522 00002534  
00002545 00002547 00002553 00002586 00002587 00002588 00002589 00002590 00002591 00002592 00002593  
00002594 00002651 00002660 00002664 00002665 00002671 00002677 00002678 00002683 00002686 00002689  
00002710 00002712 00002715 00002732 00002733 00002734 00002749 00002752 00002754 00002769 00002772  
00002774 00002789 00002801 00002802 00002821 00002822 00002823 00002828 00002837 00002839 00002840  
00002851 00002858 00002864 00002866 00002886 00002887 00002890 00002891 00002896 00002908 00002909  
00002910 00002911 00002915 00002919 00003081 00003085 00003087 00003088 00003094 00003102 00003103  
00003107 00003137 00003138 00003140 00003169 00003174 00003185 00003186 00003187 00003189 00003192  
00003196 00003200 00003202 00003215 00003216 00003217 00003220 00003224 00003228 00003236 00003237  
00003238 00003241 00003247 00003248 00003257 00003258 00003259 00003267 00003268 00003269 00003287  
00003289 00003290 00003312 00003316 00003336 00003388 00003390 00003400 00003402 00003404 00003478  
00003481 00003485 00003489 00003502 00003503 00003504 00003505 00003517 00003520

CURTYPE == INTEGER == DECLARED IN SEGMENT 2 AT 00000179

\*00001175\*\*00001179\* 00001191 00001192 00001197 \*00001198\* 00001204 00001205 00001206 00001217 \*00001227\*  
\*00001228\*\*00001229\* 00001234 \*00001237\* 00001238 \*00001244\*\*00001259\*\*00001260\* 00001264 \*00001327\* 00001335  
\*00001338\*\*00001339\*\*00001340\*\*00001350\*\*00001352\*\*00001353\* 00001355 \*00001399\*\*00001409\* 00001410 00001417  
\*00001421\*\*00001427\*\*00001435\*\*00001439\* 00001440 \*00001441\* 00001449 \*00001453\* 00001464 00001465 00001466  
\*00001471\*\*00001475\* 00001476 \*00001479\*\*00001483\* 00001486 \*00001490\* 00001493 \*00001494\*\*00001501\*\*00001506\*  
\*00001511\*\*00001516\*\*00001521\*\*00001525\*\*00001534\* 00001535 \*00001536\*\*00001542\*\*00001548\*\*00001549\*\*00001550\*  
\*00001554\* 00001560 \*00001561\*\*00001569\*\*00001585\* 00001595 00001596 00001597 00001603 00001604 00001605  
\*00001616\* 00001631 00001643 00001653 00001655 00001657 00001658 \*00001659\* 00001661 \*00001663\*\*00001664\*

```

00001686 00001691 00001697 00001716 00001718 00001720 00001721 *00001722*00001724* 00001748 00001785
00001788 00001789 00001790 00001792 00001793 *00001797* 00001846 00001847 00001851 00001852 00001855
00001858 00001861 *00001870**00001881* 00001882 00001896 00001910 00001911 *00001939* 00001940 00001968
00001972 00001981 00001987 *00002030* 00002032 00002075 00002089 00002090 00002110 00002138 00002163
00002166 00002207 00002223 00002267 00002269 00002272 00002295 00002353 00002355 00002358 00002391
00002438 00002437 00002440 00002447 00002449 00002452 00002493 00002494 00002500 00002501
CURVAL -- REAL -- DECLARED IN SEGMENT 2 AT 00000860
*00000964**00000966**00000972**00000992* 00000995 *00000997**00001016**00001021* 00001554 00002312 00002470
00002472 00002537 00002543 00002660 00002664 00002896 00003176 00003177 00003179 00003182 00003202
CVAL -- REAL -- DECLARED IN SEGMENT 37 AT 00002816
*00002896* 00002900 00002904 00002905
CVAL -- REAL -- DECLARED IN SEGMENT 40 AT 00003150
*00003202* 00003204 00003209 00003211
CX -- ALPHA -- DECLARED IN SEGMENT 2 AT 00000861
*00001041* 00001042 00001044 00001045 00001046 00001047 00001048 00001051
C1 -- ALPHA -- DECLARED IN SEGMENT 40 AT 00003151
*00003275* 00003284
C2 -- ALPHA -- DECLARED IN SEGMENT 40 AT 00003151
*00003275* 00003284
D -- REAL -- DECLARED IN SEGMENT 34 AT 00002481
*00002500**00002501**00002502**00002503**00002504**00002511* 00002512
DECL -- BOOLEAN -- VALUE PARAMETER -- DECLARED IN SEGMENT 2 AT 00000228
DECL -- BOOLEAN -- VALUE PARAMETER -- DECLARED IN SEGMENT 2 AT 00000232
00000229
DECL -- BOOLEAN -- VALUE PARAMETER -- DECLARED IN SEGMENT 2 AT 00000580
DECL -- BOOLEAN -- VALUE PARAMETER -- DECLARED IN SEGMENT 2 AT 00000584
00000581 00000590
DECLAREVARS -- PROCEDURE -- DECLARED IN SEGMENT 2 AT 00002936
00003063 00003270 00003366
DISPLAY -- REAL ARRAY -- DECLARED IN SEGMENT 2 AT 00000072
00001168 00001319 00001345 00001389 00001427 00001554 00001881 00001890 00001939 00002030 00002053
00002082 00002116 00002159 00002312 00002405 00002487 *00002512* 00002555 00002653 00002660 00002664
00002694 00002860 00002868 00002896 00003109 00003202 00003318
DIVSY -- DEFINE -- DECLARED IN SEGMENT 2 AT 00000196
00000915 00001641 00001661
DOLLAR -- DEFINE -- DECLARED IN SEGMENT 2 AT 00000856
00001039
DOSY -- DEFINE -- DECLARED IN SEGMENT 2 AT 00000204
00000908 00002354 00002356 00002357 00002448 00002450 00002451 00002520 00002521 00002522
DOT -- DEFINE -- DECLARED IN SEGMENT 2 AT 00000201
00000478 00001028 00001182 00001213 00001232 00001263 00003517 00003520
DOUBLFDOT -- DEFINE -- DECLARED IN SEGMENT 2 AT 00000200
00000478 00001029 00001598 00002660 00002664
DOWN -- BOOLEAN -- DECLARED IN SEGMENT 32 AT 00002398
*00002436* 00002456
DOWNTOSY -- DEFINE -- DECLARED IN SEGMENT 2 AT 00000205
00000942 00002436
DUMMY -- INTEGER -- DECLARED IN SEGMENT 13 AT 00001406
DUMMY -- INTEGER -- DECLARED IN SEGMENT 20 AT 00002009
DUMPOPTION -- BOOLEAN -- DECLARED IN SEGMENT 2 AT 00000157
00000362 *00001046*
EDITION -- DEFINE -- DECLARED IN SEGMENT 2 AT 00000026
00000492
ELSESY -- DEFINE -- DECLARED IN SEGMENT 2 AT 00000203
00000929 00002251 00002277 00002380
EMPTYSET -- INTEGER -- DECLARED IN SEGMENT 2 AT 00000162
*00000530* 00001197 00001355 00001585 00001597 00001605 00001658 00001721 00001790 00001793 00002075

```

```

00002138 00002223 00002316 00002433 00002447 00002901
ENDOFINPUT -- LABEL -- DECLARED IN SEGMENT 2 AT 00000183 -- OCCURS AT 00003524
00000884 00001065 00003475
ENDSY -- DEFINE -- DECLARED IN SEGMENT 2 AT 00000202
00000913 00002254 00002255 00002256 00002307 00002337 00002342 00002801 00002802
EOF -- LABEL -- DECLARED IN SEGMENT 41 AT 00003465 -- OCCURS AT 00003472
00003469
EQLSY -- DEFINE -- DECLARED IN SEGMENT 2 AT 00000198
00000477 00000927 00001758 00001769 00001770 00001779 00003200 00003228
EQUAL -- DEFINE -- DECLARED IN SEGMENT 2 AT 00000856
00001026 00001030 00001034
ERR -- BOOLEAN ARRAY -- DECLARED IN SEGMENT 2 AT 00000154
*00000395* 00003682 00003684
ERRINX -- INTEGER -- DECLARED IN SEGMENT 2 AT 00000132
*00000396* 00000397 00000398 *00000400* *00000411* 00000884 00001065 00003475 00003528
ERRLINE -- REAL ARRAY -- DECLARED IN SEGMENT 2 AT 00000131
00000398 00000408 00000410 00000484
ERRNUM -- INTEGER -- VALUE PARAMETER -- DECLARED IN SEGMENT 2 AT 00000391
ERRNUM -- INTEGER -- VALUE PARAMETER -- DECLARED IN SEGMENT 2 AT 00000392
00000395 00000399 00000400 00000401
ERRNUM -- INTEGER -- VALUE PARAMETER -- DECLARED IN SEGMENT 36 AT 00002640
ERRNUM -- INTEGER -- VALUE PARAMETER -- DECLARED IN SEGMENT 36 AT 00002642
00002641 00002643
ERROR -- PROCEDURE -- DECLARED IN SEGMENT 2 AT 00000391
00000402 00000506 00000510 00000514 00000518 00000522 00000525 00000527 00000531 00000540 00000542
00000543 00000544 00000553 00000555 00000556 00000561 00000564 00000566 00000976 00000986 00000997
00001006 00001013 00001056 00001170 00001178 00001180 00001189 00001193 00001194 00001196 00001197
00001202 00001202 00001203 00001205 00001206 00001209 00001216 00001218 00001226 00001227 00001228
00001229 00001238 00001243 00001248 00001258 00001260 00001266 00001268 00001269 00001285 00001293
00001296 00001302 00001303 00001307 00001310 00001311 00001312 00001328 00001329 00001330 00001336
00001338 00001339 00001340 00001348 00001349 00001352 00001353 00001355 00001357 00001359 00001360
00001369 00001397 00001398 00001409 00001410 00001416 00001417 00001418 00001420 00001427 00001428
00001429 00001432 00001439 00001440 00001445 00001449 00001450 00001452 00001453 00001459 00001463
00001464 00001465 00001466 00001467 00001468 00001469 00001471 00001475 00001476 00001477 00001478
00001483 00001484 00001485 00001486 00001490 00001491 00001492 00001493 00001500 00001505 00001510
00001515 00001520 00001524 00001533 00001534 00001535 00001547 00001549 00001550 00001554 00001558
00001561 00001563 00001569 00001574 00001576 00001577 00001585 00001592 00001596 00001597 00001600
00001604 00001605 00001607 00001608 00001611 00001614 00001619 00001626 00001636 00001637 00001638
00001640 00001641 00001647 00001648 00001649 00001650 00001651 00001658 00001661 00001667 00001677
00001680 00001687 00001694 00001695 00001701 00001702 00001708 00001709 00001710 00001711 00001712
00001714 00001721 00001727 00001740 00001743 00001744 00001753 00001754 00001755 00001756 00001757
00001758 00001759 00001761 00001762 00001768 00001770 00001774 00001775 00001779 00001780 00001781
00001783 00001790 00001792 00001793 00001795 00001796 00001805 00001841 00001847 00001850 00001852
00001855 00001858 00001861 00001862 00001863 00001864 00001865 00001866 00001867 00001868 00001869
00001881 00001882 00001914 00001915 00001916 00001917 00001920 00001939 00001940 00001981 00001985
00001987 00001998 00002003 00002030 00002031 00002033 00002038 00002062 00002064 00002067 00002068
00002069 00002070 00002075 00002090 00002091 00002092 00002093 00002094 00002095 00002096 00002098
00002110 00002125 00002126 00002130 00002131 00002132 00002133 00002138 00002139 00002140 00002141
00002143 00002168 00002170 00002171 00002172 00002175 00002178 00002179 00002209 00002215 00002220
00002223 00002227 00002251 00002252 00002255 00002267 00002269 00002272 00002292 00002297 00002299
00002302 00002312 00002316 00002322 00002327 00002329 00002334 00002338 00002342 00002353 00002355
00002358 00002380 00002381 00002385 00002391 00002412 00002413 00002418 00002419 00002420 00002421
00002424 00002430 00002433 00002434 00002437 00002440 00002444 00002447 00002449 00002452 00002471
00002474 00002494 00002496 00002505 00002513 00002514 00002515 00002516 00002521 00002538 00002542
00002546 00002580 00002583 00002584 00002595 00002643 00002660 00002664 00002668 00002673 00002676
00002678 00002691 00002702 00002715 00002722 00002725 00002727 00002733 00002734 00002737 00002753
00002752 00002760 00002762 00002773 00002779 00002780 00002782 00002791 00002796 00002797 00002801

```

00002803	00002830	00002832	00002836	00002840	00002843	00002862	00002864	00002875	00002882	00002883
00002884	00002885	00002886	00002896	00002900	00002901	00002902	00002906	00002909	00002915	00002917
00002993	00003007	00003080	00003096	00003099	00003100	00003105	00003123	00003124	00003125	00003126
00003127	00003131	00003139	00003177	00003179	00003180	00003184	00003187	00003198	00003202	00003207
00003213	00003214	00003215	00003226	00003234	00003235	00003236	00003251	00003253	00003256	00003258
00003266	00003267	00003282	00003302	00003304	00003307	00003314	00003327	00003328	00003329	00003330
00003338	00003335	00003336	00003379	00003388	00003394	00003401	00003432	00003494	00003498	00003500
00003503	00003505	00003506	00003507	00003508	00003519	00003524	00003686			

ERRORMESS1 -- SWITCH FORMAT -- DECLARED IN SEGMENT 44 AT 00003574  
 00003683  
 ERRORMESS2 -- SWITCH FORMAT -- DECLARED IN SEGMENT 45 AT 00003637  
 00003685  
 ERRORS -- FORMAT -- DECLARED IN SEGMENT 3 AT 00000186  
 00003681  
 EXIT -- LABEL -- DECLARED IN SEGMENT 11 AT 00001283 -- OCCURS AT 00001360  
 00001296  
 EXIT -- LABEL -- DECLARED IN SEGMENT 26 AT 00002201 -- OCCURS AT 00002236  
 00002210  
 EXIT -- LABEL -- DECLARED IN SEGMENT 28 AT 00002265 -- OCCURS AT 00002279  
 00002273  
 EXIT -- LABEL -- DECLARED IN SEGMENT 30 AT 00002351 -- OCCURS AT 00002364  
 00002359  
 EXIT -- LABEL -- DECLARED IN SEGMENT 37 AT 00002817 -- OCCURS AT 00002920  
 00002853  
 EXP -- INTEGER -- DECLARED IN SEGMENT 8 AT 00000877  
 \*00000983\*\*00000984\*\*00000985\* 00000987  
 EXPRESSION -- PROCEDURE -- DECLARED IN SEGMENT 2 AT 00001732 -- FORWARD AT 00001088  
 00001196 00001305 00001409 00001416 00001439 00001448 00001462 00001475 00001483 00001490 00001534  
 00001575 00001593 00001601 00001809 00001845 00001850 00001855 00001858 00001861 00001968 00001980  
 00001986 00002000 00002075 00002109 00002138 00002220 00002221 00002267 00002294 00002353 00002391  
 00002430 00002444  
 EXPRLEVEL -- INTEGER -- DECLARED IN SEGMENT 2 AT 00001093  
 \*0000130\*\*00001305\*\*00001737\* 00001738 \*00001799\* 00001800 \*00002267\*\*00002353\*\*00002391\*  
 EXTERNALFILE -- DEFINE -- DECLARED IN SEGMENT 2 AT 00000069  
 00000563  
 EXTFILE -- BOOLEAN -- DECLARED IN SEGMENT 38 AT 00002943  
 \*00002997\*\*00003003\* 00003005 00003023  
 EXTFILETAB -- REAL ARRAY -- DECLARED IN SEGMENT 2 AT 00000150  
 00003003 \*00003497\*  
 F -- INTEGER -- DECLARED IN SEGMENT 14 AT 00001625  
 \*00001632\* 00001633 00001640 00001643 00001649  
 F -- INTEGER -- DECLARED IN SEGMENT 15 AT 00001674  
 \*00001686\* 00001687 \*00001691\* 00001692 00001704  
 F -- INTEGER -- DECLARED IN SEGMENT 16 AT 00001734  
 \*00001748\* 00001750 00001761 00001765 00001777  
 F -- INTEGER -- DECLARED IN SEGMENT 18 AT 00001878  
 \*00001896\* 00001897 00001899 00001905 00001906  
 F -- INTEGER -- DECLARED IN SEGMENT 19 AT 00001936  
 \*00001968\* 00001970 00001973 00001974 00001975 00001976 00001985 00001992 00001993 00001994 00001995  
 F -- INTEGER -- DECLARED IN SEGMENT 21 AT 00002021  
 \*00002032\* 00002033  
 FACTOR -- PROCEDURE -- DECLARED IN SEGMENT 2 AT 00001368  
 00001550 00001620 00001627 00001652  
 FIELDLIST -- PROCEDURE -- DECLARED IN SEGMENT 2 AT 00002808 -- FORWARD AT 00002614  
 00002795 00002913 00002921  
 FILEHANDLING -- PROCEDURE -- DECLARED IN SEGMENT 2 AT 00002018  
 00002040 00002570 00002571 00002572 00002573 00002575  
 FILEID -- INTEGER -- DECLARED IN SEGMENT 18 AT 00001878

\*00001881\* 00001902 00001903 00001904 00001925 00001926 00001927  
FILEID -- INTEGER -- DECLARED IN SEGMENT 19 AT 00001936  
\*00001939\* 00001948 00001949 00001950 00001964 00001965 00001966 00002006  
FILENAME -- INTEGER -- DECLARED IN SEGMENT 2 AT 00000697  
\*00001427\* 00001428 \*00001881\*\*00001939\*\*00002006\* 00002010 00002011 00002012 \*00002030\* 00002031 00002034  
00002035 00002036  
FILEPARAM -- DEFINE -- DECLARED IN SEGMENT 2 AT 00000700  
00001427 00001881 00001939 00002030  
FILES -- DEFINE -- DECLARED IN SEGMENT 2 AT 00000099  
00001235 00001427 00001881 00001882 00001939 00001940 00002030 00002033 00002737 00002759 00002763  
00002843 00002972 00003120  
FILESY -- DEFINE -- DECLARED IN SEGMENT 2 AT 00000206  
00000934 00002749  
FILETAB -- REAL ARRAY -- DECLARED IN SEGMENT 2 AT 00000152  
\*00002995\*\*00003010\* 00003414 00003417 00003422  
FILETYPE -- DEFINE -- DECLARED IN SEGMENT 2 AT 00000100  
00001237 00002764 00003038 00003039 00003049 00003054 00003056  
FINIS -- BOOLEAN -- VALUE PARAMETER -- DECLARED IN SEGMENT 2 AT 00000613  
FINIS -- BOOLEAN -- VALUE PARAMETER -- DECLARED IN SEGMENT 2 AT 00000614  
00000617  
FINIS -- BOOLEAN -- DECLARED IN SEGMENT 2 AT 00000863  
\*00001003\*\*00001005\*\*00001006\* 00001007 00001012 \*00001061\*\*00001068\* 00001069  
FIRST -- BOOLEAN -- DECLARED IN SEGMENT 12 AT 00001371  
\*00001426\* 00001429 \*00001458\* 00001463 \*00001589\*\*00001591\*  
FIRST -- BOOLEAN -- DECLARED IN SEGMENT 29 AT 00002288  
\*00002306\*\*00002311\*  
FIRST -- BOOLEAN -- DECLARED IN SEGMENT 36 AT 00002648  
\*00002716\*\*00002718\*  
FIRST -- BOOLEAN -- DECLARED IN SEGMENT 37 AT 00002815  
\*00002825\*\*00002827\*\*00002893\*\*00002895\*  
FIRST -- INTEGER -- VALUE PARAMETER -- DECLARED IN SEGMENT 2 AT 00002936  
FIRST -- INTEGER -- VALUE PARAMETER -- DECLARED IN SEGMENT 2 AT 00002939  
00002937 00002948  
FIRST -- BOOLEAN -- DECLARED IN SEGMENT 39 AT 00003071  
\*00003091\*\*00003093\*  
FIRSTADDR -- INTEGER -- VALUE PARAMETER -- DECLARED IN SEGMENT 2 AT 00002614  
FIRSTADDR -- INTEGER -- VALUE PARAMETER -- DECLARED IN SEGMENT 2 AT 00002616  
00002615  
FIRSTADDR -- INTEGER -- VALUE PARAMETER -- DECLARED IN SEGMENT 2 AT 00002808  
FIRSTADDR -- INTEGER -- VALUE PARAMETER -- DECLARED IN SEGMENT 2 AT 00002810  
00002809 00002819  
FIRSTDIM -- BOOLEAN -- DECLARED IN SEGMENT 38 AT 00002943  
\*00002977\*\*00002979\*  
FIRSTFILE -- INTEGER -- DECLARED IN SEGMENT 40 AT 00003156  
\*00003164\* 00003413 00003425  
FIRSTLAB -- INTEGER -- DECLARED IN SEGMENT 2 AT 00000123  
00002536 00003158 \*00003159\* 00003178 00003431 \*00003435\*  
FIRSTPARAM -- INTEGER -- DECLARED IN SEGMENT 39 AT 00003070  
\*00003080\* 00003143  
FIRSTSYM -- INTEGER -- DECLARED IN SEGMENT 11 AT 00001281  
\*00001303\* 00001307  
FIRSTSYM -- INTEGER -- DECLARED IN SEGMENT 16 AT 00001734  
\*00001741\* 00001804  
FIRSTVAR -- INTEGER -- DECLARED IN SEGMENT 2 AT 00000141  
\*00003245\* 00003264  
FIRSTWITHSYM -- DEFINE -- DECLARED IN SEGMENT 2 AT 00000074  
00001169 00002503

```

FLOATING  -- DEFINE  -- DECLARED IN SEGMENT 2 AT 00000088
00000509 00001554 00001633 00001640 00001687 00001692 00001761 00001897 00001899 00001906 00001970
00001974 00001985 00001992 00002296 00002312 00002660 00002664 00002896 00003202
FNAME  -- ALPHA  -- DECLARED IN SEGMENT 38 AT 00002944
*00002998* 00002999 00003003 00003013 00003015 00003028
FNLENGTH  -- INTEGER  -- DECLARED IN SEGMENT 38 AT 00002945
*00002999* 00003000 00003013 00003015 00003028
FNSTART  -- INTEGER  -- DECLARED IN SEGMENT 38 AT 00002945
*00002999* 00003013 00003015 00003028
FORM  -- DEFINE  -- DECLARED IN SEGMENT 2 AT 00000082
00000509 00000513 00000517 00000521 00001193 00001197 00001218 00001235 00001240 00001246 00001301
00001355 00001409 00001417 00001427 00001439 00001449 00001464 00001475 00001483 00001490 00001534
00001554 00001596 00001597 00001604 00001605 00001632 00001657 00001658 00001686 00001691 00001720
00001728 00001748 00001789 00001790 00001792 00001793 00001847 00001852 00001855 00001858 00001861
00001881 00001882 00001896 00001939 00001940 00001968 00001981 00001987 00002030 00002032 00002059
00002062 00002075 00002090 00002110 00002122 00002125 00002138 00002163 00002219 00002223 00002296
00002312 00002316 00002415 00002433 00002447 00002494 00002660 00002664 00002722 00002737 00002759
00002763 00002779 00002843 00002875 00002896 00002901 00002972 00002983 00002991 00003037 00003120
00003202 00003324
FORMAL  -- DEFINE  -- DECLARED IN SEGMENT 2 AT 00000067
00001287 00001323 00001393 00001554 00001894 00002086 00002205 00002312 00002410 00002559 00002660
00002664 00002896 00003134 00003202
FORMALPROC  -- BOOLEAN  -- DECLARED IN SEGMENT 11 AT 00001282
*00001287* 00001289 00001363
FORMERFIRSTLAB  -- INTEGER  -- DECLARED IN SEGMENT 40 AT 00003156
*00003158* 00003435
FORSTAT  -- PROCEDURE  -- DECLARED IN SEGMENT 2 AT 00002395
00002461 00002591
FORSY  -- DEFINE  -- DECLARED IN SEGMENT 2 AT 00000204
00000914 00002591
FORWARDDEF  -- DEFINE  -- DECLARED IN SEGMENT 2 AT 00000068
00003297 00003299 00003372
FOUND  -- BOOLEAN  -- DECLARED IN SEGMENT 2 AT 00000419
*00001222* 00001223 *00001319* 00001320 *00001345* 00001346 *00001389* 00001390 *00001427**00001554**00001881*
*00001898* 00001891 *00001939**00002030**00002053* 00002054 *00002082* 00002083 *00002116* 00002117 *00002159*
00002160 00002202 *00002312**00002405* 00002406 *00002487* 00002488 *00002555* 00002556 *00002653* 00002654
*00002660**00002664**00002694* 00002695 *00002860* 00002861 *00002868* 00002869 *00002896**00003109* 00003110
*00003202**00003279* 00003281 *00003292* 00003294 *00003318* 00003319
FUN  -- BOOLEAN  -- DECLARED IN SEGMENT 40 AT 00003152
*00003289* 00003301 00003302 00003309 00003314 00003332 00003337 00003381
FUNC  -- DEFINE  -- DECLARED IN SEGMENT 2 AT 00000063
00000548 00001350 00001402 00002560 00002952 00002959 00003088 00003302 00003309
FUNCSY  -- DEFINE  -- DECLARED IN SEGMENT 2 AT 00000208
00000947 00003085 00003088 00003287 00003289
F1  -- INTEGER  -- DECLARED IN SEGMENT 2 AT 00000658
*00001197**00001355**00001597**00001605**00001658**00001721**00001790**00001793**00002075**00002138**00002223*
*00002316**00002433**00002447**00002901*
F2  -- INTEGER  -- DECLARED IN SEGMENT 2 AT 00000658
*00001197**00001355**00001597**00001605**00001658**00001721**00001790**00001793**00002075**00002138**00002223*
*00002316**00002433**00002447**00002901*
GEN  -- DEFINE  -- DECLARED IN SEGMENT 2 AT 00000277
00000328 00000339 00000343 00000344 00001807 00001880 00001900 00001901 00001902 00001903 00001904
00001905 00001906 00001907 00001910 00001911 00001912 00001918 00001925 00001926 00001927 00001929
00001938 00001948 00001949 00001950 00001958 00001959 00001960 00001964 00001965 00001966 00001969
00001973 00001974 00001975 00001976 00001977 00001982 00001988 00001989 00001992 00001993 00001994
00001995 00001996 00001999 00002001 00002010 00002011 00002012 00002014 00002024 00002025 00002026
00002027 00002028 00002030 00002034 00002035 00002036 00002037 00002046 00002065 00002066 00002074
00002078 00002088 00002099 00002106 00002113 00002128 00002129 00002137 00002144 00002151 00002152

```



00002165	00002169	00002218	00002222	00002246	00002250	00002258	00002266	00002267	00002275	00002278
00002295	00002295	00002309	00002317	00002330	00002333	00002340	00002343	00002352	00002353	00002361
00002375	00002379	00002390	00002391	00002401	00002429	00002431	00002432	00002443	00002445	00002446
00002454	00002455	00002456	00002457	00002459	00002472	00002543	00002582	00002956	00002959	00002960
00002961	00002965	00002966	00002967	00002971	00002974	00002975	00002976	00002979	00002980	00002982
00002986	00002990	00003011	00003012	00003013	00003014	00003015	00003018	00003022	00003026	00003027
00003028	00003029	00003031	00003034	00003036	00003040	00003041	00003042	00003044	00003045	00003046
00003047	00003048	00003051	00003054	00003056	00003058	00003059	00003060	00003064	00003167	00003171
00003185	00003188	00003337	00003338	00003342	00003347	00003349	00003352	00003354	00003360	00003362
00003365	00003367	00003374	00003389	00003395	00003398	00003409	00003410	00003411	00003416	00003418
00003423	00003439									
GENID	-- DEFINE	-- DECLARED IN SEGMENT 2 AT	00000286							
00001902	00001903	00001904	00001925	00001926	00001927	00001948	00001949	00001950	00001964	00001965
00001966	00002010	00002011	00002012	00002034	00002035	00002036	00002063	00002127	00002228	00002293
00002317	00002429	00002443	00002454	00002455	00002457	00002472	00002543	00002961	00002968	00002976
00003011	00003018	00003047	00003059	00003176	00003338	00003344	00003348	00003350	00003363	00003397
00003417	00003422									
GENINT	-- DEFINE	-- DECLARED IN SEGMENT 2 AT	00000297							
00001807	00001900	00001905	00001906	00001907	00001910	00001911	00001912	00001959	00001960	00001973
00001974	00001975	00001976	00001992	00001993	00001994	00001995	00001999	00002037	00002065	00002066
00002088	00002099	00002128	00002129	00002144	00002165	00002169	00002218	00002222	00002267	00002317
00002352	00002391	00002431	00002445	00002582	00002980	00002982	00003040	00003041	00003042	00003054
00003056										
GENREAL	-- PROCEDURE	-- DECLARED IN SEGMENT 2 AT	00000313							
00000348	00001807	00001900	00002088	00002165	00002218	00002222	00002267	00002353	00002391	00002431
00002445	00002582									
GEQSY	-- DEFINE	-- DECLARED IN SEGMENT 2 AT	00000198							
00000475	00000924	00001035	00001756	00001774						
GOTOSTAT	-- PROCEDURE	-- DECLARED IN SEGMENT 2 AT	00002464							
00002475	00002593									
GOTOSY	-- DEFINE	-- DECLARED IN SEGMENT 2 AT	00000205							
00000932	00002593									
GTRSY	-- DEFINE	-- DECLARED IN SEGMENT 2 AT	00000198							
00000475	00000925	00001034	00001757							
HASH	-- DEFINE	-- DECLARED IN SEGMENT 2 AT	00000415							
00000506	00000510	00000514	00000518	00000522	00000525	00000527	00000531	00000540	00000542	00000543
00000544	00000553	00000555	00000556	00000561	00000564	00000566	00001222	00001319	00001345	00001389
00001427	00001554	00001881	00001890	00001939	00002030	00002053	00002082	00002116	00002159	00002312
00002405	00002487	00002555	00002653	00002660	00002664	00002673	00002694	00002832	00002860	00002862
00002868	00002896	00003096	00003109	00003198	00003202	00003226	00003253	00003279	00003292	00003307
00003318										
HEADING	-- DEFINE	-- DECLARED IN SEGMENT 2 AT	00000239							
00000407	00000499	00000631	00000642	00000884	00001042	00001065	00003475	00003527	00003690	
HEADTEXT	-- REAL ARRAY	-- DECLARED IN SEGMENT 2 AT	00000131							
00000407	00000491	00000492	00000494	00000497	00000499	00000631	00000642	00000884	00001042	00001065
00003475	00003527	00003690								
I	-- INTEGER	-- DECLARED IN SEGMENT 19 AT	00001936							
*00001952*	00001953									
I	-- INTEGER	-- DECLARED IN SEGMENT 33 AT	00002466							
*00002469*	00002470*	00002471								
I	-- INTEGER	-- DECLARED IN SEGMENT 34 AT	00002480							
*00002506*	00002508									
I	-- INTEGER	-- DECLARED IN SEGMENT 35 AT	00002531							
*00002536*	00002537	00002538	00002539							
I	-- INTEGER	-- DECLARED IN SEGMENT 37 AT	00002814							
*00002845*	00002848	*00002904*	*00002905*	00002906						
I	-- INTEGER	-- DECLARED IN SEGMENT 38 AT	00002942							

```

*00002948* 00002950
I -- INTEGER -- DECLARED IN SEGMENT 39 AT 00003070
*00003115* 00003116 *00003121* 00003122 *00003135* 00003136
I -- INTEGER -- DECLARED IN SEGMENT 40 AT 00003149
*00003178* 00003179 *00003264* 00003265 *00003276* 00003278 00003282 *00003343* 00003344 00003345 00003348
00003350 00003352 *00003356* 00003357 00003363 *00003396* 00003397 00003398 *00003413* 00003414 00003417
00003422 *00003429* 00003430 *00003431* 00003432
I -- INTEGER -- DECLARED IN SEGMENT 43 AT 00003573
*00003682* 00003683 *00003684* 00003685
ICARD -- REAL ARRAY -- DECLARED IN SEGMENT 2 AT 00000128
00000884 00001065 00003475
IDCLASS -- DEFINE -- DECLARED IN SEGMENT 2 AT 00000060
00000505 00000510 00000514 00000518 00000522 00000524 00000532 00000536 00000546 00000560 00000562
00001322 00001323 00001348 00001350 00001392 00001393 00001395 00001402 00001427 00001545 00001554
00001881 00001893 00001894 00001939 00002030 00002056 00002085 00002086 00002119 00002204 00002205
00002312 00002409 00002410 00002490 00002558 00002559 00002560 00002561 00002656 00002660 00002664
00002668 00002697 00002844 00002861 00002871 00002878 00002896 00002952 00002959 00003112 00003134
00003202 00003203 00003232 00003263 00003281 00003294 00003301 00003302 00003309 00003321
IDENTIFIER -- DEFINE -- DECLARED IN SEGMENT 2 AT 00000194
00000906 00000912 00000927 00000934 00000940 00000944 00000946 00000948 00000950 00000951 00000952
00000953 00000955 00001220 00001317 00001343 00001387 00001427 00001551 00001554 00001881 00001888
00001939 00002030 00002051 00002080 00002114 00002157 00002175 00002312 00002403 00002485 00002553
00002585 00002651 00002660 00002664 00002671 00002689 00002754 00002823 00002828 00002858 00002866
00002898 00003094 00003107 00003196 00003202 00003217 00003224 00003238 00003248 00003269 00003290
00003316 00003481 00003489
IFSTAT -- PROCEDURE -- DECLARED IN SEGMENT 2 AT 00002263
00002280 00002587
IFSY -- DEFINE -- DECLARED IN SEGMENT 2 AT 00000202
00000907 00002587
ILIST -- INTEGER ARRAY -- DECLARED IN SEGMENT 37 AT 00002812
*00002833* 00002848 *00002904* 00002905
INBRACKET -- BOOLEAN -- DECLARED IN SEGMENT 10 AT 00001162
*00001172* 00001188 *00001189* 00001215 *00001216* 00001248 *00001249* 00001266 00001274
INDEX -- INTEGER -- DECLARED IN SEGMENT 37 AT 00002814
*00002856* *00002862* 00002876 00002879
INDEX -- INTEGER -- DECLARED IN SEGMENT 40 AT 00003149
*00003198* 00003212 *00003226* 00003233 *00003296* *00003307* 00003310 00003326 00003338 00003339 00003372
00003381
INFO -- DEFINE -- DECLARED IN SEGMENT 2 AT 00000066
00000525 00000532 00001227 00001286 00001397 00001398 00001554 00002312 00002660 00002664 00002674
00002847 00002878 00002896 00003202 00003210 00003211 00003308 00003339
INITIAL -- DEFINE -- DECLARED IN SEGMENT 2 AT 00000211
00000470 00000471 00000472 00000473 00002252 00002359 00002381 00002386 00002427 00002441 00003189
00003216 00003237 00003268 00003390 00003402
INITIALIZE -- PROCEDURE -- DECLARED IN SEGMENT 2 AT 00000466
00003475
INPUTDECL -- BOOLEAN -- DECLARED IN SEGMENT 2 AT 00000163
00001427 00001881 00001939 00002030 00003410 *00003491*
INPUTFILE -- INTEGER -- DECLARED IN SEGMENT 2 AT 00000162
*00000564* 00001427 00001881 00001939 00002030
INRECORD -- BOOLEAN -- DECLARED IN SEGMENT 10 AT 00001162
*00001171* 00001188 00001194 00001199 00001215 *00001230* *00001257*
INSIDFBRACKETS -- BOOLEAN -- DECLARED IN SEGMENT 2 AT 00001155
*00001274* *00002496* 00002504
INSY -- DEFINE -- DECLARED IN SEGMENT 2 AT 00000199
00000912 00001746 00001787
INSYMBOL -- PROCEDURE -- DECLARED IN SEGMENT 2 AT 00000875 -- FORWARD AT 00000226

```



LASTA1 -- REAL -- DECLARED IN SEGMENT 2 AT 00000577  
00000626 \*00000647\*  
LASTBLOCK -- INTEGER -- DECLARED IN SEGMENT 2 AT 00000578  
00000626 \*00000651\*  
LASTCHARPOS -- INTEGER -- DECLARED IN SEGMENT 2 AT 00000862  
\*00001052\* 00001057  
LASTLINE -- FORMAT -- DECLARED IN SEGMENT 3 AT 00000189  
00003548  
LASTREC -- INTEGER -- DECLARED IN SEGMENT 2 AT 00000181  
\*00002792\* 00002792 00003163 00003379 00003429 \*00003433\*\*00003511\*  
LASTSY -- INTEGER -- DECLARED IN SEGMENT 19 AT 00001936  
\*00001967\* 00001993  
LASTWITHSYM -- DEFINE -- DECLARED IN SEGMENT 2 AT 00000075  
00001169 00002511  
LBRACKET -- DEFINE -- DECLARED IN SEGMENT 2 AT 00000199  
00000476 00001182 00001186 00001263 00001580 00002715  
LEFTTYPE -- INTEGER -- DECLARED IN SEGMENT 26 AT 00002200  
\*00002207\* 00002213 00002219 00002220 00002223 \*00002228\*  
LEQSY -- DEFINE -- DECLARED IN SEGMENT 2 AT 00000197  
00000477 00000923 00001031 00001755 00001773  
LETTER -- DEFINE -- DECLARED IN SEGMENT 2 AT 00000857  
00000887 00000891 00000897 00000901 00000961  
LEVEL -- INTEGER -- VALUE PARAMETER -- DECLARED IN SEGMENT 2 AT 00002936  
LEVEL -- INTEGER -- VALUE PARAMETER -- DECLARED IN SEGMENT 2 AT 00002939  
00002937 00002947 00002950 00002998  
LEVEL1000 -- INTEGER -- DECLARED IN SEGMENT 38 AT 00002942  
\*00002947\* 00002961 00002968 00002976 00003011 00003018 00003047 00003059  
LINE -- REAL ARRAY -- DECLARED IN SEGMENT 2 AT 00000128  
00000407 00000480 00000884 00001065 00003475 00003527  
LINECNT -- INTEGER -- DECLARED IN SEGMENT 2 AT 00000132  
\*00000407\*\*00000409\*\*00000499\*\*00000630\*\*00000631\*\*00000641\*\*00000642\*\*00000884\*\*00001042\*\*00001065\*\*00003475\*  
\*00003527\*\*00003690\*  
LINEFFED -- BOOLEAN -- VALUE PARAMETER -- DECLARED IN SEGMENT 2 AT 00001933  
LINEFEED -- BOOLEAN -- VALUE PARAMETER -- DECLARED IN SEGMENT 2 AT 00001934  
00002007  
LINEPNT -- POINTER -- DECLARED IN SEGMENT 2 AT 00000129  
00000407 \*00000480\* 00000482 00000884 00001065 00002245 00002258 00002290 00002343 00002374 00002390  
00003475 00003527  
LINES -- FILE -- DECLARED IN SEGMENT 2 AT 00000034  
00000362 00000407 00000408 00000499 00000619 00000620 00000630 00000631 00000641 00000642 00000884  
00001042 00001065 00003475 00003525 00003527 00003529 00003530 00003552 00003681 00003683 00003685  
00003690  
LINESPERPAGE -- DEFINE -- DECLARED IN SEGMENT 2 AT 00000036  
00000407 00000631 00000642 00000884 00001065 00003475 00003527  
LISTINX -- INTEGER -- DECLARED IN SEGMENT 37 AT 00002813  
\*00002825\*\*00002830\*\*00002831\* 00002833 00002845 \*00002856\*\*00002902\*\*00002903\* 00002904 00002906  
LISTLENGTH -- DEFINE -- DECLARED IN SEGMENT 2 AT 00000051  
00000140 00002812 00002830 00002902 00003250  
LISTOPTION -- BOOLEAN -- DECLARED IN SEGMENT 2 AT 00000157  
00000407 00000884 \*00001043\* 00001065 00003475 \*00003476\* 00003527  
LLIM -- INTEGER -- DECLARED IN SEGMENT 10 AT 00001160  
\*00001192\* 00001196 00001201 00001202  
LLIM -- INTEGER -- DECLARED IN SEGMENT 32 AT 00002397  
\*00002417\* 00002430 00002444  
LLIM -- INTEGER -- DECLARED IN SEGMENT 37 AT 00002814  
\*00002874\*\*00002888\* 00002900  
LPAR -- DEFINE -- DECLARED IN SEGMENT 2 AT 00000199

```

00000476 00001036 00001291 00001409 00001415 00001427 00001439 00001446 00001460 00001475 00001483
00001490 00001534 00001572 00001843 00001881 00001939 00002030 00002048 00002107 00002154 00002665
00002909 00002911 00003081 00003485
LPAFOUND -- BOOLEAN -- DECLARED IN SEGMENT 2 AT 00000698
*00001427* 00001430 *00001881**00001939**00002030*
LSSSY -- DEFINE -- DECLARED IN SEGMENT 2 AT 00000197
00000476 00000922 00001030 00001032 00001746 00001754
LT -- INTEGER -- DECLARED IN SEGMENT 2 AT 00000658
*00001197**00001355**00001597**00001605**00001658**00001721**00001790**00001793**00002075**00002138**00002223*
*00002316**00002433**00002447**00002901*
MAINTYPE -- DEFINE -- DECLARED IN SEGMENT 2 AT 00000086
00001197 00001355 00001597 00001605 00001658 00001721 00001790 00001793 00002075 00002138 00002223
00002316 00002433 00002447 00002660 00002664 00002901
MARGIN -- DEFINE -- DECLARED IN SEGMENT 2 AT 00000368
00002245 00002258 00002290 00002343 00002374 00002390
MARGINCNT -- INTEGER -- DECLARED IN SEGMENT 2 AT 00000130
*00000884**00001065**00002245**00002258**00002290**00002343**00002374**00002390**00003475*
MAXADDR -- INTEGER -- DECLARED IN SEGMENT 37 AT 00002814
00002913 00002914
MAXCASES -- DEFINE -- DECLARED IN SEGMENT 2 AT 00000043
00002286 00002319 00002324 00002326 00002329
MAXCONSTS -- DEFINE -- DECLARED IN SEGMENT 2 AT 00000047
00000116 00003206
MAXEXTFILES -- DEFINE -- DECLARED IN SEGMENT 2 AT 00000052
00000150 00003496 00003498
MAXFILES -- DEFINE -- DECLARED IN SEGMENT 2 AT 00000053
00000152 00002993 00003007
MAXINT -- DEFINE -- DECLARED IN SEGMENT 2 AT 00000037
00000505 00000534 00000995 00002888
MAXLABS -- DEFINE -- DECLARED IN SEGMENT 2 AT 00000044
00000120 00003180
MAXLEVEL -- DEFINE -- DECLARED IN SEGMENT 2 AT 00000042
00000072 00002497
MAXNAMES -- DEFINE -- DECLARED IN SEGMENT 2 AT 00000041
00000057 00000506 00000510 00000514 00000518 00000522 00000525 00000527 00000531 00000540 00000542
00000543 00000544 00000553 00000555 00000556 00000561 00000564 00000566 00001222 00001319 00001345
00001389 00001427 00001554 00001881 00001890 00001939 00002030 00002053 00002082 00002116 00002159
00002312 00002405 00002487 00002555 00002653 00002660 00002664 00002673 00002694 00002832 00002860
00002862 00002868 00002896 00003096 00003109 00003198 00003202 00003226 00003253 00003279 00003292
00003307 00003318 00003384 00003430
MAXPARAMS -- DEFINE -- DECLARED IN SEGMENT 2 AT 00000045
00000107 00003080 00003099
MAXPNTRB -- DEFINE -- DECLARED IN SEGMENT 2 AT 00000054
00000148 00002702
MAXSYMS -- DEFINE -- DECLARED IN SEGMENT 2 AT 00000050
00000142 00001170 00001180 00001189 00001194 00001196 00001201 00001202 00001203 00001205 00001206
00001216 00001226 00001227 00001238 00001243 00001248 00001250 00001266 00001268 00001269 00001285
00001293 00001302 00001303 00001307 00001310 00001311 00001312 00001328 00001329 00001330 00001349
00001357 00001360 00001397 00001398 00001409 00001416 00001428 00001429 00001439 00001445 00001453
00001459 00001463 00001465 00001466 00001467 00001468 00001475 00001477 00001478 00001483 00001484
00001485 00001490 00001491 00001492 00001500 00001505 00001510 00001515 00001520 00001524 00001533
00001534 00001554 00001558 00001563 00001569 00001574 00001577 00001585 00001592 00001600 00001607
00001608 00001626 00001636 00001637 00001641 00001647 00001648 00001651 00001667 00001677 00001680
00001694 00001695 00001701 00001702 00001708 00001709 00001710 00001714 00001727 00001740 00001743
00001744 00001753 00001754 00001755 00001756 00001757 00001758 00001759 00001762 00001768 00001770
00001775 00001779 00001780 00001783 00001795 00001796 00001805 00001841 00001850 00001855 00001858
00001861 00001862 00001863 00002220 00002267 00002353 00002391 00002430 00002444 00002496

```

MAXTABLES -- DEFINE -- DECLARED IN SEGMENT 2 AT 00000040  
00000057 00000136 00003511  
MAXTEMPS -- DEFINE -- DECLARED IN SEGMENT 2 AT 00000048  
00002292 00002434 00003396 00003398  
MAXTYPES -- DEFINE -- DECLARED IN SEGMENT 2 AT 00000046  
00000081 00001614 00002660 00002664 00002668 00002691 00002727 00002762 00002782 00002797  
MAXWITHSYMS -- DEFINE -- DECLARED IN SEGMENT 2 AT 00000049  
00000144 00002505  
MIDDLEF -- DEFINE -- DECLARED IN SEGMENT 2 AT 00000211  
00000382 00000470 00000471 00000472 00000473  
MINUS -- DEFINE -- DECLARED IN SEGMENT 2 AT 00000197  
00000476 00001554 00001678 00001695 00001710 00002312 00002660 00002664 00002891 00002896 00003202  
MODE -- INTEGER -- DECLARED IN SEGMENT 14 AT 00001625  
\*00001628\*\*00001635\*\*00001645\* 00001646 00001666 00001668  
MODE -- INTEGER -- DECLARED IN SEGMENT 15 AT 00001674  
\*00001684\*\*00001693\*\*00001699\* 00001700 \*00001706\* 00001707 00001726 00001728  
MODSY -- DEFINE -- DECLARED IN SEGMENT 2 AT 00000196  
00000916 00001629  
MULOPTR -- INTEGER -- DECLARED IN SEGMENT 14 AT 00001625  
\*00001631\* 00001661 00001663  
N -- INTEGER -- DECLARED IN SEGMENT 36 AT 00002647  
\*00002667\* 00002674 \*00002675\* 00002682  
N -- INTEGER -- VALUE PARAMETER -- DECLARED IN SEGMENT 42 AT 00003542  
N -- INTEGER -- VALUE PARAMETER -- DECLARED IN SEGMENT 42 AT 00003543  
00003544 00003545  
NABS -- INTEGER -- DECLARED IN SEGMENT 2 AT 00000295  
\*00001807\*\*00001900\*\*00001905\*\*00001906\*\*00001907\*\*00001910\*\*00001911\*\*00001912\*\*00001959\*\*00001960\*\*00001973\*  
\*00001974\*\*00001975\*\*00001976\*\*00001992\*\*00001993\*\*00001994\*\*00001995\*\*00001999\*\*00002037\*\*00002065\*\*00002066\*  
\*00002088\*\*00002099\*\*00002128\*\*00002129\*\*00002144\*\*00002165\*\*00002169\*\*00002218\*\*00002222\*\*00002267\*\*00002317\*  
\*00002353\*\*00002391\*\*00002431\*\*00002445\*\*00002582\*\*00002980\*\*00002982\*\*00003040\*\*00003041\*\*00003042\*\*00003054\*  
\*00003056\*  
NAM -- INTEGER -- DECLARED IN SEGMENT 38 AT 00002942  
\*00002950\* 00002961 00002968 00002976 00002995 00002998 00003010 00003011 00003018 00003047 00003059  
NAMELENGTH -- DEFINE -- DECLARED IN SEGMENT 2 AT 00000058  
00000586 00000608 00000646 00000648 00000649 00000903 00002999  
NAMETAB -- DEFINE -- DECLARED IN SEGMENT 2 AT 00000078  
00001319 00001345 00001389 00001427 00001554 00001881 00001890 00001939 00002030 00002053 00002082  
00002116 00002159 00002312 00002405 00002487 00002500 00002555 00002653 00002660 00002664 00002694  
00002860 00002868 00002896 00003109 00003202 00003318  
NAMETAB1 -- REAL ARRAY -- DECLARED IN SEGMENT 2 AT 00000057  
\*00000506\*\*00000510\*\*00000514\*\*00000518\*\*00000522\*\*00000525\*\*00000527\*\*00000531\*\*00000540\*\*00000542\*\*00000543\*  
\*00000544\*\*00000553\*\*00000555\*\*00000556\*\*00000561\*\*00000564\*\*00000566\* 00001222 00001319 00001345 00001389  
00001427 00001554 00001881 00001890 00001939 00002030 00002053 00002082 00002116 00002159 00002312  
00002405 00002487 00002555 00002653 00002660 00002664 \*00002673\* 00002694 \*00002832\* 00002860 \*00002862\*  
00002868 00002896 00002998 \*00003096\* 00003109 \*00003198\* 00003202 \*00003226\*\*00003253\* 00003279 00003292  
\*00003307\* 00003318 00003383 00003430  
NAMETAB2 -- REAL ARRAY -- DECLARED IN SEGMENT 2 AT 00000057  
\*00000506\*\*00000510\*\*00000514\*\*00000518\*\*00000522\*\*00000525\*\*00000527\*\*00000531\*\*00000540\*\*00000542\*\*00000543\*  
\*00000544\*\*00000553\*\*00000555\*\*00000556\*\*00000561\*\*00000564\*\*00000566\* 00001222 00001319 00001345 00001389  
00001427 00001554 00001881 00001890 00001939 00002030 00002053 00002082 00002116 00002159 00002312  
00002405 00002487 00002555 00002653 00002660 00002664 \*00002673\* 00002694 \*00002832\* 00002860 \*00002862\*  
00002868 00002896 \*00003096\* 00003109 \*00003198\* 00003202 \*00003226\*\*00003253\* 00003279 00003292 \*00003307\*  
00003318  
NAMETAB3 -- REAL ARRAY -- DECLARED IN SEGMENT 2 AT 00000057  
\*00000507\*\*00000511\*\*00000515\*\*00000519\*\*00000523\*\*00000525\*\*00000527\*\*00000533\*\*00000540\*\*00000542\*\*00000543\*  
\*00000544\*\*00000553\*\*00000555\*\*00000556\*\*00000561\*\*00000565\*\*00000567\* 00001225 00001319 00001345 00001389  
00001427 00001554 00001881 00001890 00001939 00002030 00002053 00002082 00002116 00002159 00002312

```

00002405 00002487 00002555 00002653 00002660 00002664 *00002674* 00002694 *00002848* 00002860 00002868
*00002879* 00002896 00002950 00003109 *00003136* 00003202 *00003212**00003233**00003265* 00003280 00003293
*00003299**00003310* 00003318 *00003326* 00003339 *00003372*
NAME1 -- REAL -- VALUE PARAMETER -- DECLARED IN SEGMENT 2 AT 00000228
NAME1 -- REAL -- VALUE PARAMETER -- DECLARED IN SEGMENT 2 AT 00000230
00000229
NAME1 -- REAL -- VALUE PARAMETER -- DECLARED IN SEGMENT 2 AT 00000580
NAME1 -- REAL -- VALUE PARAMETER -- DECLARED IN SEGMENT 2 AT 00000582
00000581 00000586 *00000587* 00000591
NAME2 -- REAL -- VALUE PARAMETER -- DECLARED IN SEGMENT 2 AT 00000228
NAME2 -- REAL -- VALUE PARAMETER -- DECLARED IN SEGMENT 2 AT 00000230
00000229
NAME2 -- REAL -- VALUE PARAMETER -- DECLARED IN SEGMENT 2 AT 00000580
NAME2 -- REAL -- VALUE PARAMETER -- DECLARED IN SEGMENT 2 AT 00000582
00000581 *00000588* 00000591
NAMTAB -- INTEGER -- DECLARED IN SEGMENT 38 AT 00002942
*00002950* 00002951 00002952 00002959
NCASELABS -- INTEGER -- DECLARED IN SEGMENT 29 AT 00002287
*00002318* 00002319 00002329
NDIGITS -- DEFINE -- DECLARED IN SEGMENT 2 AT 00000235
00000399 00000407 00000499 00000631 00000642 00000884 00001042 00001065 00002245 00002258 00002290
00002345 00002374 00002390 00003475 00003527 00003690
NEGATIVE -- BOOLEAN -- DECLARED IN SEGMENT 2 AT 00000729
*00001554**00002312**00002660**00002664**00002896**00003202*
NEGEXP -- BOOLEAN -- DECLARED IN SEGMENT 8 AT 00000878
*00000981* 00000985
NEQSY -- DEFINE -- DECLARED IN SEGMENT 2 AT 00000198
00000477 00000926 00001033 00001759 00001769 00001780
NEWCARD -- DEFINE -- DECLARED IN SEGMENT 2 AT 00000263
00000884 00001065 00003475
NEWDISP -- PROCEDURE -- DECLARED IN SEGMENT 2 AT 00002148
00002181 00002576 00002578
NEWNAME -- DEFINE -- DECLARED IN SEGMENT 2 AT 00000449
00000506 00000510 00000514 00000518 00000522 00000525 00000527 00000531 00000540 00000542 00000543
00000544 00000553 00000555 00000556 00000561 00000564 00000566 00002673 00002832 00002862 00003096
00003198 00003226 00003253 00003307
NEWPARAM -- DEFINE -- DECLARED IN SEGMENT 39 AT 00003073
00003080 00003099
NEWTRY -- LABEL -- DECLARED IN SEGMENT 31 AT 00002371 -- OCCURS AT 00002378
00002380 00002381 00002388
NEWTYP -- DEFINE -- DECLARED IN SEGMENT 2 AT 00000351
00001614 00002660 00002664 00002668 00002691 00002727 00002762 00002782 00002797
NEWXREF -- PROCEDURE -- DECLARED IN SEGMENT 2 AT 00000580 -- FORWARD AT 00000228
00000506 00000510 00000514 00000518 00000522 00000525 00000527 00000531 00000540 00000542 00000543
00000544 00000553 00000555 00000556 00000561 00000564 00000566 00000592 00001222 00001319 00001345
00001389 00001427 00001554 00001881 00001890 00001939 00002030 00002053 00002082 00002116 00002159
00002312 00002405 00002487 00002555 00002653 00002660 00002664 00002673 00002694 00002832 00002860
00002862 00002868 00002896 00003096 00003109 00003198 00003202 00003226 00003253 00003279 00003292
00003307 00003318
NEXTCHAR -- DEFINE -- DECLARED IN SEGMENT 2 AT 00000865
00000885 00000890 00000893 00000899 00000901 00000965 00000966 00000969 00000973 00000980 00000981
00000989 00000984 00001002 00001005 00001011 00001025 00001027 00001029 00001031 00001033 00001035
00001038 00001041 00001054 00001059 00001066 00001067 00001070
NILSY -- DEFINE -- DECLARED IN SEGMENT 2 AT 00000205
00000917 00001567
NILTYPE -- INTEGER -- DECLARED IN SEGMENT 2 AT 00000161
*00000529* 00001197 00001355 00001569 00001597 00001605 00001658 00001721 00001790 00001793 00002075

```

```
00002138 00002223 00002316 00002433 00002447 00002901
NL -- INTEGER -- DECLARED IN SEGMENT 2 AT 00000578
*00000586* 00000587 00000588
NOALIST -- FORMAT -- DECLARED IN SEGMENT 3 AT 00000188
00001050
NOERRORS -- FORMAT -- DECLARED IN SEGMENT 3 AT 00000185
00003552
NOTSY -- DEFINE -- DECLARED IN SEGMENT 2 AT 00000195
00000919 00001556
NPARS -- INTEGER -- DECLARED IN SEGMENT 11 AT 00001281
*00001288**00001289* 00001296 *00001356* 00001363
NSIZE -- INTEGER -- DECLARED IN SEGMENT 2 AT 00000295
*00001807**00001900**00001905**00001906**00001907**00001910**00001911**00001912**00001959**00001960**00001973*
*00001974**00001975**00001976**00001992**00001993**00001994**00001995**00001999**00002037**00002065**00002066*
*00002088**00002099**00002128**00002129**00002144**00002165**00002169**00002218**00002222**00002267**00002317*
*00002352**00002391**00002431**00002445**00002582**00002980**00002982**00003040**00003041**00003042**00003054*
*00003056*
NUMBEGINS -- INTEGER -- DECLARED IN SEGMENT 2 AT 00000173
*00002245*
NUMBER -- DEFINE -- DECLARED IN SEGMENT 2 AT 00000212
00001275 00001364 00001386 00001399 00001562 00001585 00001635 00001693
NUMBLOCKS -- INTEGER -- DECLARED IN SEGMENT 2 AT 00000137
*00002793**00003380*
NUMCASES -- INTEGER -- DECLARED IN SEGMENT 2 AT 00000174
*00002290*
NUMCONSTS -- INTEGER -- DECLARED IN SEGMENT 2 AT 00000117
*00000534* 00003160 00003206 *00003207**00003208* 00003209 00003210 *00003436*
NUMERIC -- DEFINE -- DECLARED IN SEGMENT 2 AT 00000083
00000504 00001197 00001355 00001409 00001417 00001439 00001475 00001483 00001490 00001534 00001554
00001597 00001605 00001633 00001657 00001658 00001687 00001692 00001720 00001721 00001789 00001790
00001793 00001855 00001858 00001861 00001897 00001905 00001970 00001973 00001981 00001987 00002075
00002138 00002223 00002312 00002316 00002433 00002447 00002660 00002664 00002896 00002901 00003202
NUMERRS -- INTEGER -- DECLARED IN SEGMENT 2 AT 00000027
*00000394* 00003531 00003681
NUMEXTFILES -- INTEGER -- DECLARED IN SEGMENT 2 AT 00000151
00003002 *00003495* 00003496 00003497 00003498
NUMFILES -- INTEGER -- DECLARED IN SEGMENT 2 AT 00000153
00002993 *00002994* 00002995 00003007 *00003009* 00003010 00003164 00003413 *00003425*
NUMFORWARDS -- INTEGER -- DECLARED IN SEGMENT 40 AT 00003149
*00003300**00003373* 00003394
NUMLABS -- INTEGER -- DECLARED IN SEGMENT 2 AT 00000123
00002469 00002536 00003159 00003178 *00003180**00003181* 00003182 00003431 *00003434*
NUMPARAMS -- INTEGER -- DECLARED IN SEGMENT 2 AT 00000113
*00003080* 00003091 *00003099* 00003115 00003121 00003135 00003143 00003162 00003308 *00003438*
NumpNTRS -- INTEGER -- DECLARED IN SEGMENT 2 AT 00000149
*00002702* 00002703 00002704 00003273 00003276 *00003284*
NumpNTRSINWITH -- DEFINE -- DECLARED IN SEGMENT 2 AT 00000076
00001173 00002502
NUMPOINTERS -- INTEGER -- DECLARED IN SEGMENT 2 AT 00001156
*00001173**00001256* 00001267 *00001270* 00002502 *00002518*
NUMREPS -- INTEGER -- DECLARED IN SEGMENT 2 AT 00000175
*00002373*
NUMSYMS -- INTEGER -- DECLARED IN SEGMENT 2 AT 00000143
00001165 *00001170**00001180**00001189**00001194**00001196**00001201**00001202**00001203**00001205**00001206*
*00001215**00001226**00001227**00001238* 00001242 *00001243**00001248* 00001250 00001252 *00001256**00001266*
*00001268**00001269**00001285**00001293**00001302**00001303**00001307**00001310**00001311**00001312**00001328*
*00001329**00001330**00001349**00001357**00001360**00001397**00001398**00001409**00001416**00001428**00001429*
```



\*00001439\*\*00001445\*\*00001453\*\*00001459\*\*00001463\*\*00001465\*\*00001466\*\*00001467\*\*00001468\*\*00001475\*\*00001477\*  
\*00001478\*\*00001483\*\*00001484\*\*00001485\*\*00001490\*\*00001491\*\*00001492\*\*00001500\*\*00001505\*\*00001510\*\*00001515\*  
\*00001520\*\*00001524\*\*00001533\*\*00001534\*\*00001554\*\*00001558\*\*00001563\*\*00001569\*\*00001574\*\*00001577\*\*00001585\*  
\*00001592\*\*00001600\*\*00001607\*\*00001608\*\*00001626\*\*00001636\*\*00001637\*\*00001641\*\*00001647\*\*00001648\*\*00001651\*  
\*00001667\*\*00001677\*\*00001680\*\*00001694\*\*00001695\*\*00001701\*\*00001702\*\*00001708\*\*00001709\*\*00001710\*\*00001714\*  
\*00001727\*\*00001740\* 00001741 \*00001743\*\*00001744\*\*00001753\*\*00001754\*\*00001755\*\*00001756\*\*00001757\*\*00001758\*  
\*00001759\*\*00001762\*\*00001768\*\*00001770\*\*00001775\*\*00001779\*\*00001780\*\*00001783\*\*00001795\*\*00001796\*\*00001805\*  
\*00001807\*\*00001841\*\*00001850\*\*00001855\*\*00001858\*\*00001861\*\*00001862\*\*00001863\*\*00001900\*\*00002088\*\*00002165\*  
\*00002218\*\*00002220\*\*00002222\*\*00002267\*\*00002353\*\*00002391\*\*00002430\*\*00002431\*\*00002444\*\*00002445\*\*00002496\*  
00002505 00002506 \*00002517\*\*00002582\*  
NUMTEMPS -- INTEGER -- DECLARED IN SEGMENT 2 AT 00000176  
\*00002292\*\*00002344\*\*00002434\* 00002443 00002457 \*00002460\*  
NUMTYPES -- INTEGER -- DECLARED IN SEGMENT 2 AT 00000104  
\*00000528\*\*00000558\*\*00001614\*\*00002660\*\*00002664\*\*00002668\*\*00002691\*\*00002727\*\*00002762\*\*00002782\*\*00002797\*  
00003161 \*00003437\*  
NUMXRF -- INTEGER -- DECLARED IN SEGMENT 2 AT 00000137  
00000628 \*00000632\*\*00000636\*\*00000643\*  
NWITHSYMS -- INTEGER -- DECLARED IN SEGMENT 2 AT 00000145  
00002482 00002503 00002505 00002508 \*00002509\* 00002511 \*00002525\*  
OFSY -- DEFINE -- DECLARED IN SEGMENT 2 AT 00000203  
00000911 00002298 00002300 00002301 00002734 00002752 00002772 00002882 00002883 00002884 00002885  
00002886 00002887  
ORSY -- DEFINE -- DECLARED IN SEGMENT 2 AT 00000197  
00000910 00001689 00001702  
OUTPUTDECL -- BOOLEAN -- DECLARED IN SEGMENT 2 AT 00000163  
00001427 00001881 00001939 00002030 \*00003492\*  
OUTPUTFILE -- INTEGER -- DECLARED IN SEGMENT 2 AT 00000162  
\*00000567\* 00001427 00001881 00001939 00002030  
OVERFLOW -- LABEL -- DECLARED IN SEGMENT 8 AT 00000879 -- OCCURS AT 00000997  
00000991  
P -- INTEGER -- DECLARED IN SEGMENT 11 AT 00001281  
\*00001286\*\*00001288\*\*00001297\*  
P -- POINTER -- DECLARED IN SEGMENT 19 AT 00001937  
\*00001951\*\*00001956\*  
PACK -- PROCEDURE -- DECLARED IN SEGMENT 2 AT 00002043  
00002100 00002579  
PACKED -- BOOLEAN -- DECLARED IN SEGMENT 2 AT 00002612  
\*00002650\*\*00002710\*  
PACKEDSY -- DEFINE -- DECLARED IN SEGMENT 2 AT 00000209  
00000944 00002710  
PAGECNT -- INTEGER -- DECLARED IN SEGMENT 2 AT 00000132  
\*00000407\*\*00000499\*\*00000631\*\*00000642\*\*00000884\*\*00001042\*\*00001065\*\*00003475\*\*00003527\*\*00003690\*  
PARAM -- INTEGER -- DECLARED IN SEGMENT 11 AT 00001281  
\*00001297\* 00001298 00001299 00001315 00001325 00001348  
PARAM -- BOOLEAN -- VALUE PARAMETER -- DECLARED IN SEGMENT 2 AT 00002936  
PARAM -- BOOLEAN -- VALUE PARAMETER -- DECLARED IN SEGMENT 2 AT 00002940  
00002937 00002981 00002991 00003005 00003020 00003051 00003055  
PARAMETER -- DEFINE -- DECLARED IN SEGMENT 12 AT 00001374  
00001409 00001439 00001475 00001483 00001490 00001534  
PARAMETERLIST -- PROCEDURE -- DECLARED IN SEGMENT 2 AT 00003068  
00003144 00003311  
PARAMFILE -- DEFINE -- DECLARED IN SEGMENT 2 AT 00000112  
00001329 00003122 00003345  
PARAMKIND -- DEFINE -- DECLARED IN SEGMENT 2 AT 00000109  
00001299 00001315 00001348 00003097 00003357  
PARAMLEVEL -- DEFINE -- DECLARED IN SEGMENT 2 AT 00000110  
00003098

```

PARAMNAME -- DEFINE -- DECLARED IN SEGMENT 2 AT 00000108
00003136 00003344 00003348 00003350 00003363
PARAMTAB -- REAL ARRAY -- DECLARED IN SEGMENT 2 AT 00000107
00001288 00001297 *00003099**00003116**00003122* 00003136 *00003143* 00003339 00003344 00003345 00003348
00003350 00003357 00003363 00003366
PARAMTABTOP -- INTEGER -- DECLARED IN SEGMENT 40 AT 00003155
*00003162* 00003438
PARAMTYPE -- DEFINE -- DECLARED IN SEGMENT 2 AT 00000111
00001298 00003116
PASCALGOL -- FILE -- DECLARED IN SEGMENT 2 AT 00000035
00000361 00001049 00001050 00003461 00003470 00003548 00003549
PASCRUN -- FILE -- DECLARED IN SEGMENT 41 AT 00003463
00003469
PASSPARAMS -- PROCEDURE -- DECLARED IN SEGMENT 2 AT 00001279
00001365 00001541 00001547 00002581
PLUS -- DEFINE -- DECLARED IN SEGMENT 2 AT 00000196
00000475 00001554 00001678 00001680 00001689 00001694 00001709 00002312 00002660 00002664 00002891
00002896 00003202
PNTRTAB1 -- REAL ARRAY -- DECLARED IN SEGMENT 2 AT 00000148
*00002703* 00003278
PNTRTAB2 -- REAL ARRAY -- DECLARED IN SEGMENT 2 AT 00000148
*00002703* 00003278
PNTRTAB3 -- REAL ARRAY -- DECLARED IN SEGMENT 2 AT 00000148
*00002704* 00003282
POINTERS -- DEFINE -- DECLARED IN SEGMENT 2 AT 00000092
00001197 00001246 00001355 00001597 00001605 00001658 00001721 00001777 00001790 00001793 00002075
00002138 00002163 00002223 00002316 00002433 00002447 00002691 00002901 00003324
POINTTYPE -- DEFINE -- DECLARED IN SEGMENT 2 AT 00000093
00001197 00001259 00001355 00001597 00001605 00001658 00001721 00001790 00001793 00002075 00002138
00002166 00002223 00002316 00002433 00002447 00002698 00002901 00003282
POWER -- INTEGER -- DECLARED IN SEGMENT 4 AT 00000317
*00000332**00000333* 00000339 *00000340* 00000341
PREAD -- PROCEDURE -- DECLARED IN SEGMENT 2 AT 00001875
00001930 00002568 00002569
PRINTERRORS -- PROCEDURE -- DECLARED IN SEGMENT 2 AT 00000405
00000412 00000884 00001065 00003475 00003528
PRINTLINE -- DEFINE -- DECLARED IN SEGMENT 2 AT 00000249
00000407 00000884 00001065 00003475 00003527
PRINTXREF -- PROCEDURE -- DECLARED IN SEGMENT 2 AT 00000613
00000654 00003691
PROC -- DEFINE -- DECLARED IN SEGMENT 2 AT 00000064
00000536 00001545 00002561 00003088 00003105 00003131 00003294 00003301 00003309
PROCNUM -- INTEGER -- VALUE PARAMETER -- DECLARED IN SEGMENT 2 AT 00002018
PROCNUM -- INTEGER -- VALUE PARAMETER -- DECLARED IN SEGMENT 2 AT 00002019
00002022 00002033
PROCSY -- DEFINE -- DECLARED IN SEGMENT 2 AT 00000208
00000949 00003085 00003287 00003404
PROGNAME -- ALPHA -- DECLARED IN SEGMENT 2 AT 00003455
*00003483* 00003554
PROGNAMELENGTH -- INTEGER -- DECLARED IN SEGMENT 2 AT 00003454
*00003483* 00003554
PROGRAMSY -- DEFINE -- DECLARED IN SEGMENT 2 AT 00000208
00000949 00003478
PTYPE -- INTEGER -- DECLARED IN SEGMENT 11 AT 00001281
*00001298* 00001301 00001310 00001311 00001355
PUTCONST -- DEFINE -- DECLARED IN SEGMENT 2 AT 00001110
00001196 00001201 00001202 00001205 00001227 00001310 00001311 00001312 00001397 00001398 00001416

```

```

00001465 00001466 00001467 00001478 00001485 00001492 00001554 00001569 00001585 00001607 00001862
00002220 00002430 00002444
PUTDUMMY -- DEFINE -- DECLARED IN SEGMENT 2 AT 00001116
00001303 00001558 00001626 00001651 00001677 00001714 00001740 00001743 00001744 00001783 00002267
00002353 00002391
PUTID -- DEFINE -- DECLARED IN SEGMENT 2 AT 00001121
00001180 00001285 00001328 00001329 00001330 00001349 00001428
PUTSYM -- DEFINE -- DECLARED IN SEGMENT 2 AT 00001104
00001189 00001196 00001201 00001202 00001203 00001205 00001206 00001216 00001226 00001243 00001248
00001266 00001293 00001307 00001310 00001311 00001312 00001328 00001329 00001357 00001360 00001409
00001416 00001439 00001445 00001453 00001463 00001465 00001466 00001467 00001468 00001475 00001477
00001478 00001483 00001484 00001485 00001490 00001491 00001492 00001534 00001563 00001574 00001577
00001600 00001607 00001636 00001637 00001647 00001667 00001680 00001694 00001695 00001701 00001708
00001727 00001753 00001754 00001755 00001756 00001757 00001758 00001759 00001762 00001768 00001770
00001775 00001779 00001780 00001795 00001796 00001805 00001841 00001850 00001855 00001858 00001861
00001862 00001863 00002220 00002267 00002353 00002391 00002430 00002444 00002496
PUTTEXT -- DEFINE -- DECLARED IN SEGMENT 2 AT 00001095
00001170 00001180 00001189 00001194 00001196 00001201 00001202 00001203 00001205 00001206 00001216
00001226 00001227 00001238 00001243 00001248 00001266 00001268 00001269 00001285 00001293 00001302
00001303 00001307 00001310 00001311 00001312 00001328 00001329 00001330 00001349 00001357 00001360
00001397 00001398 00001409 00001416 00001428 00001429 00001439 00001445 00001453 00001459 00001463
00001465 00001466 00001467 00001468 00001475 00001477 00001478 00001483 00001484 00001485 00001490
00001491 00001492 00001500 00001505 00001510 00001515 00001520 00001524 00001528 00001533 00001534
00001554 00001558 00001563 00001569 00001574 00001577 00001585 00001592 00001600 00001607 00001608
00001626 00001636 00001637 00001641 00001647 00001648 00001651 00001667 00001677 00001680 00001694
00001695 00001701 00001702 00001708 00001709 00001710 00001714 00001727 00001740 00001743 00001744
00001753 00001754 00001755 00001756 00001757 00001758 00001759 00001762 00001768 00001770 00001775
00001779 00001780 00001783 00001795 00001796 00001805 00001841 00001850 00001855 00001858 00001861
00001862 00001863 00002220 00002267 00002353 00002391 00002430 00002444 00002496
PWRITE -- PROCEDURE -- DECLARED IN SEGMENT 2 AT 00001933
00002015 00002565 00002567
PX -- INTEGER -- DECLARED IN SEGMENT 39 AT 00003070
*00003097* 00003098* 00003099
P1 -- INTEGER -- DECLARED IN SEGMENT 39 AT 00003070
*00003091* 00003115 00003121 00003135
QUOTES -- DEFINE -- DECLARED IN SEGMENT 2 AT 00000856
00001000 00001005
RBRACKET -- DEFINE -- DECLARED IN SEGMENT 2 AT 00000200
00000477 00001208 00001209 00001210 00001583 00001610 00001611 00001612 00002733
REALCONST -- DEFINE -- DECLARED IN SEGMENT 2 AT 00000194
00000971 00000980 00000989 00001554 00002312 00002660 00002664 00002896 00003202
REALOVERFLOW -- UNKNOWN -- DECLARED IN SEGMENT 2 AT 00000191
00000991 00000993 00000997
REALTYPE -- INTEGER -- DECLARED IN SEGMENT 2 AT 00000161
*00000505* 00001197 00001355 00001410 00001476 00001486 00001493 00001501 00001511 00001516 00001535
00001536 00001554 00001597 00001605 00001657 00001658 00001659 00001661 00001663 00001720 00001721
00001722 00001789 00001790 00001793 00001870 00002075 00002138 00002223 00002312 00002316 00002433
00002447 00002660 00002664 00002896 00002901 00002983 00003202
REALVAR -- BOOLEAN -- DECLARED IN SEGMENT 38 AT 00002943
00002954 *00002957* 00002966 *00002967* 00002971* 00002989 *00002990* 00003064
RECINX -- INTEGER -- DECLARED IN SEGMENT 36 AT 00002647
*00002792* 00002793 00002795 00002798
RECORD -- DEFINE -- DECLARED IN SEGMENT 2 AT 00000097
00001218 00002494 00002798
RECORDSY -- DEFINE -- DECLARED IN SEGMENT 2 AT 00000206
00000383 00000943 00002789
RECSIZE -- INTEGER -- DECLARED IN SEGMENT 38 AT 00002942

```

```

*00003037* 00003040 00003041 00003042
RECTAR -- DEFINE -- DECLARED IN SEGMENT 2 AT 00000098
00001222 00001225 00002500 00002798
RECTAB -- INTEGER -- VALUE PARAMETER -- DECLARED IN SEGMENT 2 AT 00002614
RECTAB -- INTEGER -- VALUE PARAMETER -- DECLARED IN SEGMENT 2 AT 00002616
00002615
RECTAB -- INTEGER -- VALUE PARAMETER -- DECLARED IN SEGMENT 2 AT 00002808
RECTAB -- INTEGER -- VALUE PARAMETER -- DECLARED IN SEGMENT 2 AT 00002810
00002809 00002832 00002848 00002862 00002879 00002913
RECTYPE -- DEFINE -- DECLARED IN SEGMENT 2 AT 00000073
00001175 00002501
RELOPTR -- INTEGER -- DECLARED IN SEGMENT 16 AT 00001734
*00001749* 00001787
REPEATSTAT -- PROCEDURE -- DECLARED IN SEGMENT 2 AT 00002368
00002392 00002590
REPEATSY -- DEFINE -- DECLARED IN SEGMENT 2 AT 00000203
00000941 00002590
REPNUM -- INTEGER -- DECLARED IN SEGMENT 31 AT 00002370
*00002373* 00002374 00002390
RESWORDOPTION -- BOOLEAN -- DECLARED IN SEGMENT 2 AT 00000157
00000407 00000884 00000955 *00001044* 00001065 00003475 00003527
RPAR -- DEFINE -- DECLARED IN SEGMENT 2 AT 00000199
00000477 00001296 00001359 00001361 00001409 00001418 00001419 00001432 00001433 00001439 00001450
00001469 00001470 00001475 00001483 00001490 00001534 00001576 00001864 00001865 00001866 00001867
00001868 00001869 00001871 00001885 00001920 00001921 00001943 00002003 00002004 00002038 00002039
00002094 00002095 00002096 00002097 00002139 00002140 00002141 00002142 00002176 00002178 00002179
00002180 00002678 00002683 00002886 00002915 00003138 00003139 00003140 00003503 00003504
RT -- INTEGER -- DECLARED IN SEGMENT 2 AT 00000658
*00001197**00001355**00001597**00001605**00001658**00001721**00001790**00001793**00002075**00002138**00002223*
*00002316**00002433**00002447**00002901*
SAVEFACTOR -- INTEGER -- DECLARED IN SEGMENT 2 AT 00000028
00003557 00003558 00003564 00003566 *00000973**00000987* 00000992
SEARCH -- DEFINE -- DECLARED IN SEGMENT 2 AT 00000436
00001319 00001345 00001389 00001427 00001554 00001881 00001890 00001939 00002030 00002053 00002082
00002116 00002159 00002312 00002405 00002487 00002555 00002653 00002660 00002664 00002694 00002860
00002868 00002896 00003109 00003202 00003318
SEARCHTAB -- DEFINE -- DECLARED IN SEGMENT 2 AT 00000421
00001222 00001319 00001345 00001389 00001427 00001554 00001881 00001890 00001939 00002030 00002053
00002082 00002116 00002159 00002312 00002405 00002487 00002555 00002653 00002660 00002664 00002694
00002860 00002868 00002896 00003109 00003202 00003279 00003292 00003318
SEMICOLON -- DEFINE -- DECLARED IN SEGMENT 2 AT 00000200
00000386 00000477 00002248 00002253 00002305 00002337 00002338 00002340 00002341 00002382 00002595
00002821 00002851 00002890 00002919 00003137 00003187 00003213 00003214 00003215 00003234 00003235
00003236 00003266 00003267 00003304 00003333 00003335 00003336 00003388 00003403 00003503 00003505
00003506 00003507 00003508
SET -- DEFINE -- DECLARED IN SEGMENT 2 AT 00000090
00001197 00001355 00001597 00001605 00001614 00001619 00001643 00001649 00001658 00001704 00001721
00001765 00001790 00001792 00001793 00002075 00002138 00002223 00002316 00002433 00002447 00002783
00002787 00002901
SETSY -- DEFINE -- DECLARED IN SEGMENT 2 AT 00000207
00000921 00002769
SETTYPE -- DEFINE -- DECLARED IN SEGMENT 2 AT 00000091
00001197 00001355 00001597 00001605 00001615 00001658 00001721 00001790 00001793 00002075 00002138
00002223 00002316 00002433 00002447 00002783 00002901
SIGNED -- BOOLEAN -- DECLARED IN SEGMENT 2 AT 00000729
*00001554**00002312**00002660**00002664**00002896**00003202*
SIGNED -- BOOLEAN -- DECLARED IN SEGMENT 15 AT 00001675
*00001679* 00001685

```

```

SIMPLEEXPRESSION  -- PROCEDURE  -- DECLARED IN SEGMENT 2 AT 00001672
00001729 00001745 00001784
SIMPLEVARIABLE  -- BOOLEAN  -- DECLARED IN SEGMENT 2 AT 00001155
*00001174**00001179**00001184* 00001336 00002495
SIZE  -- DEFINE  -- DECLARED IN SEGMENT 2 AT 00000102
00000504 00001204 00001205 00001614 00002167 00002658 00002660 00002664 00002680 00002692 00002726
00002743 00002764 00002784 00002799
SKIP  -- PROCEDURE  -- DECLARED IN SEGMENT 2 AT 00000379
00000385 00000388 00001209 00001296 00001359 00001409 00001418 00001432 00001439 00001450 00001469
00001475 00001483 00001490 00001534 00001576 00001611 00001864 00001865 00001866 00001867 00001868
00001869 00001920 00002003 00002038 00002094 00002095 00002096 00002139 00002140 00002141 00002178
00002179 00002209 00002233 00002255 00002270 00002300 00002334 00002338 00002342 00002356 00002387
00002425 00002438 00002450 00002521 00002547 00002583 00002595 00002678 00002734 00002801 00002803
00002840 00002882 00002883 00002884 00002885 00002886 00002909 00002915 00003139 00003184 00003187
00003213 00003214 00003215 00003234 00003235 00003236 00003256 00003258 00003266 00003267 00003304
00003333 00003335 00003336 00003388 00003403 00003503 00003505 00003506 00003507 00003508
SLASH  -- DEFINE  -- DECLARED IN SEGMENT 2 AT 00000195
00000477 00001637 00001663
START  -- LABEL  -- DECLARED IN SEGMENT 8 AT 00000879  -- OCCURS AT 00000881
00000884 00001071
START  -- LABEL  -- DECLARED IN SEGMENT 40 AT 00003153  -- OCCURS AT 00003168
00003405
STARTLEVEL  -- INTEGER  -- DECLARED IN SEGMENT 34 AT 00002480
*00002482* 00002525
STARTSYM  -- INTEGER  -- DECLARED IN SEGMENT 10 AT 00001160
*00001165* 00001252 00001254 00001255
STARTSYM  -- INTEGER  -- DECLARED IN SEGMENT 12 AT 00001370
*00001558* 00001563 *00001592* 00001600
STARTSYM  -- INTEGER  -- DECLARED IN SEGMENT 14 AT 00001625
*00001628* 00001647 *00001651* 00001667
STARTSYM  -- INTEGER  -- DECLARED IN SEGMENT 15 AT 00001674
*00001677* 00001701 00001708 *00001714* 00001727
STARTSYM  -- INTEGER  -- DECLARED IN SEGMENT 16 AT 00001734
*00001743* 00001753 00001762 00001768 00001773 00001774 *00001783* 00001795
STATEMENT  -- PROCEDURE  -- DECLARED IN SEGMENT 2 AT 00002529  -- FORWARD AT 00002196
00002249 00002276 00002278 00002336 00002363 00002378 00002458 00002524 00002596
STATM  -- LABEL  -- DECLARED IN SEGMENT 27 AT 00002243  -- OCCURS AT 00002249
00002251 00002252 00002256
STATM  -- LABEL  -- DECLARED IN SEGMENT 30 AT 00002351  -- OCCURS AT 00002363
00002359
STATM  -- LABEL  -- DECLARED IN SEGMENT 32 AT 00002399  -- OCCURS AT 00002458
00002427 00002441
STRING  -- REAL ARRAY  -- DECLARED IN SEGMENT 2 AT 00000126
00000490
STRINGPNT  -- POINTER  -- DECLARED IN SEGMENT 2 AT 00000127
*00000490* 00001009 00001016 00001020 00001951
STRUCT  -- DEFINE  -- DECLARED IN SEGMENT 2 AT 00000103
00000504 00000559 00001206 00001238 00001264 00001335 00001614 00002213 00002660 00002664 00002679
00002692 00002738 00002742 00002760 00002765 00002783 00002798 00002963 00002985 00003038 00003049
00003123 00003124
STYPE  -- INTEGER  -- DECLARED IN SEGMENT 12 AT 00001370
00001594 *00001595* 00001597 00001602 *00001603* 00001605 00001615
SUBRANGE  -- DEFINE  -- DECLARED IN SEGMENT 2 AT 00002619
00002660 00002664
SUBTYPE  -- DEFINE  -- DECLARED IN SEGMENT 2 AT 00000085
00001197 00001355 00001597 00001605 00001658 00001721 00001790 00001793 00002075 00002138 00002223
00002316 00002433 00002447 00002660 00002664 00002901

```

```

SX -- REAL -- DECLARED IN SEGMENT 2 AT 00001092
*00001807**00001900**00002088**00002165**00002218**00002222**00002267**00002353**00002391**00002431**00002445*
*00002582*
SX -- INTEGER -- DECLARED IN SEGMENT 36 AT 00002647
00002719 00002736 *00002743* 00002746 00002756 00002764 00002776 00002795 *00002796* 00002799 00002800
SX -- INTEGER -- DECLARED IN SEGMENT 37 AT 00002814
00002842 00002847
SYMBOL -- INTEGER ARRAY -- DECLARED IN SEGMENT 2 AT 00000146
*00000475* 00001025
SYMBOL -- INTEGER -- VALUE PARAMETER -- DECLARED IN SEGMENT 2 AT 00000379
SYMBOL -- INTEGER -- VALUE PARAMETER -- DECLARED IN SEGMENT 2 AT 00000380
00000382
SYMBOLIC -- DEFINE -- DECLARED IN SEGMENT 2 AT 00000084
00000517 00002660 00002664 00002679
SYMKIND -- INTEGER ARRAY -- DECLARED IN SEGMENT 2 AT 00000147
00000382 *00000470* 00001427 00001881 00001883 00001939 00001941 00002030 00002210 00002252 00002273
00002359 00002381 00002386 00002427 00002441 00002594 00003189 00003216 00003237 00003268 00003390
00003402
SYMTAB -- REAL ARRAY -- DECLARED IN SEGMENT 2 AT 00000142
*00001170**00001180**00001189**00001194**00001196**00001201**00001202**00001203**00001205**00001206**00001216*
*00001226**00001227**00001238**00001242**00001243**00001248**00001253**00001254**00001255**00001266**00001268*
*00001269**00001285**00001293**00001302**00001303**00001307**00001310**00001311**00001312**00001328**00001329*
*00001330**00001349**00001357**00001360**00001397**00001398**00001409**00001416**00001428**00001429**00001439*
*00001445**00001453**00001459**00001463**00001465**00001466**00001467**00001468**00001475**00001477**00001478*
*00001489**00001484**00001485**00001490**00001491**00001492**00001500**00001505**00001510**00001515**00001520*
*00001524**00001533**00001534**00001554**00001558**00001563**00001569**00001574**00001577**00001585**00001592*
*00001600**00001607**00001608**00001626**00001636**00001637**00001641**00001647**00001648**00001651**00001667*
*00001677**00001680**00001694**00001695**00001701**00001702**00001708**00001709**00001710**00001714**00001727*
*00001740**00001743**00001744**00001753**00001754**00001755**00001756**00001757**00001758**00001759**00001762*
*00001768**00001770**00001773**00001774**00001775**00001779**00001780**00001783**00001795**00001796**00001804*
*00001805* 00001807 *00001841**00001850**00001855**00001858**00001861**00001862**00001863* 00001900 00002088
00002165 00002218 *00002220* 00002222 *00002267**00002353**00002391**00002430* 00002431 *00002444* 00002445
*00002496* 00002508 00002582
T -- REAL -- DECLARED IN SEGMENT 10 AT 00001161
*00001168* 00001169 00001172 00001173 *00001175**00001191* 00001193 00001197 00001198 *00001217* 00001218
00001222 00001225 *00001234* 00001235 00001237 00001240 00001246 00001259
T -- INTEGER -- DECLARED IN SEGMENT 12 AT 00001370
*00001540* 00001542
T -- INTEGER -- DECLARED IN SEGMENT 23 AT 00002045
*00002058* 00002059 00002061 00002062
T -- INTEGER -- DECLARED IN SEGMENT 24 AT 00002105
*00002121* 00002122 00002124 00002125
T -- INTEGER -- DECLARED IN SEGMENT 29 AT 00002287
*00002324* 00002325 *00002326* 00002327
T -- INTEGER -- DECLARED IN SEGMENT 36 AT 00002647
*00002716* 00002723 *00002730* 00002739 00002741 *00002744**00002758* 00002759 00002760 00002763
T -- INTEGER -- DECLARED IN SEGMENT 39 AT 00003070
*00003119* 00003120 00003123 00003124
T -- INTEGER -- DECLARED IN SEGMENT 40 AT 00003149
*00003323* 00003324 *00003339* 00003340 00003343 00003356 00003366 *00003381* 00003385
TAB -- INTEGER ARRAY -- NAME PARAMETER -- DECLARED IN SEGMENT 2 AT 00002936
TAB -- INTEGER ARRAY -- NAME PARAMETER -- DECLARED IN SEGMENT 2 AT 00002938
00002950
TABLE -- INTEGER -- VALUE PARAMETER -- DECLARED IN SEGMENT 2 AT 00000228
TABLE -- INTEGER -- VALUE PARAMETER -- DECLARED IN SEGMENT 2 AT 00000231
00000229
TABLE -- INTEGER -- VALUE PARAMETER -- DECLARED IN SEGMENT 2 AT 00000580
TABLE -- INTEGER -- VALUE PARAMETER -- DECLARED IN SEGMENT 2 AT 00000583

```

0000058, 00000589  
TEMPSYM -- ALPHA -- DECLARED IN SEGMENT 2 AT 00001091  
\*00001189\*\*00001196\*\*00001201\*\*00001202\*\*00001203\*\*00001205\*\*00001206\*\*00001216\*\*00001226\*\*00001243\*\*00001248\*  
\*00001268\*\*00001293\*\*00001307\*\*00001310\*\*00001311\*\*00001312\*\*00001328\*\*00001329\*\*00001357\*\*00001360\*\*00001409\*  
\*00001416\*\*00001439\*\*00001445\*\*00001453\*\*00001463\*\*00001465\*\*00001466\*\*00001467\*\*00001468\*\*00001475\*\*00001477\*  
\*00001478\*\*00001483\*\*00001484\*\*00001485\*\*00001490\*\*00001491\*\*00001492\*\*00001534\*\*00001563\*\*00001574\*\*00001577\*  
\*00001600\*\*00001607\*\*00001636\*\*00001637\*\*00001647\*\*00001667\*\*00001680\*\*00001694\*\*00001695\*\*00001701\*\*00001708\*  
\*00001727\*\*00001753\*\*00001754\*\*00001755\*\*00001756\*\*00001757\*\*00001758\*\*00001759\*\*00001762\*\*00001768\*\*00001770\*  
\*00001775\*\*00001779\*\*00001780\*\*00001795\*\*00001796\*\*00001805\*\*00001841\*\*00001850\*\*00001855\*\*00001858\*\*00001861\*  
\*00001862\*\*00001863\*\*00002220\*\*00002267\*\*00002353\*\*00002391\*\*00002430\*\*00002444\*\*00002496\*

TEMPVARNUM -- INTEGER -- DECLARED IN SEGMENT 29 AT 00002287  
\*00002291\* 00002292 00002293 00002317

TERM -- PROCEDURE -- DECLARED IN SEGMENT 2 AT 00001623  
00001669 00001683 00001715

TERMINAL -- DEFINE -- DECLARED IN SEGMENT 2 AT 00000211

00000470 00000471 00001427 00001881 00001883 00001939 00001941 00002030 00002210 00002273 00002594

TERMMESS -- FORMAT -- DECLARED IN SEGMENT 3 AT 00000190  
00003525

TEXT -- REAL ARRAY -- DECLARED IN SEGMENT 2 AT 00000126

\*00000322\*\*00000328\*\*00000339\*\*00000343\*\*00000344\* 00000488 \*00000493\*\*00000645\*\*00000647\*\*00001019\* 00001021  
\*00001180\*\*00001285\*\*00001328\*\*00001329\*\*00001330\*\*00001349\*\*00001428\*\*00001807\*\*00001880\*\*00001900\*\*00001901\*  
\*00001902\*\*00001903\*\*00001904\*\*00001905\*\*00001906\*\*00001907\*\*00001910\*\*00001911\*\*00001912\*\*00001918\*\*00001925\*  
\*00001926\*\*00001927\*\*00001929\*\*00001938\*\*00001948\*\*00001949\*\*00001950\*\*00001958\*\*00001959\*\*00001960\*\*00001964\*  
\*00001965\*\*00001966\*\*00001969\*\*00001973\*\*00001974\*\*00001975\*\*00001976\*\*00001977\*\*00001982\*\*00001988\*\*00001989\*  
\*00001992\*\*00001993\*\*00001994\*\*00001995\*\*00001996\*\*00001999\*\*00002001\*\*00002010\*\*00002011\*\*00002012\*\*00002014\*  
\*00002024\*\*00002025\*\*00002026\*\*00002027\*\*00002028\*\*00002030\*\*00002034\*\*00002035\*\*00002036\*\*00002037\*\*00002046\*  
\*00002065\*\*00002066\*\*00002074\*\*00002078\*\*00002088\*\*00002099\*\*00002106\*\*00002113\*\*00002128\*\*00002129\*\*00002137\*  
\*00002144\*\*00002151\*\*00002152\*\*00002165\*\*00002169\*\*00002218\*\*00002222\*\*00002245\*\*00002246\*\*00002250\*\*00002258\*  
\*00002266\*\*00002267\*\*00002275\*\*00002278\*\*00002290\*\*00002293\*\*00002295\*\*00002309\*\*00002317\*\*00002330\*\*00002333\*  
\*00002340\*\*00002343\*\*00002352\*\*00002353\*\*00002361\*\*00002374\*\*00002375\*\*00002379\*\*00002390\*\*00002391\*\*00002401\*  
\*00002429\*\*00002431\*\*00002432\*\*00002443\*\*00002445\*\*00002446\*\*00002454\*\*00002455\*\*00002456\*\*00002457\*\*00002459\*  
\*00002472\*\*00002543\*\*00002582\*\*00002956\*\*00002959\*\*00002960\*\*00002961\*\*00002965\*\*00002966\*\*00002967\*\*00002971\*  
\*00002974\*\*00002975\*\*00002976\*\*00002979\*\*00002980\*\*00002982\*\*00002986\*\*00002990\*\*00003011\*\*00003012\*\*00003013\*  
\*00003014\*\*00003015\*\*00003018\*\*00003022\*\*00003026\*\*00003027\*\*00003028\*\*00003029\*\*00003031\*\*00003034\*\*00003036\*  
\*00003040\*\*00003041\*\*00003042\*\*00003044\*\*00003045\*\*00003046\*\*00003047\*\*00003048\*\*00003051\*\*00003054\*\*00003056\*  
\*00003058\*\*00003059\*\*00003060\*\*00003064\*\*00003167\*\*00003171\*\*00003185\*\*00003188\*\*00003337\*\*00003338\*\*00003342\*  
\*00003347\*\*00003349\*\*00003352\*\*00003354\*\*00003360\*\*00003362\*\*00003365\*\*00003367\*\*00003374\*\*00003389\*\*00003395\*  
\*00003398\*\*00003409\*\*00003410\*\*00003411\*\*00003416\*\*00003418\*\*00003423\*\*00003439\* 00003536 00000559 00001240  
00002763 00002991 00003037

TEXTPNT -- POINTER -- DECLARED IN SEGMENT 2 AT 00000127

00000323 \*00000488\* 00000489 00000494 00000495 00000646 00000649 00001020 00001180 00001285 00001328  
00001329 00001330 00001349 00001428 00002245 00002258 00002290 00002343 00002374 00002390

TEXTPNT0 -- POINTER -- DECLARED IN SEGMENT 2 AT 00000127

00000328 00000339 00000343 00000344 \*00000488\* 00001807 00001880 00001900 00001901 00001902 00001903  
00001904 00001905 00001906 00001907 00001910 00001911 00001912 00001918 00001925 00001926 00001927  
00001929 00001938 00001948 00001949 00001950 00001958 00001959 00001960 00001964 00001965 00001966  
00001969 00001973 00001974 00001975 00001976 00001977 00001982 00001988 00001989 00001992 00001993  
00001994 00001995 00001996 00001999 00002001 00002010 00002011 00002012 00002014 00002024 00002025  
00002026 00002027 00002028 00002030 00002034 00002035 00002036 00002037 00002046 00002065 00002066  
00002074 00002078 00002088 00002099 00002106 00002113 00002128 00002129 00002137 00002144 00002151  
00002152 00002165 00002169 00002218 00002222 00002246 00002250 00002258 00002266 00002267 00002275  
00002278 00002293 00002295 00002309 00002317 00002330 00002333 00002340 00002343 00002352 00002353  
00002361 00002375 00002379 00002390 00002391 00002401 00002429 00002431 00002432 00002443 00002445  
00002446 00002454 00002455 00002456 00002457 00002459 00002472 00002543 00002582 00002956 00002959  
00002960 00002961 00002965 00002966 00002967 00002971 00002974 00002975 00002976 00002979 00002980  
00002982 00002986 00002990 00003011 00003012 00003013 00003014 00003015 00003018 00003022 00003026  
00003027 00003028 00003029 00003031 00003034 00003036 00003040 00003041 00003042 00003044 00003045

```

00003046 00003047 00003048 00003051 00003054 00003056 00003058 00003059 00003060 00003064 00003167
00003171 00003185 00003188 00003337 00003338 00003342 00003347 00003349 00003352 00003354 00003360
00003362 00003365 00003367 00003374 00003389 00003395 00003398 00003409 00003410 00003411 00003416
00003418 00003423 00003439
TEXTTYPE -- INTEGER -- DECLARED IN SEGMENT 2 AT 00000161
*00000558* 00000559 00000562
TFORM -- INTEGER -- DECLARED IN SEGMENT 2 AT 00000728
*00001554**00002312**00002660**00002664**00002896**00003202*
THENSY -- DEFINE -- DECLARED IN SEGMENT 2 AT 00000202
00000928 00002268 00002270 00002271
THISID -- ALPHA -- DECLARED IN SEGMENT 2 AT 00000418
00001179 *00001225* 00001227 00001286 00001287 *00001319* 00001322 00001323 00001327 *00001345* 00001348
00001350 *00001389* 00001392 00001393 00001395 00001397 00001398 00001399 00001402 *00001427* 00001540
00001545 *00001554**00001881**00001890* 00001893 00001894 *00001939**00002030**00002053* 00002056 00002058
00002065 00002066 *00002082* 00002085 00002086 *00002116* 00002119 00002121 00002128 00002129 *00002159*
00002204 00002205 00002228 *00002312**00002405* 00002409 00002410 00002414 *00002487* 00002490 *00002555*
00002558 00002559 00002560 00002561 *00002653* 00002656 00002658 *00002660**00002664**00002694* 00002697
00002698 *00002860* 00002861 *00002868* 00002871 00002873 *00002896**00003109* 00003112 00003114 *00003202*
*00003280* 00003281 00003282 *00003293* 00003294 00003297 00003301 00003302 *00003318* 00003321 00003323
00003326
THISINDEX -- INTEGER -- DECLARED IN SEGMENT 2 AT 00000417
*00000506* 00000507 *00000510* 00000511 *00000514* 00000515 *00000518* 00000519 *00000522* 00000523 *00000525*
*00000527**00000531* 00000533 *00000540**00000542**00000543**00000544**00000553**00000555**00000556**00000561*
*00000564* 00000565 *00000566* 00000567 00001180 *00001222* 00001225 00001285 *00001319* 00001328 00001329
00001330 *00001345* 00001349 *00001389**00001427**00001554**00001881**00001890**00001939**00002030**00002053*
00002063 *00002082**00002116* 00002127 *00002159* 00002227 00002228 *00002312**00002405* 00002408 *00002487*
*00002555**00002653**00002660**00002664**00002673* 00002674 *00002694**00002832* 00002833 *00002860**00002862*
*00002868**00002896**00003096* 00003097 *00003109**00003198**00003202**00003226**00003253* 00003254 *00003279*
00003280 *00003292* 00003293 00003296 00003299 *00003307**00003318*
THISLEVEL -- INTEGER -- DECLARED IN SEGMENT 2 AT 00000417
00001166 00001168 00001178 00001180 00001285 *00001319* 00001328 00001329 00001330 *00001345* 00001349
*00001389**00001427**00001554**00001881**00001890**00001939**00002030**00002053* 00002063 00002064 *00002082*
*00002116* 00002126 00002127 *00002159* 00002227 00002228 *00002312**00002405* 00002408 00002412 00002413
*00002487**00002555* 00002563 *00002653**00002660**00002664**00002694**00002860**00002868**00002896**00003109*
*00003202* 00003299 *00003318*
THISTAB -- INTEGER -- DECLARED IN SEGMENT 2 AT 00000417
*00001319**00001345**00001389* 00001404 *00001427**00001554**00001881**00001890**00001939**00002030**00002053*
*00002082**00002116**00002159**00002312**00002405**00002487**00002555**00002653**00002660**00002664**00002694*
*00002860**00002868**00002896**00003109**00003202**00003318*
TNAME -- ALPHA -- DECLARED IN SEGMENT 2 AT 00000418
*00000506**00000510**00000514**00000518**00000522**00000525**00000527**00000531**00000540**00000542**00000543*
*00000544**00000553**00000555**00000556**00000561**00000564**00000566**00001222**00001319**00001345**00001389*
*00001427**00001554**00001881**00001890**00001939**00002030**00002053**00002082**00002116**00002159**00002312*
*00002405**00002487**00002555**00002653**00002660**00002664**00002673**00002694**00002832**00002860**00002862*
*00002868**00002896**00003096**00003109**00003198**00003202**00003226**00003253**00003279**00003292**00003307*
*00003318*
TOPLEVEL -- INTEGER -- DECLARED IN SEGMENT 2 AT 00000172
00001319 00001345 00001389 00001427 00001554 00001881 00001890 00001939 00002030 00002053 00002082
00002116 00002159 00002312 00002405 00002482 00002487 00002497 *00002499* 00002512 *00002525* 00002555
00002653 00002660 00002664 00002694 00002860 00002868 00002896 00003109 *00003166* 00003202 00003318
*00003386*
TOPREC -- INTEGER -- DECLARED IN SEGMENT 40 AT 00003155
*00003163* 00003429 00003433
TOSY -- DEFINE -- DECLARED IN SEGMENT 2 AT 00000205
00000909 00002435 00002438 00002439
TSIZE -- INTEGER -- NAME PARAMETER -- DECLARED IN SEGMENT 2 AT 00002637
TSIZE -- INTEGER -- NAME PARAMETER -- DECLARED IN SEGMENT 2 AT 00002638

```



```

*00002658**00002660* 00002661 00002662 *00002664**00002680**00002692* 00002699 00002707 *00002746**00002764*
*00002784**00002799*
TSIZE -- INTEGER -- NAME PARAMETER -- DECLARED IN SEGMENT 36 AT 00002640
TSIZE -- INTEGER -- NAME PARAMETER -- DECLARED IN SEGMENT 36 AT 00002642
*00002644*
TTYPE -- INTEGER -- NAME PARAMETER -- DECLARED IN SEGMENT 2 AT 00002637
TTYPE -- INTEGER -- NAME PARAMETER -- DECLARED IN SEGMENT 2 AT 00002638
*00002658**00002660* 00002661 00002662 *00002664**00002680**00002691* 00002699 00002707 *00002746**00002762*
*00002782**00002797*
TTYPE -- INTEGER -- NAME PARAMETER -- DECLARED IN SEGMENT 36 AT 00002640
TTYPE -- INTEGER -- NAME PARAMETER -- DECLARED IN SEGMENT 36 AT 00002642
*00002644*
TT1 -- INTEGER -- DECLARED IN SEGMENT 2 AT 00000658
*00001197**00001355**00001597**00001605**00001658**00001721**00001790**00001793**00002075**00002138**00002223*
*00002316**00002433**00002447**00002901*
TT2 -- INTEGER -- DECLARED IN SEGMENT 2 AT 00000658
*00001197**00001355**00001597**00001605**00001658**00001721**00001790**00001793**00002075**00002138**00002223*
*00002316**00002433**00002447**00002901*
TX -- INTEGER -- DECLARED IN SEGMENT 2 AT 00001093
*00001807**00001900**00002088**00002165**00002218**00002222**00002267**00002353**00002391**00002431**00002445*
*00002582*
TX -- INTEGER -- DECLARED IN SEGMENT 36 AT 00002647
00002719 00002720 00002722 00002723 00002724 00002736 00002737 00002738 00002742 *00002744* 00002746
00002756 00002757 00002758 00002764 00002776 00002777 00002779 00002780 00002783 00002785 00002786
TX -- INTEGER -- DECLARED IN SEGMENT 37 AT 00002814
00002842 00002843 00002844
TX -- INTEGER -- DECLARED IN SEGMENT 40 AT 00003149
00003231 00003262 *00003339* 00003340 00003343 00003352 00003356 00003366
TYP -- INTEGER -- DECLARED IN SEGMENT 38 AT 00002942
*00002951* 00002980 00002982 *00002983* 00002984
TYPE -- DEFINE -- DECLARED IN SEGMENT 2 AT 00000059
00001179 00001227 00001327 00001350 00001399 00001427 00001540 00001554 00001881 00001939 00002030
00002058 00002065 00002066 00002121 00002128 00002129 00002228 00002312 00002414 00002658 00002660
00002664 00002668 00002698 00002844 00002873 00002878 00002896 00002951 00003114 00003202 00003239
00003282 00003323 00003326
TYPEDEFCL -- PROCEDURE -- DECLARED IN SEGMENT 2 AT 00002637
00002719 00002736 00002756 00002776 00002805 00002842 00003231 00003262
TYPEINDEX -- INTEGER -- DECLARED IN SEGMENT 2 AT 00000349
*00001614* 00001615 00001616 *00002660**00002664**00002668* 00002680 00002681 00002682 *00002691* 00002693
00002698 00002704 *00002727* 00002728 00002729 00002730 *00002762* 00002766 *00002782* 00002784 00002785
00002786 *00002797* 00002799 00002800
TYPERR -- PROCEDURE -- DECLARED IN SEGMENT 36 AT 00002640
00002661 00002662 00002699 00002707
TYPES -- DEFINE -- DECLARED IN SEGMENT 2 AT 00000065
00000506 00000510 00000514 00000518 00000522 00000560 00002656 00002697 00002861 00002871 00003112
00003232 00003281 00003321
TYPESY -- DEFINE -- DECLARED IN SEGMENT 2 AT 00000206
00000933 00003220 00003404
TYPETABTOP -- INTEGER -- DECLARED IN SEGMENT 40 AT 00003155
*00003161* 00003437
TYPETAB1 -- REAL ARRAY -- DECLARED IN SEGMENT 2 AT 00000081
*00000505**00000509**00000513**00000517**00000521**00000559* 00001175 00001191 00001197 00001204 00001205
00001206 00001217 00001234 00001238 00001264 00001301 00001335 00001355 00001409 00001417 00001427
00001439 00001449 00001464 00001475 00001483 00001490 00001534 00001554 00001596 00001597 00001604
00001605 *00001615* 00001632 00001657 00001658 00001686 00001691 00001720 00001721 00001748 00001789
00001790 00001792 00001793 00001847 00001852 00001855 00001858 00001861 00001881 00001882 00001896
00001939 00001940 00001968 00001981 00001987 00002030 00002032 00002058 00002062 00002075 00002090

```

```

00002110 00002121 00002125 00002138 00002163 00002166 00002167 00002213 00002219 00002223 00002296
00002312 00002316 00002415 00002433 00002447 00002494 00002500 00002658 *00002660**00002664**00002681*
*00002693**00002698* 00002722 *00002728* 00002737 00002738 00002741 *00002744* 00002758 *00002766* 00002779
*00002784**00002799* 00002843 00002873 00002896 00002901 00002951 00002984 00003038 00003049 00003119
00003202 *00003282* 00003323
TYPETAB2 -- REAL ARRAY -- DECLARED IN SEGMENT 2 AT 00000081
*00000505* 00001192 00001310 00001465 00001910 00002065 00002128 00002220 00002417 *00002660**00002664*
*00002682* 00002724 *00002729* 00002780 *00002785**00002800* 00002874 00002980 00003039 00003054
TYPETAB3 -- REAL ARRAY -- DECLARED IN SEGMENT 2 AT 00000081
*00000505**00000517**00000521* 00001192 00001311 00001466 00001911 00002066 00002129 00002220 00002417
*00002660**00002664**00002682* 00002724 *00002729* 00002780 *00002786**00002800* 00002874 00002982 00003039
00003056
TYPEX1 -- INTEGER -- DECLARED IN SEGMENT 2 AT 00002611
*00002660**00002664*
TYPEX2 -- INTEGER -- DECLARED IN SEGMENT 2 AT 00002611
*00002660**00002664*
TYPE1 -- INTEGER -- DECLARED IN SEGMENT 14 AT 00001625
*00001631* 00001632 00001653 00001655 00001657 00001658 00001659 00001664
TYPE1 -- INTEGER -- DECLARED IN SEGMENT 15 AT 00001674
*00001691* 00001716 00001718 00001720 00001721 00001722 00001724
TYPE1 -- INTEGER -- DECLARED IN SEGMENT 16 AT 00001734
*00001748* 00001785 00001786 00001789 00001790 00001793
T1 -- INTEGER -- DECLARED IN SEGMENT 2 AT 00000166
*00000956**00000957* 00000958 *00001252* 00001253 *00001614**00001615**00001807**00001900**00002088**00002218*
*00002222**00002267**00002353**00002391**00002431**00002445**00002582*
T1 -- INTEGER -- DECLARED IN SEGMENT 5 AT 00000468
*00000496* 00000497 00000498 *00000504* 00000505 *00000509**00000513**00000517**00000521**00000559*
T1 -- INTEGER -- DECLARED IN SEGMENT 25 AT 00002150
*00002165**00002166**00002167* 00002168 00002169
T1 -- INTEGER -- DECLARED IN SEGMENT 36 AT 00002647
*00002660**00002664**00002679**00002680* 00002681 *00002691**00002692* 00002693 *00002723**00002726* 00002728
*00002741**00002742**00002743* 00002744 *00002763**00002764**00002765* 00002766 *00002783**00002784**00002798*
*00002799*
T1 -- INTEGER -- DECLARED IN SEGMENT 37 AT 00002814
*00002873* 00002875
T1 -- INTEGER -- DECLARED IN SEGMENT 38 AT 00002942
*00002951* 00002963 00002972 00002983 *00002984* 00002985 00002991 00003037 00003038 00003039 00003049
00003054 00003056
T2 -- INTEGER -- DECLARED IN SEGMENT 2 AT 00000166
T2 -- INTEGER -- DECLARED IN SEGMENT 36 AT 00002647
*00002724* 00002725 00002726 00002729
T3 -- INTEGER -- DECLARED IN SEGMENT 2 AT 00000166
*00001170**00003114* 00003116 00003119 *00003125**00003126**00003127**00003132**00003134* 00003136
T3 -- INTEGER -- DECLARED IN SEGMENT 5 AT 00000468
*00000509**00000506* 00000507 *00000508**00000510* 00000511 *00000512**00000514* 00000515 *00000516**00000518*
00000519 *00000520**00000522* 00000523 *00000524* 00000525 *00000526* 00000527 *00000531**00000532* 00000533
*00000536* 00000540 00000542 00000543 00000544 *00000546* 00000553 00000555 00000556 *00000558**00000560*
00000561 *00000562**00000563* 00000565 00000567
T3 -- INTEGER -- DECLARED IN SEGMENT 36 AT 00002647
*00002668**00002674**00002724* 00002725 00002726 00002729 *00002741* 00002744
T3 -- INTEGER -- DECLARED IN SEGMENT 37 AT 00002814
*00002844**00002847* 00002848 *00002878* 00002879
T3 -- INTEGER -- DECLARED IN SEGMENT 40 AT 00003149
*00003203**00003210**00003211* 00003212 *00003232* 00003233 *00003263* 00003265 *00003308**00003309* 00003310
T4 -- INTEGER -- DECLARED IN SEGMENT 2 AT 00000166
*00001169* 00001170
T5 -- INTEGER -- DECLARED IN SEGMENT 2 AT 00000166
*00001169* 00001170

```

```

ULIM  -- INTEGER  -- DECLARED IN SEGMENT 10 AT 00001160
      *00001192* 00001196
ULIM  -- INTEGER  -- DECLARED IN SEGMENT 32 AT 00002397
      *00002417* 00002430 00002444
ULIM  -- INTEGER  -- DECLARED IN SEGMENT 37 AT 00002814
      *00002874**00002888* 00002900
UNPACK -- PROCEDURE -- DECLARED IN SEGMENT 2 AT 00002103
      00002145 00002580
UNTILSY -- DEFINE -- DECLARED IN SEGMENT 2 AT 00000204
      00000937 00002383 00002386 00002387 00002388
USER  -- ALPHA  -- DECLARED IN SEGMENT 2 AT 00000169
      00003031 *00003460* 00003461 00003555 00003562
VAL  -- REAL  -- DECLARED IN SEGMENT 12 AT 00001372
      *00001554*
VALUEPARAMS -- BOOLEAN -- DECLARED IN SEGMENT 40 AT 00003152
      *00003355* 00003359 *00003361* 00003365
VALX1 -- REAL  -- DECLARED IN SEGMENT 2 AT 00002610
      *00002660**00002664*
VALX2 -- REAL  -- DECLARED IN SEGMENT 2 AT 00002610
      *00002660**00002664*
VAR  -- DEFINE -- DECLARED IN SEGMENT 2 AT 00000061
      00000562 00001315 00001322 00001392 00001427 00001881 00001893 00001939 00002030 00002056 00002085
      00002119 00002204 00002409 00002490 00002558 00002844 00002878 00003087 00003117 00003263
VARIABLE -- PROCEDURE -- DECLARED IN SEGMENT 2 AT 00001158
      00001276 00001334 00001394 00001896 00002088 00002162 00002207 00002492 00003271
VARINDEX -- INTEGER -- DECLARED IN SEGMENT 2 AT 00000141
      *00003243* 00003245 00003250 *00003251**00003252* 00003254 00003264 00003270
VARLIST -- INTEGER ARRAY -- DECLARED IN SEGMENT 2 AT 00000140
      *00003254* 00003265 00003270
VARNUM -- INTEGER -- DECLARED IN SEGMENT 32 AT 00002397
      *00002408* 00002429 00002454 00002455
VARSY -- DEFINE -- DECLARED IN SEGMENT 2 AT 00000207
      00000920 00003085 00003087 00003241 00003247
VARTYPE -- INTEGER -- DECLARED IN SEGMENT 32 AT 00002397
      *00002414* 00002415 00002417 *00002418**00002433**00002447*
VERYFIRSTWITHSYM -- INTEGER -- DECLARED IN SEGMENT 34 AT 00002480
      *00002482* 00002525
V1  -- INTEGER -- DECLARED IN SEGMENT 4 AT 00000317
      *00000334* 00000335 00000336
V2  -- INTEGER -- DECLARED IN SEGMENT 4 AT 00000317
      *00000335* 00000337
WHILESTAT -- PROCEDURE -- DECLARED IN SEGMENT 2 AT 00002349
      00002365 00002589
WHILESY -- DEFINE -- DECLARED IN SEGMENT 2 AT 00000204
      00000936 00002589
WITHSTAT -- PROCEDURE -- DECLARED IN SEGMENT 2 AT 00002478
      00002526 00002592
WITHSY -- DEFINE -- DECLARED IN SEGMENT 2 AT 00000208
      00000930 00002592
WITHTAB -- REAL ARRAY -- DECLARED IN SEGMENT 2 AT 00000144
      00001170 *00002508*
WRITEALGOL -- PROCEDURE -- DECLARED IN SEGMENT 2 AT 00000358 -- FORWARD AT 00000227
      00000321 00000327 00000328 00000339 00000343 00000344 00000365 00001807 00001880 00001900 00001901
      00001902 00001903 00001904 00001905 00001906 00001907 00001910 00001911 00001912 00001918 00001925
      00001926 00001927 00001929 00001938 00001948 00001949 00001950 00001955 00001958 00001959 00001960
      00001964 00001965 00001966 00001969 00001973 00001974 00001975 00001976 00001977 00001982 00001988
      00001989 00001992 00001993 00001994 00001995 00001996 00001999 00002001 00002010 00002011 00002012

```

00002014	00002024	00002025	00002026	00002027	00002028	00002030	00002034	00002035	00002036	00002037
00002046	00002063	00002065	00002066	00002074	00002078	00002088	00002099	00002106	00002113	00002127
00002128	00002129	00002137	00002144	00002151	00002152	00002165	00002169	00002218	00002222	00002228
00002246	00002250	00002258	00002266	00002267	00002275	00002278	00002293	00002295	00002309	00002317
00002330	00002333	00002340	00002343	00002352	00002353	00002361	00002375	00002379	00002390	00002391
00002401	00002429	00002431	00002432	00002443	00002445	00002446	00002454	00002455	00002456	00002457
00002459	00002472	00002543	00002582	00002956	00002959	00002960	00002961	00002965	00002966	00002967
00002968	00002971	00002974	00002975	00002976	00002979	00002980	00002982	00002986	00002990	00003011
00003012	00003013	00003014	00003015	00003018	00003022	00003025	00003026	00003027	00003028	00003029
00003030	00003031	00003034	00003036	00003040	00003041	00003042	00003043	00003044	00003045	00003046
00003047	00003048	00003051	00003054	00003056	00003058	00003059	00003060	00003064	00003167	00003171
00003176	00003185	00003188	00003337	00003338	00003342	00003344	00003347	00003348	00003349	00003350
00003352	00003354	00003360	00003362	00003363	00003365	00003367	00003374	00003389	00003395	00003397
00003398	00003409	00003410	00003411	00003416	00003417	00003418	00003422	00003423	00003439	00003547

WRITEEXPR -- DEFINE -- DECLARED IN SEGMENT 2 AT 00001128  
00001807 00001900 00002088 00002165 00002218 00002222 00002267 00002353 00002391 00002431 00002445  
00002582

X -- REAL -- VALUE PARAMETER -- DECLARED IN SEGMENT 2 AT 00000313  
X -- REAL -- VALUE PARAMETER -- DECLARED IN SEGMENT 2 AT 00000314  
00000312 00000322 00000328 00000329 00000344

XLINE -- REAL ARRAY -- DECLARED IN SEGMENT 2 AT 00000128  
00000407 00000481 00000884 00001065 00003475 00003527

XLINEPNT -- POINTER -- DECLARED IN SEGMENT 2 AT 00000129  
\*00000481\* 00000483 00000884 00000958 00001065 00003475

XREFBLOCK -- DEFINE -- DECLARED IN SEGMENT 2 AT 00000576  
00000589 00000626 00000651

XREFCARD -- DEFINE -- DECLARED IN SEGMENT 2 AT 00000575  
00000635 00000650

XREFCOMPARE -- BOOLEAN PROCEDURE -- DECLARED IN SEGMENT 2 AT 00000601  
\*00000605\* 00000610 00003691

XREFFILE -- FILE -- DECLARED IN SEGMENT 2 AT 00000135  
00000591 00000621 00003691

XREFLINE -- REAL ARRAY -- DECLARED IN SEGMENT 2 AT 00000136  
00000619 00000630 00000632 00000641 00000643 00003689

XREFMAX -- PROCEDURE -- DECLARED IN SEGMENT 2 AT 00000594  
00000598 00003691

XREFOPTION -- BOOLEAN -- DECLARED IN SEGMENT 2 AT 00000157  
00000506 00000510 00000514 00000518 00000522 00000525 00000527 00000531 00000540 00000542 00000543  
00000544 00000553 00000555 00000556 00000561 00000564 00000566 \*00001047\* 00001222 00001319 00001345  
00001389 00001427 00001554 00001881 00001890 00001939 00002030 00002053 00002082 00002116 00002159  
00002312 00002405 00002487 00002555 00002653 00002660 00002664 00002673 00002694 00002832 00002860  
00002862 00002868 00002896 00003096 00003109 00003198 00003202 00003226 00003253 00003279 00003292  
00003307 00003318 00003687

XREFPNT -- POINTER -- DECLARED IN SEGMENT 2 AT 00000137  
\*00000632\*\*00000633\* 00000635 \*00000636\*\*00000643\* 00000644 00000646 00000649 00000650 \*00000651\*

Z -- REAL ARRAY -- DECLARED IN SEGMENT 42 AT 00003533  
\*00003544\*\*00003545\*\*00003553\*\*00003554\*\*00003555\*\*00003556\*\*00003557\*\*00003558\*\*00003559\*\*00003560\*\*00003561\*  
\*00003562\*\*00003563\*\*00003566\*\*00003567\*\*00003569\*

ZEROLAB -- BOOLEAN -- DECLARED IN SEGMENT 29 AT 00002288  
\*00002322\*

ZIPARRAY -- REAL ARRAY -- DECLARED IN SEGMENT 42 AT 00003533  
00003550 00003570

ZIPNUM -- PROCEDURE -- DECLARED IN SEGMENT 42 AT 00003542  
00003546 00003566

ZIPPNT -- POINTER -- DECLARED IN SEGMENT 42 AT 00003534  
\*00003544\*\*00003545\*\*00003550\* 00003551 \*00003553\*\*00003554\*\*00003555\*\*00003556\*\*00003557\*\*00003558\*\*00003559\*  
\*00003560\*\*00003561\*\*00003562\*\*00003563\*\*00003566\*\*00003567\*\*00003569\*

ZIPTEXT -- DEFINE -- DECLARED IN SEGMENT 42 AT 00003536

00003544 00003545 00003553 00003554 00003555 00003556 00003557 00003558 00003559 00003560 00003561  
00003562 00003563 00003566 00003567 00003569

CROSS REFERENCE STATISTICS

-----  
PHASE ONE - SORT 589 IDENTIFIERS  
0:25 ELAPSED TIME (MIN:SEC)  
0:12 PROCESSOR TIME  
0:20 I/O TIME

PHASE TWO - SORT 19683 REFERENCES  
3:37 ELAPSED TIME (MIN:SEC)  
3:12 PROCESSOR TIME  
0:32 I/O TIME

PHASE THREE - PRINT CROSS REFERENCE ( 1766 LINES)  
1:51 ELAPSED TIME (MIN:SEC)  
1:01 PROCESSOR TIME  
0:30 I/O TIME

LABEL 00000000LINE 00178109?COMPILE PASCAL/JUNK XALGOL XYNTAX

XALGOL /PASCAL

?EXECUTE 0/R

PACKET 225  
INPUT 5 CARDS FROM CRA  
TIME 1405  
DATE 78109 WEDNESDAY, 04/19/78

\*\*\* BURROUGHS B5700 DCMCP MARK XVI,0,178 AND INTRINSICS MARK XVI,0,132 \*\*\*

#NO MESSAGES TODAY

14:05:37 ?EXECUTE 0/R  
14:05:37 ?COMMON=1  
14:05:37 ?FILE LINE= LINE PRINT  
14:05:37 ?FILE S= PASCRUN/DISK  
14:05:38 ?END  
14:05:38 4:0/R= 3 BOJ 1405 11/22/76  
14:05:39 PBD0226 OUT 011 LINE:0/R= 3  
14:05:40 DKA IN SER PASCRUN DISK:0/R= 3  
14:05:49 DKA REL PASCRUN DISK:0/R= 3  
14:05:50 PBD0226 REL 011 LINE 476:0/R= 3  
14:05:50 0/R= 3 EOJ 1405  
14:05:51 FOR 0/R= 3: PROCESS= 6 SECS, IO= 11 SECS, OLAY= 0  
14:05:51 PKT#0225 REMOVED

```

$ CARD SEQXEQ RESET LIST% 00001
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX 00002
% 00003
% THE PASCAL RUN TIME=SYSTEM, % 00004
% ----- % 00005
% 00006
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX 00007
BEGIN% 00008
INTEGER V00167,V00168,V00169;% 00009
FILE INPUT "INPUT" (2,10);% 00010
FILE OUTPUT 1 (2,17);% 00011
% 00012
DEFINE PROCEDU #PROCEDURE#,% 00013
  FUNCTN #REAL PROCEDURE#,% 00014
  DOWNT0 #STEP -1 UNTIL#,% 00015
  UPT0 #STEP 1 UNTIL#,% 00016
  B #BOOLEAN#,% 00017
  F00603 #INPUT#,% 00018
  F00742 #OUTPUT#,% 00019
  LASTCH #[5:6]#,% 00020
  BUFSIZE #[13:8]#,% 00021
  BUFPNT #[21:8]#,% 00022
  EOF #[22:1]#,% 00023
  EOLN #[23:1]#,% 00024
  INP #[24:1]#,% 00025
  OUTP #[25:1]#,% 00026
  ENDFOUND#[26:1]#,% 00027
  MEMSIZE #10000#,% 00028
  MAXINT #5497558,3887#;% 00029
% 00030
ARRAY MEM[0:MEMSIZE DIV 1022,0:1022], TEXT,CHAR[0:0], TEMPTXT[0:19],% 00031
  V00603[0:9], V00742[0:16];% 00032
INTEGER MEMPNT,T,1,100603,100742;% 00033
POINTER CHARPNT,TEXTPNT;% 00034
LABEL TERMINATE;% 00035
FORMAT TERMMESS ("**** PROGRAM EXECUTION TERMINATED AT LINE ",I*,""),% 00036
  CHECKERR ("**** THE VALUE ",I*," IS NOT IN THE RANGE ",I*,".",",",% 00037
    I*,"."),% 00038
  ERRMARK (X*,"X"),% 00039
  CONCATERR("**** CONCAT ERROR: [",I*,";",I*,";",I*,";"]"),% 00040
  ILLEGALC("**** ILLEGAL CARRIAGE CONTROL CHARACTER:""",A1,"");% 00041
SWITCH FORMAT ERRMESS I=% 00042
  (%),% 00043
  ("**** NO READING WHILE EOF IS TRUE,"), %1 00044
  ("**** NO WRITING WHILE EOF IS FALSE,"), %2 00045
  ("**** ILLEGAL CHARACTER,"), %3 00046
  ("**** OVERFLOW ERROR,"), %4 00047
  ("**** NO RESET/REWRITE ON INPUT/OUTPUT,"), %5 00048
  ("**** LINE IMAGE OVERFLOW,"); %6 00049
MONITOR EXPOVRI=REALOVERFLOW;% 00050
% 00051
INTEGER PROCEDURE NUMDIGITS(N);% 00052
VALUE N; INTEGER N;% 00053

```

```

NUMDIGITS:=IF N<0 THEN 1+NUMDIGITS(-N) ELSE%
                IF N>9 THEN 1+NUMDIGITS(N DIV 10) ELSE 1;%
%
PROCEDURE RUNERR(ERRNUM,LINENUM);          %*** RUN TIME ERROR ***
VALUE ERRNUM,LINENUM;%
INTEGER ERRNUM,LINENUM;%
BEGIN%
    WRITE(OUTPUT,ERRMESS(ERRNUM));%
    WRITE(OUTPUT,TERMMESS,NUMDIGITS(LINENUM),LINENUM);%
    GO TO TERMINATE;%
END OF RUNERR;%
%
INTEGER PROCEDURE CHECK(VAL,LIM1,LIM2,LINENUM);%
VALUE VAL,LIM1,LIM2,LINENUM;%
INTEGER VAL,LIM1,LIM2,LINENUM;%
BEGIN%
    IF VAL<LIM1 OR VAL>LIM2 THEN%
        BEGIN WRITE(OUTPUT,CHECKERR,NUMDIGITS(VAL),VAL,NUMDIGITS(LIM1),%
                    LIM1,NUMDIGITS(LIM2),LIM2);%
                RUNERR(4,LINENUM);%
        END;%
    CHECK:=VAL;%
END OF CHECK;%
%
ALPHA PROCEDURE CURDAT;%
CURDAT:=" "&TIME(5)[41:35:36];%
%
ALPHA PROCEDURE WEEKDA;%
WEEKDA:=TIME(6)&" "[41:5:6];%
%
INTEGER PROCEDURE TRUNC(X,LINENUM);%
VALUE X,LINENUM;%
REAL X; INTEGER LINENUM;%
BEGIN%
    IF ABS(X)>MAXINT THEN RUNERR(4,LINENUM);%
    TRUNC:=IF X<0 THEN -ENTIER(-X) ELSE ENTIER(X);%
END OF TRUNC;%
%
INTEGER PROCEDURE ROUND(X,LINENUM);%
VALUE X,LINENUM;%
REAL X; INTEGER LINENUM;%
BEGIN%
    IF ABS(X)>MAXINT THEN RUNERR(4,LINENUM);%
    ROUND:=X;%
END OF ROUND;%
%
BOOLEAN PROCEDURE ODD(N);%
VALUE N; INTEGER N;%
ODD:=N MOD 2 = 1;%
%
REAL PROCEDURE SQR(X,LINENUM);%
VALUE X,LINENUM;%
REAL X; INTEGER LINENUM;%
BEGIN%
    IF ABS(X)>2.0769187034 THEN RUNERR(4,LINENUM);%
    SQR:=X*X;%
END OF SQR;%
%

```

```

00054
00055
00056
00057
00058
00059
00060
00061
00062
00063
00064
00065
00066
00067
00068
00069
00070
00071
00072
00073
00074
00075
00076
00077
00078
00079
00080
00081
00082
00083
00084
00085
00086
00087
00088
00089
00090
00091
00092
00093
00094
00095
00096
00097
00098
00099
00100
00101
00102
00103
00104
00105
00106
00107
00108
00109
00110
00111

```



```

BOOLEAN PROCEDURE INCL1(A,B);          **** IS THE SET "A" INCLUDED      00112
VALUE A,B; REAL A,B;                 **** IN THE SET "B",          00113
INCL1:=REAL(BOOLEAN(A) AND NOT BOOLEAN(B))=0;% 00114
% 00115
BOOLEAN PROCEDURE INCL2(A,B);          **** IS THE SET "B" INCLUDED      00116
VALUE A,B; REAL A,B;                 **** IN THE SET "A",          00117
INCL2:=REAL(BOOLEAN(B) AND NOT BOOLEAN(A))=0;% 00118
% 00119
BOOLEAN PROCEDURE INTST(A,B);          **** IS THE VALUE "A" AN ELEMENT 00120
VALUE A,B; REAL A,B;                 **** IN THE SET "B",          00121
INTST:=IF A<0 OR A>38 THEN FALSE ELSE 0&B[0:38=A:1]=1;% 00122
% 00123
PROCEDURE NEW(P,SIZE);%                00124
VALUE SIZE; REAL P; INTEGER SIZE;%    00125
BEGIN%                                  00126
  PI=IF MEMPNT+SIZE>MEMSIZE THEN 0 ELSE MEMPNT;% 00127
  MEMPNT:=MEMPNT+SIZE;%                00128
END OF NEW;%                             00129
% 00130
PROCEDURE DISPOSE(P,SIZE);%            00131
VALUE SIZE; REAL P; INTEGER SIZE;%    00132
BEGIN%                                  00133
END OF DISPOSE;%                         00134
% 00135
PROCEDURE PACK(A,LLIM,ULIM,I,Z,LINENUM);% 00136
VALUE LLIM,ULIM,I,LINENUM;%           00137
ARRAY A[*]; ALPHA Z;%                 00138
INTEGER LLIM,ULIM,I,LINENUM;%         00139
BEGIN%                                  00140
  Z:=0;%                                 00141
  FOR T1:=0 STEP 1 UNTIL 6 DO%          00142
    Z:=A[CHECK(I+T1,LLIM,ULIM,LINENUM)] & Z [41:35:36];% 00143
  END;%                                  00144
% 00145
PROCEDURE UNPACK(Z,A,LLIM,ULIM,I,LINENUM);% 00146
VALUE Z,LLIM,ULIM,I,LINENUM;%         00147
ARRAY A[*]; ALPHA Z;%                 00148
INTEGER LLIM,ULIM,I,LINENUM;%         00149
FOR T1:=0 STEP 1 UNTIL 6 DO%           00150
  A[CHECK(I+T1,LLIM,ULIM,LINENUM)]= 0 & Z [5:4:6*T1:6];% 00151
% 00152
REAL PROCEDURE @CONCAT(A,B,AS,BS,N,LINENUM);% 00153
VALUE A,B,AS,BS,N,LINENUM;%           00154
REAL A,B; INTEGER AS,BS,N,LINENUM;%   00155
BEGIN%                                  00156
  IF AS<1 OR BS<1 OR N<0 OR AS+N>48 OR BS+N>48 THEN% 00157
    BEGIN%                                00158
      WRITE(OUTPUT,CONCATERR,NUMDIGITS(AS),AS,NUMDIGITS(BS),% 00159
        BS,NUMDIGITS(N),N);%              00160
      RUNERR(0,LINENUM);%                 00161
    END;%                                  00162
    CONCAT:=A & B [47-AS:47-BS:N];%       00163
  END OF CONCAT;%                         00164
% 00165
BOOLEAN PROCEDURE BIT(N,LINENUM);      **** SET BIT NO "N" IN A WORD, 00166
VALUE N,LINENUM; INTEGER N,LINENUM;%  00167
BIT:=BOOLEAN(0 & 1 [38-CHECK(N,0,38,LINENUM):0:1]);% 00168

```

```

%
BOOLEAN PROCEDURE BITS(N1,N2,LINENUM);  %*** SET BITS "N1",,"N2",
VALUE N1,N2,LINENUM;%
INTEGER N1,N2,LINENUM;%
BITS:=BOOLEAN(0 & 3"77777777777777" [38"CHECK(N1,0,38,LINENUM):38;%
CHECK(N2,0,38,LINENUM)-N1+1]);%
%
PROCEDURE RLINE(F,BUF,INFO);%
FILE F; ARRAY BUF[0]; INTEGER INFO;%
BEGIN%
  LABEL ENDFILE;%
  INFO,EOLN:=0;  INFO,BUFPNT:=1;%
  READ(F,999,BUF[*]) [ENDFILE];%
  REPLACE CHARPNT BY POINTER(BUF[*]) FOR 1;%
  INFO,LASTCH:=CHAR[0];%
  IF FALSE THEN%
  BEGIN ENDFILE; INFO,ENDFOUND:=1;%
  END;%
END OF RLINE;%
%
REAL PROCEDURE PREAD(F,BUF,INFO,MODE,LINENUM);%
VALUE MODE,LINENUM;%
FILE F; ARRAY BUF[0];%
INTEGER INFO,MODE,LINENUM;%
BEGIN%
  DEFINE GETCHAR=%
  BEGIN%
    IF BOOLEAN(INFO,EOLN) THEN%
    BEGIN%
      RLINE(F,BUF,INFO);  CH:=INFO,LASTCH;%
    END ELSE%
    IF INFO,BUFPNT=INFO,BUFSIZE THEN%
    BEGIN CH:=" "; INFO,EOLN:=1 END ELSE%
    BEGIN%
      REPLACE CHARPNT BY POINTER(BUF[*])+INFO,BUFPNT FOR 1;%
      CH:=CHAR[0];  INFO,BUFPNT:=INFO,BUFPNT+1;%
    END END OF GETCHAR;%
%
  DEFINE READERR(ERRNUM)=%
  BEGIN%
    WRITE(OUTPUT,999,BUF[*]);%
    WRITE(OUTPUT,ERRMARK,INFO,BUFPNT-1);%
    RUNERR(ERRNUM,LINENUM);%
  END OF READERR;%
%
  REAL RES;  ALPHA CH;%
  BOOLEAN NEGATIVE,NEGEXP;  INTEGER POWER,EXP;%
  LABEL OVERFLOW,RETURN;%
%
  IF BOOLEAN(INFO,EOF) THEN RUNERR(1,LINENUM);%
  IF BOOLEAN(INFO,ENDFOUND) THEN%
  BEGIN%
    INFO,EOF:=1;  PREAD:=0;%
    GO TO RETURN;%
  END;%
  IF MODE=1 THEN  %*** MODE = CHAR ***
  BEGIN%
    PREAD:=INFO,LASTCH;  GETCHAR;  INFO,LASTCH:=CH;%

```

```

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
00225
00226

```

```

END ELSE%
BEGIN                                     %*** MODE = REAL/INTEGER ***
  CH:=INFO, LASTCH;%
  WHILE CH=" " AND NOT BOOLEAN(INFO, ENDFOUND) DO GETCHAR;%
  IF BOOLEAN(INFO, ENDFOUND) THEN%
  BEGIN%
    INFO, EOF:=1; PREAD:=0;%
    GO TO RETURN;%
  END;%
  IF CH="+" OR CH="-" THEN BEGIN NEGATIVE:=CH="-"; GETCHAR END;%
  IF CH>9 THEN READERR(3);%
  RES:=CH; GETCHAR;%
  WHILE CH<9 DO BEGIN RES:=10*RES+CH; GETCHAR END;%
  IF MODE=3 THEN % MODE = REAL,
  BEGIN%
    IF CH="." THEN%
    BEGIN%
      GETCHAR; IF CH>9 THEN READERR(3);%
      WHILE CH<9 DO BEGIN RES:=10*RES+CH; POWER:=POWER-1; GETCHAR END;%
    END;%
    IF CH="E" THEN%
    BEGIN%
      GETCHAR;%
      IF CH="+" OR CH="-" THEN BEGIN NEGEXP:=CH="-"; GETCHAR END;%
      IF CH>9 THEN READERR(3);%
      WHILE CH<9 DO BEGIN EXP:=10*EXP+CH; GETCHAR END;%
      IF NEGEXP THEN EXP:=-EXP;%
    END;%
    POWER:=POWER+EXP;%
    REALOVERFLOW:=OVERFLOW; RES:=RES*10*POWER;%
    IF FALSE THEN OVERFLOW; READERR(4);%
    REALOVERFLOW:=0;%
  END ELSE IF RES>MAXINT THEN READERR(4);%
  PREAD:=IF NEGATIVE THEN -RES ELSE RES;%
  INFO, LASTCH:=CH;%
END;%
RETURN;%
END OF PREAD;%
%
%
PROCEDURE WLINE(F, BUF, INFO); %*** PRINT A LINE, ***
FILE F; ARRAY BUF[0]; INTEGER INFO;%
BEGIN%
  ALPHA CC;%
  IF BOOLEAN(INFO, OUTP) THEN%
  BEGIN%
    REPLACE CHARPNT BY POINTER(BUF[*]) FOR 1; CC:=CHAR[0];%
    REPLACE POINTER(BUF[*]) BY " ";%
    IF CC=" " THEN WRITE(OUTPUT, 999, BUF[*]) ELSE%
    IF CC="+" THEN WRITE(OUTPUT[N0], 999, BUF[*]) ELSE%
    BEGIN%
      IF CC="0" THEN WRITE(OUTPUT) ELSE%
      IF CC="=" THEN WRITE(OUTPUT[DBL]) ELSE%
      IF CC="1" THEN WRITE(OUTPUT[PAGE]) ELSE%
      WRITE(OUTPUT, ILLEGALCC, CC);%
      WRITE(OUTPUT, 999, BUF[*]);%
    END;%
  END;%

```

```

00227
00228
00229
00230
00231
00232
00233
00234
00235
00236
00237
00238
00239
00240
00241
00242
00243
00244
00245
00246
00247
00248
00249
00250
00251
00252
00253
00254
00255
00256
00257
00258
00259
00260
00261
00262
00263
00264
00265
00266
00267
00268
00269
00270
00271
00272
00273
00274
00275
00276
00277
00278
00279
00280
00281
00282
00283

```

```

END ELSE WRITE(F,999,BUF[*]);%
REPLACE POINTER(BUF[*]) BY " " FOR INFO,BUFSIZE;%
INFO,BUFPNT:=0;%
END OF WLINE;%
%
%
PROCEDURE CHFIL(F);%
FILE F;%
BEGIN%
  ARRAY A[0:6];%
  SEARCH(F,A[*]);%
  IF A[0]#-1 THEN%
    BEGIN%
      F,AREAS := 20;%
      F,AREASIZE := 300;%
    END;%
  END OF CHFIL;%
%
%
PROCEDURE WALFA(F,BUF,INFO,A1,A2,A3,A4,A5,A6,A7,A8,A9,A10,A11,A12,ALENG,
LINENUM);%
VALUE A1,A2,A3,A4,A5,A6,A7,A8,A9,A10,A11,A12,ALENG,LINENUM;%
FILE F; ARRAY BUF[0]; INTEGER INFO,ALENG,LINENUM;%
ALPHA A1,A2,A3,A4,A5,A6,A7,A8,A9,A10,A11,A12;%
BEGIN%
  ALPHA A; POINTER PNT;%
  LABEL EXIT;%
  IF NOT BOOLEAN(INFO,EOF) THEN RUNERR(2,LINENUM);%
  IF INFO,BUFPNT+ALENG>INFO,BUFSIZE THEN WLINE(F,BUF,INFO);%
  PNT:=POINTER(BUF[*])+INFO,BUFPNT;%
  INFO,BUFPNT:=INFO,BUFPNT+ALENG;%
  FOR A:=A1,A2,A3,A4,A5,A6,A7,A8,A9,A10,A11,A12 DO%
    BEGIN%
      TEXT[0]:=A;%
      REPLACE PNT:PNT BY TEXTPNT FOR MIN(ALENG,7);%
      ALENG:=ALENG-7; IF ALENG<0 THEN GO TO EXIT;%
    END;%
  EXIT;%
END OF WALFA;%
%
%
PROCEDURE PWRITE(F,BUF,INFO,E,EMODE,M,N,LINENUM);%
VALUE E,EMODE,M,N,LINENUM;%
FILE F; ARRAY BUF[0]; REAL E;%
INTEGER INFO,EMODE,M,N,LINENUM;%
BEGIN%
  INTEGER NCHARS,NEXP,I; POINTER CPNT;%
  DEFINE PUTCHAR(C)= % PUTS A CHARACTER INTO TEMPTXT
  BEGIN CHAR[0]:=C; NCHARS:=NCHARS+1;%
  REPLACE CPNT:CPNT BY CHARPNT FOR 1;%
  END#;%
%
  PROCEDURE PUTINT(N); % PUTS AN INTEGER INTO TEMPTXT
  VALUE N; INTEGER N; % WITH ZERO SUPPRESSION.
  IF N<9 THEN PUTCHAR(N) ELSE%
  BEGIN PUTINT(N DIV 10); PUTCHAR(ENTIER(N MOD 10)) END;%
%
%

```

```

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
00339
00340

```

```

CPNT:=POINTER(TEMPTEXT[*]);% 00341
IF NOT BOOLEAN(INFO,EOF) THEN RUNERR(2,LINENUM);% 00342
IF EMODE=1 THEN %*** MODE = INTEGER *** 00343
BEGIN% 00344
  IF E<0 THEN BEGIN PUTCHAR("-"); E:=-E END;% 00345
  PUTINT(E);% 00346
END ELSE% 00347
IF EMODE=2 THEN %*** MODE = REAL *** 00348
BEGIN% 00349
  PUTCHAR(" ");% 00350
  IF E<0 THEN BEGIN PUTCHAR("-"); E:=-E END;% 00351
  IF E>MAXINT OR N<0 THEN % FLOATING-POINT, 00352
  BEGIN% 00353
    IF E>0 THEN% 00354
    BEGIN% 00355
      WHILE E<1 DO BEGIN NEXP:=NEXP-1; E:=10*E END;% 00356
      WHILE E<=10 DO BEGIN NEXP:=NEXP+1; E:=E/10 END;% 00357
    END;% 00358
    I:=MAX(M-8,1);% 00359
    E:=E+0,5*10*(-I);% 00360
    IF E GEQ 10 THEN BEGIN NEXP:=NEXP+1; E:=E/10 END;% 00361
    PUTCHAR(ENTIER(E)); E:=E-ENTIER(E); PUTCHAR(",");% 00362
    DO BEGIN% 00363
      E:=10*E; PUTCHAR(ENTIER(E));% 00364
      E:=E-ENTIER(E); I:=I-1;% 00365
    END UNTIL I<=0;% 00366
    PUTCHAR("E");% 00367
    IF NEXP<0 THEN BEGIN PUTCHAR("-"); NEXP:=NEXP END;% 00368
    ELSE PUTCHAR("+");% 00369
    PUTCHAR(NEXP DIV 10); PUTCHAR(ENTIER(NEXP MOD 10));% 00370
  END ELSE% 00371
  BEGIN % FIXED-POINT, 00372
    E:=E+0,5*10*(-N);% 00373
    PUTINT(ENTIER(E)); PUTCHAR(","); E:=E-ENTIER(E);% 00374
    IF N>150 THEN RUNERR(6,LINENUM);% 00375
    FOR I:=1 STEP 1 UNTIL N DO% 00376
    BEGIN E:=10*E; PUTCHAR(ENTIER(E));% 00377
      E:=E-ENTIER(E);% 00378
    END END END ELSE% 00379
  IF EMODE=3 THEN %*** MODE = BOOLEAN *** 00380
  BEGIN% 00381
    IF E<0,5 THEN REPLACE CPNT BY "FALSE" ELSE REPLACE CPNT BY "TRUE"; 00382
    NCHARS:=IF E<0,5 THEN 5 ELSE 4;% 00383
  END ELSE% 00384
  IF EMODE=5 THEN %*** MODE = ALFA *** 00385
  BEGIN% 00386
    TEXT[0]:=E; NCHARS:=MIN(M,7);% 00387
    REPLACE CPNT:CPNT BY TEXTPNT FOR 7;% 00388
  END ELSE% 00389
  BEGIN %*** MODE = CHAR *** 00390
    PUTCHAR(E);% 00391
  END;% 00392
  IF NCHARS>M THEN M:=NCHARS;% 00393
  IF INFO,BUFPNT+M>INFO,BUFSIZE THEN WLINE(F,BUF,INFO);% 00394
  IF M>INFO,BUFSIZE THEN RUNERR(6,LINENUM);% 00395
  REPLACE POINTER(BUF[*])+(INFO,BUFPNT+M-NCHARS) BY% 00396
  POINTER(TEMPTEXT[*]) FOR NCHARS;% 00397

```

INFO,BUFPNT:=INFO,BUFPNT+M;%	00398
END OF PWRITE;%	00399
%	00400
%	00401
PROCEDURE PUT(F,BUF,INFO,LINENUM);%	00402
VALUE LINENUM;%	00403
FILE F; ARRAY BUF[*];%	00404
INTEGER INFO,LINENUM;%	00405
BEGIN%	00406
IF INFO,BUFSIZE=0 THEN%	00407
BEGIN%	00408
IF NOT BOOLEAN(INFO,EOF) THEN RUNERR(2,LINENUM);%	00409
WRITE(F,1023,BUF[*]);%	00410
END ELSE PWRITE(F,BUF,INFO,INFO,LASTCH,4,1,1,LINENUM);%	00411
END OF PUT;%	00412
%	00413
%	00414
PROCEDURE GET(F,BUF,INFO,LINENUM);%	00415
VALUE LINENUM;%	00416
FILE F; ARRAY BUF[*];%	00417
INTEGER INFO,LINENUM;%	00418
BEGIN%	00419
ALPHA X; LABEL ENDFILE;%	00420
IF INFO,BUFSIZE=0 THEN%	00421
BEGIN%	00422
IF BOOLEAN(INFO,EOF) THEN RUNERR(1,LINENUM);%	00423
READ(F,1023,BUF[*]) [ENDFILE];%	00424
IF FALSE THEN ENDFILE: INFO,EOF:=1;%	00425
END ELSE X:=PREAD(F,BUF,INFO,1,LINENUM);%	00426
END OF GET;%	00427
%	00428
%	00429
PROCEDURE PPAGE(F,BUF,INFO,LINENUM);%	00430
VALUE LINENUM;%	00431
FILE F; ARRAY BUF[*];%	00432
INTEGER INFO,LINENUM;%	00433
BEGIN%	00434
IF NOT BOOLEAN(INFO,EOF) THEN RUNERR(2,LINENUM);%	00435
WRITE(F[PAGE]);%	00436
END OF PPAGE;%	00437
%	00438
%	00439
PROCEDURE RESET(F,BUF,INFO,LINENUM);%	00440
VALUE LINENUM;%	00441
FILE F; ARRAY BUF[*];%	00442
INTEGER INFO,LINENUM;%	00443
BEGIN%	00444
IF BOOLEAN(INFO,INP) OR BOOLEAN(INFO,OUTP) THEN RUNERR(5,LINENUM);%	00445
REWIND(F); INFO,EOF:=0; INFO,EOLN:=0; INFO,BUFPNT:=0;%	00446
INFO,ENDFOUND:=0;%	00447
IF INFO,BUFSIZE=0 THEN GET(F,BUF,INFO,LINENUM);%	00448
ELSE RLINE(F,BUF,INFO);%	00449
END OF RESET;%	00450
%	00451
PROCEDURE REWRITE(F,BUF,INFO,LINENUM);%	00452
VALUE LINENUM;%	00453
FILE F; ARRAY BUF[*];%	00454
INTEGER INFO,LINENUM;%	00455

BEGIN%	00456
IF BOOLEAN(INFO,INP) OR BOOLEAN(INFO,OUTP) THEN RUNERR(5,LINENUM);%	00457
REWIND(F); INFO,EOF:=1; INFO,BUFPNT:=0; INFO,ENDFOUND:=0;%	00458
IF INFO,BUFSIZE>0 THEN%	00459
REPLACE POINTER(BUF[*]) BY " " FOR INFO,BUFSIZE;%	00460
END OF REWRITE;%	00461
%	00462
%	00463
PROCEDURE INIT(INPUTDECL);%	00464
VALUE INPUTDECL;%	00465
BOOLEAN INPUTDECL;%	00466
BEGIN%	00467
MEMPNT:=1;%	00468
CHARPNT:=POINTER(CHAR[*])+7; TEXTPNT:=POINTER(TEXT[*])+1;%	00469
T:=0; T,BUFSIZE:=80; T,BUFPNT:=80; T,EOLN:=1; T,INP:=1;%	00470
I00603:=T; IF INPUTDECL THEN RLINE(INPUT,V00603,I00603);%	00471
T:=0; T,BUFSIZE:=132; T,EOLN:=1; T,OUTP:=1; T,EOF:=1;%	00472
I00742:=T;%	00473
REPLACE POINTER(V00742[*]) BY " " FOR 17 WORDS;%	00474
END OF INIT;%	00475

LABEL 00000000LINE 00178109?EXECUTE 0/R

0 /R

?EXECUTE PASCAL/PASCAL

PACKET 223  
INPUT 205 CARDS FROM CRA  
TIME 1355  
DATE 78109 WEDNESDAY, 04/19/78

\*\*\* BURROUGHS B5700 DCMCP MARK XVI,0,178 AND INTRINSICS MARK XVI,0,132 \*\*\*

#NO MESSAGES TODAY

13155155 ?EXECUTE PASCAL/PASCAL  
13155155 ?DATA029 SOURCE  
13155156 5:PASCAL/PASCAL=01 SCHEDULED 1355, NEEDS 27264  
13156153 1XS  
13156153 5:PASCAL/PASCAL= 1 BOJ 1356 04/19/78  
13156156 DKA IN SER PASC RUN DISK:PASCAL/PASCAL= 1  
13156156 DKA OUT SER PASC325 U000000:PASCAL/PASCAL= 1  
13157103 DKA REL PASC RUN DISK:PASCAL/PASCAL= 1  
13157110 PBDO224 OUT 011 LINES:PASCAL/PASCAL= 1  
13157112 CDB IN SOURCE:PASCAL/PASCAL= 1  
13158105 +OPERATOR KEYED IN: 1TL  
13158105 TIME LIMITS FOR PASCAL/PASCAL= 1 ARE: PRT=NO LIMIT; IOT=NO LIMIT,  
13158120 +OPERATOR KEYED IN: 1CT30:45  
13158121 INV KBD 1CT30:45  
13158131 +OPERATOR KEYED IN: 1CT 30:45  
13158131 INV KBD 1CT 30:45  
13158140 +OPERATOR KEYED IN: 1CT30,45  
13158140 PRT AND IOT ESTIMATES CHANGED FOR PASCAL/PASCAL= 1  
13158140 TIME LIMITS FOR PASCAL/PASCAL= 1 ARE: PRT= 0:30; IOT= 0:45,  
13159108 +OPERATOR KEYED IN: 1CT200,200  
13159109 PRT AND IOT ESTIMATES CHANGED FOR PASCAL/PASCAL= 1  
13159109 TIME LIMITS FOR PASCAL/PASCAL= 1 ARE: PRT= 3:20; IOT= 3:20,  
13159126 +OPERATOR KEYED IN: 1TI  
13159127 TIME FOR PASCAL/PASCAL= 1 IS: CP= 15, IO= 56 IN 2134  
13159158 +OPERATOR KEYED IN: 1CT300  
13159159 PRT ESTIMATE CHANGED FOR PASCAL/PASCAL= 1  
13159159 TIME LIMITS FOR PASCAL/PASCAL= 1 ARE: PRT= 5:00; IOT= 3:20,  
14100115 +OPERATOR KEYED IN: 1CT,500  
14100116 IOT ESTIMATE CHANGED FOR PASCAL/PASCAL= 1  
14100117 TIME LIMITS FOR PASCAL/PASCAL= 1 ARE: PRT= 5:00; IOT= 8:20,  
14101103 DKA LOK PASC325 U000000:PASCAL/PASCAL= 1  
14101104 CC USER=U000000;  
14101104 COMPIL INSANE/U000000 XALGOL;  
14101104 XALGOL FILE CARD=PASC325/U000000 SERIAL;  
14101104 END  
14101106 5:IXALGOL/INSANE=01 SCHEDULED 1401, NEEDS 14976  
14101107 PBDO224 REL 011 LINES 213:PASCAL/PASCAL= 1  
14101107 CDB REL SOURCE:PASCAL/PASCAL= 1  
14101107 ?DATA029 INPUT  
14101108 PASCAL/PASCAL= 1 EOJ 1401  
14101109 FOR PASCAL/PASCAL= 1; PROCESS= 40 SECS, IO= 104 SECS, OLAY= 47



```

14101109      51XALGOL/INSANE= 1 BOJ 1401 04/17/78
14101111      DKA IN SER PASC325 U000000;XALGOL/INSANE= 1
14101114      DKA OUT RDM INSANE U000000;XALGOL/INSANE= 1
14101138      +OPERATOR KEYED IN: 1TL
14101138      TIME LIMITS FOR XALGOL/INSANE= 1 ARE: PRT=NO LIMIT; IOT=NO LIMIT.
14101158      +OPERATOR KEYED IN: 1CT300,300
14101158      PRT AND IOT ESTIMATES CHANGED FOR XALGOL/INSANE= 1
14101158      TIME LIMITS FOR XALGOL/INSANE= 1 ARE: PRT= 5:00; IOT= 5:00.
14101158      DKA LOK INSANE U000000;XALGOL/INSANE= 1
14101158      DKA REL PASC325 U000000;XALGOL/INSANE= 1
14101159      XALGOL/INSANE= 1 EOJ 1401
14101159      FOR XALGOL/INSANE= 1: PROCESS= 38 SECS, IO= 18 SECS, OLAY= 1
14102100      51INSANE/U000000= 1 BOJ 1401 04/19/78
14102101      CDB IN INPUT;INSANE/U000000= 1
14102101      PBD0224 OUT 021 OUTPUT;INSANE/U000000= 1
14102120      +OPERATOR KEYED IN: 1TL
14102120      TIME LIMITS FOR INSANE/U000000= 1 ARE: PRT=NO LIMIT; IOT=NO LIMIT.
14102134      +OPERATOR KEYED IN: 1CT200,200
14102134      PRT AND IOT ESTIMATES CHANGED FOR INSANE/U000000= 1
14102134      TIME LIMITS FOR INSANE/U000000= 1 ARE: PRT= 3:20; IOT= 3:20.
14102150      +OPERATOR KEYED IN: 1TI
14102151      TIME FOR INSANE/U000000= 1 IS: CP= 44, IO= 14 IN 51
14103133      +OPERATOR KEYED IN: 1TI
14103133      TIME FOR INSANE/U000000= 1 IS: CP= 1:23, IO= 21 IN 1:33
14104149      +OPERATOR KEYED IN: 1TI
14104149      TIME FOR INSANE/U000000= 1 IS: CP= 2:32, IO= 35 IN 2:49
14105152      +OPERATOR KEYED IN: 1TI
14105152      TIME FOR INSANE/U000000= 1 IS: CP= 3:22, IO= 44 IN 3:52
14106117      +OPERATOR KEYED IN: 1TI
14106117      TIME FOR INSANE/U000000= 1 IS: CP= 3:45, IO= 49 IN 4:18
14106118      +OPERATOR KEYED IN: 1TL
14106118      TIME LIMITS FOR INSANE/U000000= 1 ARE: PRT= 3:20; IOT= 3:20.
14106142      +OPERATOR KEYED IN: 1DS
14106142      -OPRTR DS=ED INSANE/U000000= 1, S= 10, A= 214:0
14106143      CDB REL INPUT;INSANE/U000000= 1
14106143      ?END
14106143      PBD0224 REL 021 OUTPUT 7944;INSANE/U000000= 1
14106143      INSANE/U000000= 1 DS=ED 1406
14106144      FOR INSANE/U000000= 1: PROCESS= 247 SECS, IO= 62 SECS, OLAY= 1
14106144      PKT#0223 REMOVED

```