BEGIN 1 % NULL
GEN("PUT",3,5));
GEN("PAGE",5,3));
% 4) REWRITE 50203000
% 5) PAGE 50204000

BEGIN 1 % BLOCK;

BEGIN 1 % COMPSTAT;

BEGIN 1 % COMPILATE BODY

BEGIN 1 % COMPILATE STATEMENT

BEGIN 1 % END OF INPUT ENCOUNTERED UNEXPECTEDLY

BEGIN 1 % TOO MANY FILES IN USE.

END OF B5700 PASCAL COMPILER

BEGIN 1 % PATCH 2 FOR PASCAL CONTAINS 171 CARDS.

BEGIN 1 % PATCH TO MERGE DAG LANGMYHS PPG10 TO PPG11 COSY PATCHES

BEGIN 1 % WITH NILS OTTE'S MODIFIED PPG10 SOURCE.

BEGIN 1 % DAVID A COOPER, HERIOT-WATT UNIVERSITY, JANUARY 1978.

BEGIN 1 % FILE CARD "SOURCE" (1,10,30); % SOURCE CODE FILE 10035000

FILE 1 (1,17); % PRINT FILE 10036000

FILE PASCALGOL DISK SERIAL (20,1600) (1,10,30,SAVE 0); % CODE FILE 10037000

ARRAY PARAMTAB, FORMPARAM, FORMPARAM2(INMAXPARAMS);

FILE XREFFILE DISK SERIAL (20,1600) (1,3,30);

ALPHA ARRAY X8UFF(0,12);

BOOLEAN XINT;

INTEGER ARRAY SYMINK(0,162);

%USED IN ERROR RECOVERY.
ERRORIOOBS(COMPILE ERRORS)

%THE VALUE 07 IS SUBSTITUTED IN " " SO THIS ERROR DOES NOT INCREMENT TH
E

PACKEDSY=61#, ASSERTSY=62#;

IF ERRNUM=100
THE NERRS:=NUMERRS=1;

ERROR NUMBER 100 ALONE SHOULD NOT

PREVENT THE XALGOL COMPILATION BEING

ZIPPED AS THE VALUE 7 IS SUBSTITUTED

FOR A BAD SAVE CONSTANT IN "S"

OPTION.

7(VALID), MIDDLE, INITIAL;
"40READ", "6READLN", "5RESET", "6UNPACK", "50WRITE",
"6QJZXL" DO

IF decl THEN AX := -AX;

ABS(A(2)) LEQ ABS(B(2));

ABS
% BOOLEAN procedure XREFINPUT(A);
ARRAY A[0];
BEGIN
LABEL EOF;
INTEGER I;
READ(XREFFILE,5,XBUFF[*])(FOF);
FOR I=0,1,2 DO
A[I] := XBUFF[I];
IF FALSE THEN EOF:
BEGIN
CLOSE(XREFFILE,RELEASE);
XING := TRUE;
END;
XREFINPUT := XINB;
END OF XREFINPUT;
A2 := A2;
BOOLEAN LPARFOUND,SAVEXREFOPT;
SAVEXREFOPT := XREFOPTION := FALSE;
IF SAVEXREFOPT THEN NEWXREF(CURNAME1,CURNAME2,THISLEVEL,
FALSE);
XREFOPTION := SAVEXREFOPT;
% ASSERT 62 ASSERTSY INITIAL
IF CURNAME1="ASSERT" THEN ASSERTSY ELSE
END;
%
% THE FOLLOWING LINES DECODE ANY OCCURRENCE OF THE "$" OPTION AND
% SETS THE GLOBAL INTEGER VARIABLE "SAVEFACTOR" WHICH CONTROLS THE
% TYPE OF COMPILATION INITIATED BY THE ZIP. THERE ARE THREE LEGAL FORMS
% OF THE "$" OPTION AS FOLLOWS:
% "$" WILL GIVE NO ZIP IE. PASCAL SYNTAX CHECK ONLY
% "$" WILL GIVE A ZIP FOR COMPILE AND GO
% "$??" WILL GIVE A ZIP FOR COMPIL TO LIBRARY
% WHERE ?? IS THE TWO DIGIT DECIMAL SAVE
% CONSTANT GIVEN THE OBJECT CODE FILE
% NB. IF THE SAVE CONSTANT IS TO BE
% LESS THAN 10 THE FIRST DIGIT
% MUST BE INCLUDED IE. A "0".
%
ELSE IF CX="S" THEN
BEGIN
IF C="-" THEN SAVEFACTOR:=1 ELSE
IF C="*" THEN SAVEFACTOR:=0 ELSE
IF C LEQ 9 THEN
BEGIN
SAVEFACTOR := 10 X CJ; NEXTCHAR;
SAVEFACTOR := SAVEFACTOR + CJ;
IF C GT 9 THEN ERROR (100);
END;
ELSE
BEGIN
ERROR(100);
END
END
%
SAVEFACTOR = 7

END;

% INTEGER EXPRLVLE=TX,EXPINVARCNT;
BOOLEAN INBRACKET,INRECORD,SIMPLEVAR;
SIMPLEVAR = FALSE;
CURTYPE = THISID,TYPE;
SIMPLEVAR = TRUE;
SIMPLEVAR = FALSE;
EXPINVARCNT = EXPINVARCNT + 1;
SIMPLEVAR = SIMPLEREVAR;
IF EXPINVARCNT = 0 THEN WRITEXEPR;
LABEL EFH;

BEGIN
GEN("QQJZKL",6,2);
INSYMBOL;
GO TO EFH;
END;

EFH:
EXPRLVLE = 1;
IF THISID,IDCLASS = VAR OR THISID.IDCLASS = CONST AND BOOLEAN(THISID,FORMAL) THEN %
EXPRLVLE = 0;
PROCEDURE ASSETSTAT;
BEGIN
GEN("IF NOT","*",7);
INSYMBOL BOOLEAN EXPRLVLE;
GEN("=",7,2,2,2);
GEN("RUNERR","","","",7,6);
GEN("","",7,2,2,2);
GEN("CARDCNT",7,7);
END OF ASSETSTAT;
IF CURNAME = "QQJZKL" THEN FILEHANDLING(6) ELSE
IF CURSY = ASSERTSY THEN ASSERTSTAT ELSE
IF PARAM THEN GEN("0","",1,7) ELSE BEGIN
GEN("0","",1,2,6);
GEN("","",1,7);
END OF ASSERTSTAT
FORMPARAM1[NUMPARAMS] = CURNAME;
FORMPARAM2[NUMPARAMS] = CURNAME;
INTEGER INDEX, CTYPE, NUMFORWARDS, T, TX, I;
ALPHA T3;
LABEL LL1;
LABEL LL2;
LABEL LL3;
IF CURLEVEL GEQ MAXTABLES THEN ERROR(101) ELSE BLOCKTABLE[CURLEVEL+1] = NUMBLOCKS = NUMBLOCKS + 1;
NAMETAB3[CURLEVEL][THISINDEX].FORWARDDEF = 0;
T = NAMETAB3[CURLEVEL][THISINDEX].INFO;
TX = T + PARAMTAB[T];
FOR I=1 TO 1 STEP 1 UNTIL TX DO
NAME=LNAME(FORMPARAM1[I],FORMPARAM2[I],CURLEVEL+1);
REPLACE POINTER(NAMETAB3[CURLEVEL+1][I]) BY 0;
FOR MAXNAMES+1 WORDS;
IF CURLEVEL GEO LASTREC THEN ERROR(101) %

SAVEFACTOR=0.1%  * DEFAULT ZIP IS COMPILE AND GO UNLESS
%   CHANGED BY THE USE OF THE "S" OPTION
%   THE FOLLOWING  LINES ADD A "0" ONTO THE FRONT OF THE PROGRAM NAME OR90042200
%   THE FIRST SIX CHARACTERS THEREOF IF IT IS LONGER THAN SIX CHARACTERS
%   THIS GIVING THE NAME OF THE XALGOL OBJECT CODE FILE PRODUCED.
%   PROGNAM = CURNAME1.(35136))  PROGNAMELENGTH = MIN(6,CURLENGTH)+1;
%   BEGIN%
WRITE(LINE,NOERRORS) %
IF ERROR(100)%
THEN WRITE(LINE,ERROR100WESS) %
IF SAVEFACTOR20 THEN%  *A ZIP IS REQUIRED
$VOID
END%

("100,ILLEGAL SAVE CONSTANT IN "****S****" OPTION, THE VALUE 07 IS91106500
SUBSTITUTED")%  * SO THIS ERROR DOES NOT INCREASE THE COMPIlATION ERROR COUNT.
RS COUNT,"%  91106700
("101,PROCEDURES/FUNCTIONS NESTED TOO DEEP.")%  91106800
REWIND(XREFILE);
92003500
$SEQ(XFUNEXP,RENAMEXREF,RENAMEFNAME,REFINPUT,0,RENAMEXREF,RENAMEFNAME,3,1000,6000))%  92005000
$# PATCH 500 FOR PASCAL.XVI.O CONTAINS 5 CARDS. PRT CELLS 25 TO 30
$1 THIS PATCH CORRECTS THE DOCUMENTATION FOR THE COMPILERS PRT CELLS 25 TO 27
$1 (NOT 21 TO 23), FURTHERMORE, IT USES PRT CELL 30 FOR THE CARD COUNT (IN PLACE
$1 OF 27) TO BE CONSISTANT WITH THE OTHER SYSTEM COMPILERS. PRT CELL 27 IS USED
$1 FOR THE PAGE COUNT FORMERLY AT SEQUENCE 1014300.
$1 NILS A OTTE, UNIVERSITY OF NATAL, DURBAN, AUG = NOV 1977.
$1
$1 INTEGER NUMERRS%,  % RR+251 NUMBER OF ERRORS IN PROGRAM,
%   SAVEFACTOR%,  % RR+261 SAVEFACTOR FOR CODE FILE.
%   PAGECNT%,  % RR+271 NUMBER OF PAGES PRINTED.
%   CARDCNT%,  % RR+301 NUMBER OF CARDS READ.
INTEGER LINCNCT, ERRINX, % PAGECNT % PRT+27
$# PATCH 501 FOR PASCAL.XVI.O CONTAINS 3 CARDS. *PRT25* FOR USER'S PASCAL PROG.
$1 THIS PATCH INCORPORATES THE PRE-DEFINED IDENTIFIER "PRT25" LOCATED
$1 AT PRT CELL 25 AS PER DOCUMENTATION. (THE DOCUMENTATION MUST BE
$1 AMENDED TO DELETE PRT26 AND PRT27 FROM THE PRE-DEFINED IDENTIFIER LIST.)
$1 ** NOTE THAT FILE PASCAL.PRELUDE MUST BE UPDATED FOR "PRT25",
$1 THE VARIABLE "PRT25" MAY BE SET BY THE G COMMON = N CONTROL CARD.
$1 NILS A OTTE, UNIVERSITY OF NATAL, DURBAN, AUG = NOV 1977.
$1
$1 NEWNAME("50PRT25",0,0);  %** "PRT25" *** 20368100
T3=INTYPE; T3.IDCLASS=VAR;  % GLOBAL INTEGER VARIABLE 20369200
NAMETAB(30,THTINDEX) = T3;  20369300
$# PATCH 502 FOR PASCAL.XVI.O CONTAINS 3 CARDS, LINE COUNT WHEN DEBUGGING
$1 TO CORRECT THE LINE COUNT WHEN THE DEBUGGING OPTION TO LIST THE ALGOL
$1 CODE GENERATED IS SET (**S**), OTHERWISE LINES PER PAGE GOES WRONG.
$1 NILS A OTTE, UNIVERSITY OF NATAL, DURBAN, AUG = NOV 1977.
$1
DEFINELINESPERPAGE = 60 %,

$1

10038000
IF DUMP OPTION THEN BEGIN IF (LINECNT+1=LINECNT+1) THEN LINES PER PAGE 20149000
THEN HEADING WRITE(LINE,10,ALGOLCAR0(11)) END 20149100

$# PATCH 503 FOR PASCAL.XVI.O CONTAINS 9 CARDS. INTEGER TO REAL FOR TYPE TAB.
$# WHEN MORE THAN 63 ENTRIES WERE ENTERED IN THE "TYPEB**" ARRAYS, THE
$# PASCAL COMPILER WAS DISCONTINUED DUE TO INTEGER OVERFLOW, WHICH COULD OCCUR
$# IN A NUMBER OF PROCEDURES AS A RESULT OF ASSIGNING TO AN INTEGER AN ARRAY
$# ELEMENT WHOSE EXPONENT FIELD WAS NOT ZERO. THE FIELD "ARRTYPE" IS
$# [43110] AND HAS THE 4 HIGH ORDER BITS IN THE EXPONENT FIELD. THIS PATCH
$# ALTERS THE DECLARATIONS OF ALL IDENTIFIERS TO WHICH "TYPEB1" MAY BE
$# ASSIGNED FROM INTEGER TO REAL TO CORRECT THIS ERROR.
$# NILS A OTTE, UNIVERSITY OF NATAL, DURBAN, AUG - NOV 1977.

INTEGER ITJ REAL TJ
INTEGER ITJ REAL TJ
INTEGER CASETYPE ADDR MAXANDR INDEX CTYPE TX S X T3 LLIM ULIM IT
REAL T, CV
INTEGER LEVEL1000, YT, XAM, NAMTAB, I, J, RESIZE
ALPHA Ti, FNAME
INTEGER FIRSTPAR, CURINFO Pi PX I T3 J REAL TJ
INTEGER INDEX CTYPE, NUMFORWARDS, T3 TX IT
REAL T, CV

$# PATCH 504 FOR PASCAL.XVI.O CONTAINS 23 CARDS. IMPLEMENT FORWARD DECLARATIONS
$# FORWARD DECLARATIONS OF PROCEDURES ENDED IN CHAOS DUE TO THE PARAMETERS AND
$# THEIR TYPES NOT BEING KEPT RESULTING IN GLOBALS BEING REFERENCED WHERE
$# POSSIBLE, AND FORWARD DECLARATIONS NOT WORKING AT ALL.
$# THE PROBLEM WAS THAT THE INFORMATION ON THE PARAMETERS WAS BEING STORED
$# IN THE "NAMETAB**" ROWS FOR THE CURRENT LEVEL, WHICH WERE BEING SET TO ZERO
$# ON EXIT FROM PROCEDURE BLOCKS AT THAT LEVEL THEREAFTER.
$# THIS PATCH CORRECTS THE ERROR BY MARKING THE ENTRIES FOR PARAMETERS OF
$# FORWARD PROCEDURES AND FUNCTIONS, SETTING TO ZERO ONLY THOSE ELEMENTS WHICH
$# ARE NOT SO MARKED ON EXIT FROM A BLOCK, AND UNMARKING THE RELEVANT PARAMETERS
$# WHEN THE PROCEDURE OR FUNCTION IS DEFINED. THE MARKING OF THE PARAMETERS
$# IS DONE IN SUCH A WAY THAT THE SAME IDENTIFIER NAME MAY BE USED AT THE SAME
$# LEVEL WITHOUT SYNTAX ERROR TO REPORT THAT THE IDENTIFIER IS ALREADY DEFINED
$# THE UNMARKING REPLACES THE IDENTIFIER NAME IN "NAMETAB**" TO ALLOW FOR THE
$# SAME NAME OR ONE THAT HASHES TO THE SAME PLACE TO HAVE BEEN USED PREVIOUSLY
$# AND THEN DELETED.
$# NILS A OTTE, UNIVERSITY OF NATAL, DURBAN, AUG - NOV 1977.

IF FOUND AND THISID=IDCLASS=FUNC THEN
NAMETAB:[CURLEVEL][THISINDEX][FORWARDDEF]=0
(THISID,IDCLASS=FUNC AND NOT FUN) THEN ERROR(43)
TXI=(T1=THISID,INFO)+PARAMTABK1
% UNMARK FORWARD PARAMS 80556000
FOR I=1 TO IF = 1 UNTIL TX DO % TO ALLOW REFERENCE 80557000
BEGIN T3=PARAMTABK1,PARAMNAME
CURNAME1=ABS(NAMETAB[CURLEVEL+1,T3])
CURNAME2= NAMETAB[CURLEVEL+1,T3]
NAMETAB[CURLEVEL+1,T3]=0
NEWNAME(CURNAME1,CURNAME2,CURLEVEL+1)
IF T3=THISINDEX THEN BEGIN
PARAMTABK1,PARAMNAME=THISINDEX
NAMETAB[CURLEVEL+1,THISINDEX]=
NAMETAB[CURLEVEL+1,T3]
END END % OF UNMARKING FORWARD PARAMETERS,
TXI=(T1=NAMETAB[CURLEVEL][INDEX][INFO]+PARAMTABK1)
FOR I=1 TO IF = 1 UNTIL TX DO % MARK FORWARD PARAMETERS 80636000
NAMETAB[CURLEVEL+1,PARAMTABK1,PARAMNAME],(4611)=1
80636210
TX1=CURFUNC1; CURFUNC1=IF FUN THEN INDEX ELSE -1;
FOR I=0 STEP 1 UNTIL MAXNAMES DO % LEAVE FORWARD PARAMETERS
END FOR; IF NAMETAB[CURLEVEL]1=0 THEN NAMETAB[CURLEVEL]1=0;
CURLEVEL=1; CURFUNC1=TX1; CURLEVEL=1;
$# PATCH 505 FOR PASCAL.XVI.O CONTAINS 9 CARDS. CHECK FOR HASH TABLE FULL
$# WHEN THERE ARE "MAXNAMES" IDENTIFIERS AT ONE LEVEL THE "NAMETAB" ROWS
$# BECOME FULL AND THIS USED TO PUT THE ONE COMPILER INTO AN INFINITE LOOP
$# EITHER IN "NEWNAME" OR "SEARCHTAB". THIS PATCH INSERTS TEST FOR WRAP AROUND
$# LEADING BACK TO THE MASHED STARTING POINT, FOR WHICH IT GIVES SYNTAX ERROR
$# 40, TOO MANY IDENTIFIERS DECLARED,
$# NILS A OTTE, UNIVERSITY OF NATAL, DURBAN, AUG = NOV 1977.
$1
DEFINE HASH(HASH1) = ENTER((HASH1) MOD MAXNAMES), #1;
BEGIN ALPHA NAME1 INTEGER WRAPAROUND;
WRAPAROUND=THISINDEX=HASH(CURNAME1), #1;
IF THISINDEX=WRAPAROUND THEN NAME1=0; % TABLE IS FULL
ALPHA NAME1 INTEGER WRAPAROUND;
WRAPAROUND=THISINDEX=HASH(CURNAME1), #1;
IF THISINDEX=WRAPAROUND THEN % TABLE AT THIS LEVEL IS FULL
BEGIN ERROR,NAME1=NAME2=NAMETAB[2][TAB][INDEX2]=20242000
END
$# PATCH 506 FOR PASCAL.XVI.O CONTAINS 2 CARDS. RESERVED WORD ENDING AT CC 80
$# IF A RESERVED WORD ENDED AT CARD COLUMN 79 OR 80 AND IF THE "BOLDFACE" FOR
$# RESERVED WORDS OPTION IS SET (*$R*), AN INVALID INDEX OCCURRED IN THE
$# SCANNER "INSYMBO", THE PROBLEM IS CURED BY CORRECTLY COMPUTING THE STARTING
$# AND ENDING POINT OF THE RESERVED WORDS.
$# NILS A OTTE, UNIVERSITY OF NATAL, DURBAN, AUG = NOV 1977.
$1
BEGIN T1 = CARDLENGTH=CHARCNT=CURLENGTH=1;
FOR CURLENGTH=REAL (CHARCNT=0);
$# PATCH 507 FOR PASCAL.XVI.O CONTAINS 5 CARDS. "VARIABLE", "SIMPLEVARIABLE"
$# IN PROCEDURE "VARIABLE", "SIMPLEVARIABLE" IS SET TRUE IF A SUBSCRIPT IS
$# SIMPLE. RESULTING IN ALCODE CODE BEING WRITTEN PREMATURELY DURING RECURSIVE
$# CALLS ON PROCEDURE "EXPRESSION", WHICH IN SOME CASES LEAD TO ALGOL SYNTAX
$# ERRORS, SINCE WRITING THE ALGOL CODE IS DEPENDANT ON "EXPRLEVEL" BEING ZERO,
$# THIS PATCH BUMPS ITS VALUE PRIOR TO ANALYSING THE SUBSCRIPT, AND SETS
$# "SIMPLEVARIABLE" FALSE AFTERWARDS.
$# NILS A OTTE, UNIVERSITY OF NATAL, DURBAN, AUG = NOV 1977.
$1
EXPRLEVEL = EXPRLEVEL+1; % DO NOT "WRITEEXPR" YET
EXPRLEVEL = EXPRLEVEL-1;
SIMPLEVARIABLE = FALSE; % RECURSION ON "VARIABLE"
EXPRLEVEL = EXPRLEVEL+1;
EXPRLEVEL = EXPRLEVEL-1;
$# PATCH 508 FOR PASCAL.XVI.O CONTAINS 1 CARD. "CONCAT" A FUNCTION OF ANY TYPE
$# THE INTRINSIC FUNCTION "CONCAT" COULD ONLY BE ASSIGNED TO A VARIABLE DECLARED
$# "REAL" TO AVOID TYPE CONFLICT SYNTAX ERRORS. THIS PATCH MAKES "CONCAT"
$# TYPELESS.
$# NILS A OTTE, UNIVERSITY OF NATAL, DURBAN, AUG = NOV 1977.
$1
CURTYPE = D\% ALFATYPE OR REALTYPE;
$# PATCH 511 FOR PASCAL.XVI.O CONTAINS 7 CARDS. ALLOW UP-LEVEL ADDRESSING
$# TO ALLOW UP-LEVEL IDENTIFIER REFERENCES, FORMERLY, REFERENCES TO GLOBAL
$# IDENTIFIERS WHICH WERE NOT IN THE OUTER BLOCK WERE FLAGGED BY SYNTAX ERROR
$# 5, UP-LEVEL ADDRESSING NOT IMPLEMENTED DUE TO HARDWARE RESTRICTION.
$# ALTHOUGH THE RESTRICTION EXISTS IN EXTENDED ALGOL, IT IS NOT TRUE THAT THE
PROCEDURE WRITEXPR;  *** FIX STRUCTURE FOR ASSIGNMENT
BEGIN  % USED ONLY IN ASSIGNMENT OF STRUCTURES
   IF NUMPOINTER > 0 DO
      BEGIN NUMPOINTER := NUMPOINTER + 1;
         IF NUMSYMS > 4 THEN WRITEEXPR;
         REPAIR_POINTER(SYNTAXNUMSYMS) BY "00:00:00 1022,00 0 T 00000 10222";
      END;
   END % OF WHILE
   WRITEEXPR;  GEN ("\", 1, 7 )
END WRITEEXPR;

ERROR(95); % STRUCTURES NOT IMPLEMENTED.
GEN("ASSIGN(\", 7, 1 ) WRITEEXPR
EXPRESSION;
GEN(TYPETABLE[LEFTYPE], SIZE) GEN("\", 1, 7 )
IF TYPETABLE[LEFTYPE].SIZE != TYPETABLE[CURTYPE].SIZE THEN ERROR(95)
END;
CHECKTYPES( LEFTYPE, CURTYPE )
("95 SIZE OF STRUCTURES IN ASSIGNMENT ARE NOT THE SAME.");

$# PATCH 913 FOR PASCAL-XVI.0 CONTAINS 16 CARDS. FIX POINTERS VIA POINTERS
$ TO CORRECT THE CODE GENERATFD FOR CHAINED REFERENCES THROUGH THE HEAP.
IF NUMSYMS+6 ≤ MAXSYMS THEN 40175000
  NUMSYMS := NUMSYMS+2; 40180000
IF NUMPOINTERs > 0 THEN BEGIN 40180500
  % POINTER VIA POINTER
  BEGIN REPLAC€ POINTER(SYMTAB[NUMSYMS+1]) BY
  "001100100 102200 0 T MOD00 102211";
  NUMSYMS := NUMSYMS+4;
END; 40180700
ELSE BEGIN 40180800
  INBRACKET := FALSE;
  BEGIN NUMPOINTERs := NUMPOINTERs+1;
  IF NUMSYMS+4 ≤ MAXSYMS THEN
  BEGIN REPLAC€ POINTER(SYMTAB[NUMSYMS+1]) BY
  "001100100 102200 0 T MOD00 102211";
  NUMSYMS := NUMSYMS+4;
END; 40180900
  ELSE ERROR(63) % EXPRESSION IS TOO LONG FOR SYMTAB[*] 40181000
  END; 40181000
$# PATCH 514 FOR PASCAL.XVI.O CONTAINS 2 CARDS, PROCESS TIME FUNCTION FOR RUN
$# PATCH TO CHANGE THE NAME OF THE FUNCTION ON THE OS730 VERSION WHICH SUPPLIES $# THE PROCESS TIME USED BY THE PASCAL PROGRAM ON THE CURRENT RUN FROM "ELAPSED" $# WHICH MEANS PLATFORM TIME, TO "CPU TIME" WHICH IS THE WIDELY ACCEPTED TERM
$# FOR THIS QUANTITY.
$# NILS A OTTE, UNIVERSITY OF NATAL, DURBAN. AUG - NOV 1977.
$# NEWNAME("7CPU TIME", "E", 0;)
$# NAME=3010 THIS INDEX]1-T3; 20300000
$# IF CURNAME="7CPU TIME" AND CURNAME2="E" THEN % "CPU TIME" 40452000
$# PATCH 516 FOR PASCAL.XVI.O, CONTAINS 2 CARDS, CORRECT "NO LISTING" ERROR $# THIS PATCH CORRECTS AN ERROR WHEREBY IF LISTING WAS TURNEO OFF $# AND PAGE THOQ WAS INVOKED, A HEADING WAS PRINTED REGARDLESS.
$# DAVID A COOPER, HERIOT-WATT UNIVERSITY, JUNE, 1976;
$# IF CX="L" THEN IF CO=1 THEN 30264000
$# IF LISTOITION THEN HEADING ELSE 30264500
$# PATCH 517 FOR PASCAL.XVI.O, CONTAINS 2 CARDS, $# THIS PATCH CORRECTS AN ERROR THAT CAUSED A FILE DECLARATION $# TO HAVE ITS NAME STRING SPLIT OVER TWO LINES IN THE GENERATED XALGOL.
$# ALSO CHANGES SYMTAB FORM TYPE REAL TO TYPE ALPHA. $# DAVID A COOPER, HERIOT-WATT UNIVERSITY, JUNE, 1976;
$# ALPHA ARRAY SYMATB[0:MAXSYMS]; % USED BY "EXPRESSION", 10144000
$# IF ALGOLCNT LSS 1 THEN WRITE(XALGOL)
$# PATCH 518 FOR PASCAL.XVI.O, CONTAINS 224 CARDS, $# THIS PATCH CHANGES THE WAY THAT MULTI DIMENSION ARRAYS $# REPRESENTING RECORDS ARE DECLARED. PREVIOUSLY THEY WERE DECLARED $# THE WRONG WAY ROUND FOR XALGOL, THIS PATCH SORTS THE DIMENSIONS $# INTO ASCENDING ORDER FROM LEFT TO RIGHT AND GENERATES APPROPRIATE $# DEFINES AND CODE FOR HANDLING THE ARRAYS.
DEFINE
PERMSUB = 0 #, MAXTOTALSUBSCR = 100 #,
         ARRNAM = 1 #;
ARRAY ARRSUBPERMTABL(01,01,MAXTOTALSUBSCR),
         INTEGER PASSPERMTAB, MAXPERMTAB, REMEMBERPSNZ;
$Boolean SIMPLEVARIABLE, INSIDEBrACKETS, INSIDEparenS$;
$IF INSIDEparenS AND TYPETABL(1, CURTYPE), STRUCT > 0 AND
TYPETABL(CURTYPE), FORM < FILES THEN
PUDT("H", 1000XTHISLEVEL+THISINDEX*5)
ELSE
PUDT("V", 1000XTHISLEVEL+THISINDEX*5)
INSIDEparenS = TRUE;
INSIDEparenS = FALSE;
$GENID("H", 1000XTHISLEVEL+THISINDEX*5)
GENID("H", 1000XTHISLEVEL+THISINDEX*5);
$SET VOIDT
$POP VOIDT
DEFINE
LONSUB = 0 #,
MISUB = 1 #,
NEXTSUB = 2 #,
MAXNOFSUBSCRIPTS = 20 #,
STOPPERSUBTAB = 21 #;
ARRAY ARRSUBSTRANGE(1201, MAXNOFSCRIPTS);
INTEGER FIRSTRANGE, NEXTFREEENTRY, PASSSUBRANGE, PREVPASS,
MP, POSNO, SUBDIFF;
IF ARRAYVAR THEN GEN("/", 1,7) ELSE ARRAYVAR = TRUE;
IF NOT PARAM THEN
BEGIN
GEN("DEFINE", 7, 2)
GENID("V", LEVEL1000+NUM*5)
GEN("T", 1,7)
END;
FIRSTRANGE = STOPPERSUBTAB; NEXTFREEENTRY = 0;
POSNO = 1;
MP = 10; FIRSTDIM = TRUE;
DO
BEGIN
IF FIRSTDIM THEN FIRSTDIM = FALSE ELSE
BEGIN
IF NOT PARAM THEN GEN("V", 1,7)
END;
IF NOT PARAM THEN GENID("V", LEVEL1000+NUM*MP*POSNO, IF MP=10 80064115
THEN 6 ELSE 7), POSNO = POSNO + 1
IF POSNO = MP THEN MP = MP+10;
IF NEXTFREEENTRY = STOPPERSUBTAB THEN
BEGIN
ERROR(0)
END
ELSE
BEGIN
80064190
END
80064150
ARRSUBSCRIPTRANGE[LOWSUBS,NEXTFREEENTRY]=TYPETAB2[TYP];
ARRSUBSCRIPTRANGE[HISUBS,NEXTFREEENTRY] = TYPETAB3[TYP];
END;
SUBDIFF = TYPETAB3[TYP] - TYPETAB2[TYP];
IF FIRSTRANGE = STOPPERSUBTAB THEN
BEGIN
FIRSTRANGE = NEXTFREEENTRY;
NEXTFREEENTRY = NEXTFREEENTRY + 1;
ARRSUBSCRIPTRANGE[EXTSUBS,FIRSTRANGE] := STOPPERSUBTAB;
END;
ELSE
BEGIN
PASSSUBRANGE := FIRSTRANGE;
PREVPASS := STOPPERSUBTAB; NEXTFREEENTRY = NEXTFREEENTRY + 1;
WHILE (SUBDIFF >= ARRSUBSCRIPTRANGE[HISUBS,PASSSUBRANGE])
AND (ARRSUBSCRIPTRANGE[EXTSUBS,PASSSUBRANGE] #
STOPPERSUBTAB) DO
BEGIN
PREVPASS := PASSSUBRANGE;
PASSSUBRANGE := ARRSUBSCRIPTRANGE[EXTSUBS,
PASSSUBRANGE];
END;
IF PREVPASS = STOPPERSUBTAB THEN
BEGIN
IF SUBDIFF < ARRSUBSCRIPTRANGE[HISUBS,
PASSSUBRANGE] =
ARRSUBSCRIPTRANGE[LOWSUBS,
PASSSUBRANGE] THEN
BEGIN
ARRSUBSCRIPTRANGE[EXTSUBS,PASSSUBRANGE] :=
NEXTFREEENTRY - 1;
ARRSUBSCRIPTRANGE[EXTSUBS,NEXTFREEENTRY] :=
STOPPERSUBTAB;
END;
ELSE
BEGIN
ARRSUBSCRIPTRANGE[EXTSUBS,NEXTFREEENTRY] :=
FIRSTRANGE;
FIRSTRANGE := NEXTFREEENTRY+1;
END;
ELSE
BEGIN
IF SUBDIFF < ARRSUBSCRIPTRANGE[HISUBS,PASSSUBRANGE] =
ARRSUBSCRIPTRANGE[LOWSUBS,PASSSUBRANGE] THEN
BEGIN
ARRSUBSCRIPTRANGE[EXTSUBS,PASSSUBRANGE] :=
NEXTFREEENTRY - 1;
ARRSUBSCRIPTRANGE[EXTSUBS,NEXTFREEENTRY] :=
STOPPERSUBTAB;
END;
ELSE
BEGIN
ARRSUBSCRIPTRANGE[EXTSUBS,PREVPASS] :=
NEXTFREEENTRY -1;
DO
BEGIN
REMEMBERPOSN := PASSPERMTAB;
GEN("DEFINE","e+2");
NAMOFTHING := ARRSUBPERMTAB[ARRNAM,PASSPERMTAB];
GENID("v",1000xCURLEVEL+NAMOFTHING+5);
GEN("e","i","j");
FIRSTTIME := TRUE;
DO
BEGIN
IF FIRSTTIME THEN FIRSTTIME := FALSE ELSE GEN("e","i"),180421190,
("e","i","j");80421200
DIFF := PASSPERMTAB-REMEMBERPOSN+1;
GENID("v",1000xCURLEVEL+NAMOFTHING+10)+DIFF
{IF DIFF > 9 THEN 100 ELSE 10}+DIFF
{IF DIFF > 9 THEN 7 ELSE 6};
PASSPERMTAB := PASSPERMTAB + 1; END
UNTIL PASSPERMTAB = MAXPERMTAB OR
ARRSUBPERMTAB[ARRNAM,PASSPERMTAB] = "i"
GEN("e","i","j");
GEN("e","i","j");
GENID("h",1000xCURLEVEL+NAMOFTHING+5);
GEN("e","i","j");
PASSEMPERMTAB := REMEMBERPOSN; FIRSTTIME := TRUE;
DO
BEGIN
IF FIRSTTIME THEN FIRSTTIME := FALSE ELSE GEN("e","i"),180421380,
("e","i","j");80421390
DIFF := ARRSUBPERMTAB[PERMSUB,PASSPERMTAB]+1
GENID("v",1000xCURLEVEL+NAMOFTHING+10)+DIFF
{IF DIFF > 9 THEN 7 ELSE 6};
PASSPERMTAB := PASSPERMTAB + 1;
END
UNTIL PASSPERMTAB = MAXPERMTAB OR
ARRSUBPERMTAB[ARRNAM,PASSPERMTAB] = "i"
GEN("e","i","j","i","j");
GEN("e","i","j","i","j");
END
UNTIL PASSPERMTAB = MAXPERMTAB
MAXPERMTAB := 0;
END
ENDJ
BEGIN
INTEGER NAM,TI,SCRATCHEH
NAM := PARAMTAB[1],0:10;
SCRATCH := NAMETAB[1:CURLEVEL+1:NAM];
SCRATCH := SCRATCH+TYPE;
TI := TYPETAB[SCRATCH];
IF TI+STRUCT < 0 AND TI+FORM < FILES THEN
GENID("h",1000xCURLEVEL+1)+NAM+5)
ELSE
GENID("v",1000xCURLEVEL+1)+NAM+5);
END
MAXPERMTAB := 0;
INSIDEHAREN := FALSE;
$ PATCH 519 FOR PASCAL XVI,O, CONTAINS 1 CARDs, INCREASE RUNTIME STACK,
"XALGOL STACK = 2048; STACK = 1024; END."

PATCH 600 FOR PASCAL.XVI.O. CONTAINS 22 CARDS, UAGS DEC77 PATCHES.
PATCHES RECEIVED FROM D.LANGMYHR AND TRANPOSED FROM COSY FORMAT BY
DAVID A. COOPER, FEBRUARY 1978.

IF(F1 NEQ SET OR RT NEQ EMPTYSET) AND%
(C2 NEQ SET OR LT NEQ EMPTYSET) THEN%
IF(F1 NEQ POINTERS OR RT NEQ NILTYPE) AND%
(C2 NEQ POINTERS OR LT NEQ NILTYPE) THEN%
BEGIN ERROR(63) %

GEN("PREAD(" ,5,2") WRITEEXPR GEN("","1,7"); %

GENID("F",FILEID,5) GEN("","1,7"); %
IF F=NUMERIC THEN %
BEGIN %
GEN("","1,7"); GENINT(TYPE=TAB2(CURTYPE)); %
GEN("","1,7"); GENINT(TYPE=TAB3(CURTYPE)); %
END ELSE GEN("","0","4,4"); %

SET VOID POP VOID
IF NAMETAB.IDCLASS=FUNC THEN GEN("FUNCTION","7,2"); %
ELSE GEN("PROCEDURE","8,1"); %

IF FOUND AND (THISID,IDCLASS=PROC OR THISID,IDCLASS=FUNC) THEN

PATCH 601 FOR PASCAL.XVI.O. CONTAINS 147 CARDS. EXTENDE SET MODS.
PATCHES RECEIVED FROM D.LANGMYHR AND TRANPOSED FROM COSY FORMAT BY
DAVID A. COOPER, FEBRUARY 1978.
THIS PATCH MODIFIES THE SET HANDLING ROUTINES TO ALLOW SETS OF 0..93
ELEMENTS.
NB. THE RUN TIME SYSTEM MUST BE CHANGED ACCORDINGLY.....

PROCEDURE SPLIT(SPLITINX,WIDTH); %
VALUE SPLITINX, WIDTH; %
INTEGER SPLITINX, WIDTH; %
BEGIN %
    INTEGER IJ %
    IF NUMSYMS+WIDTH LEQ MAXSYMS THEN%
    BEGIN %
        FOR I=NKSYMS STEP -1 UNTIL SPLITINX DO%
            SYMTAB[I+WIDTH] = SYMTAB[I]; %
        FOR I=1 STEP 1 UNTIL WIDTH DO%
            SYMTAB[SPLITINX+I-1] = "300000000"; %
        NUMSYMS = NUMSYMS + WIDTH; %
    END %
ELSE
    BEGIN %
        ERROR(63) %
        NUMSYMS = IJ; %
    END %
END OF SPLIT; %
% END %

IF TYPETABTYPEFORM=SET THEN % *** SET VARIABLES
BEGIN % *** --- --------

INTEGER THISSYM, I; %

SPLIT(STARTSYM+1); SYMTAB[STARTSYM] := "SLOAD"; %
IF SIMPLEVAR THEN %
BEGIN %

PUTSYM("",""); %
PUTID("W",1000+THISLEVEL+THISINDEX,5); %
END %
ELSE %

IF INBRACKET AND NOT INRECORD THEN %
BEGIN %

PUTSYM("",""), THISSYM = NUMS; %
PUTSYM(O); PUTSYM(" "); PUTSYM(" "); %
FOR I=STARTSYM+1 STEP 1 UNTIL THISSYM DO %
PUTSYM(SYMTAB[I]);
PUTTEXT(" "); I %
END %
ELSE %

THISSYM = NUMS; %
IF INBRACKET THEN PUTSYM("""); %
FOR I=1 STEP 1 UNTIL NUMPOINTERS DO %
BEGIN %

PUTTEXT("(1)DIV"); PUTTEXT(" 1022"); %
PUTTEXT("(1)MOD"); PUTTEXT(" 1022"); %
END %

PUTSYM("",""); %
FOR I=STARTSYM+1 STEP 1 UNTIL THISSYM DO %
PUTTEXT(SYMTAB[I]); %
PUTTEXT(" "); I %
IF INBRACKET THEN PUTSYM("""); %
FOR I=1 STEP 1 UNTIL NUMPOINTERS DO %
BEGIN %

PUTTEXT("(1)DIV"); PUTTEXT(" 1022"); %
PUTTEXT("(1)MOD"); PUTTEXT(" 1022"); %
END %
NUMPOINTERS := 0; %
END;

PUTSYM("",""), PUTSYM(CHARCOUNT); PUTSYM(""), %
END OF SET VARIABLES; %

IF TYPETABTYPEFORM=SET THEN
BEGIN %

GEN("","1,7"); %
GENID("W",1000+THISLEVEL+THISINDEX,5); %
END %

BOOLEAN FIRST, SPLITTED; %
PUTTEXT("SETS("); PUTTEXT(" 3,2"), PUTSYM(CHARCOUNT); %
PUTSYM(""), %
CURTYPE := EMPTYS; CURMODE := NUMBER; %
STARTSYM := NUMSYM + 1; %
PUTSYM(" SETS("); %
PUTSYM(""), SYMTAB[STARTSYM] := "SETBS(" %

00052965
00186005
00186010
00186025
00186030
00186035
00186040
00186045
00186050
00186055
00186060
00186065
00186070
00186075
00186080
00186085
00186090
00186095
00186100
00186105
00186110
00186115
00186120
00186125
00186130
00186135
00186140
00186145
00186150
00186155
00186160
00186165
00186170
00186175
00186180
00186185
IF SPLITTED THEN PUTSYM(""); %
IF CURSY=COMMA THEN %
BEGIN %
SPLIT(STARTSYM,1); SYMTAB[STARTSYM] := "SUNIO("; %
PUTSYM(";"); %
SPLITTED := TRUE; %
END %
NEWTYPE; T1 := SET; T1.SIZE := 2; T1.STRUCT := 0; %
CURMODE := NUMBER; %
IF CURSTYPE=BOOLETYPE THEN %
IF CURSY NEQ ANDSY THEN ERROR(64))
END ELSE %
IF F=SET THEN %
BEGIN %
IF CURSY=ASTERISK THEN %
BEGIN %
SPLIT(STARTSYM,1); SYMTAB[STARTSYM] := "SINTS("; %
PUTSYM(";"); %
END ELSE ERROR(64)); %
MODE := NUMBER; %
IF F=SET THEN PUTSYM("*"); %
SPLIT(STARTSYM,1); %
IF CURSY=PLUS THEN SYMTAB[STARTSYM] := "SUNIO(" ELSE %
IF CURSY=MINUS THEN SYMTAB[STARTSYM] := "SDIFF(" ELSE %
ERROR(64)); %
PUTSYM("*"); MODE := NUMBER; %
IF F=SET THEN PUTSYM("*"); %
IF CURSY=EQLSY THEN SYMTAB[STARTSYM] := "SEQUA(" %
ELSE %
IF CURSY=NEQSY THEN %
BEGIN %
SPLIT(STARTSYM,1); SYMTAB[STARTSYM] := " NOT "; %
SYMTAB[STARTSYM,1] := "SEQUA("; %
IF TYPETAB1[LEFTTYPE].FORM=SET THEN %
BEGIN %
SYMTAB[1] := "SSTOR("; NUMSYMS := NUMSYMS - 3; %
EXPRESSION; %
PUTSYM("*"); CHECKTYPES(LEFTTYPE,CURTYPE); %
WRITEEXPR %
END ELSE %
IF TYPETAB2[1].1 < OR TYPETAB3[1].1 GTR 93 THEN ERROR(51))
T1.SIZE := T1.SIZE := 2; TYPETAB1[TYPEINDEX] := T1; %
IF T1.FORM=SET THEN %
BEGIN %
GEN("","@7); GENID("K",LEVEL1000+NAM,5); %
END %
IF T1.FORM=SET THEN %
BEGIN %
GEN("","@7); %
END %
BEGIN %
IF T1.FORM=SET THEN %
BEGIN %
GEN("","@7); %
END %
VALUE NAME1, NAME2, TABLE, DECLJ
REAL NAME1, NAME2;
INTEGER NAME1, NAME2;
DECLJ BOOLEAN DECL;

PROCEDURE PRINTERORSJ FORWARD;
PROCEDURE HEADING;
BEGIN
WHILE NEWSEGMENT = HERE DO;
END OF HEADING;
PROCEDURE PRINTLINE;
BEGIN
WHILE NEWSEGMENT = HERE DO;
END OF PRINTLINE;
PROCEDURE NEWCARDJ;
BEGIN
WHILE RESULT = 1CARD[*], ETC DO;
END OF NEWCARDJ;
PROCEDURE GEN(N, GEN, GEN2, GEN3) = GEN(T, GEN1, GEN2, GEN3) *;
PROCEDURE GEN(T, N, N, N, N, N, N); BEGIN
WHILE START = NUM DO;
END OF GEN;
BEGIN
IF GENT THEN
BEGIN
WHILE TEXT = TXT DO;
END ELSE
BEGIN
WHILE CH = TXT DO;
END;
END;
BEGIN
WHILE GEN1 DO;
BEGIN
WHILE N = N DO;
END
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
BEGIN
WHILE N = N DO;
REAL T1; T2; INTEGER F1, F2, LT, RT;
END OF CHECKTYPES;
PROCEDURE FILEPARAM ( DEFAULTFILE ); **** CHECKS THE FIRST PARAMETER
VALUE DEFAULTFILE; INTEGER DEFAULTFILE;**** TO SEE IF IT IS A FILE.
BEGIN
  DEFINE RESULTS = FILENAME & LPARFOUND #1;
END OF FILEPARAM;
REAL CURVAL; INTEGER CURLENGTH;
PROCEDURE CONSTANT ( CVAL, CTYPE );
REAL CVAL; INTEGER CTYPE;
BEGIN
  INTEGER TFORM; BOOLEAN SIGNED, NEGATIVE;
END OF CONSTANT;
$ $
ALPHA C* CX); { CURNAME1 & CURNAME2 MOVED TO 20205000 }
INTEGER LASTCHARPOS; { CURVAL, CURLENGTH MOVED TO 20872000 }
PROCEDURE INSYMBOL; {*** IDENTIFIES THE NEXT SYMBOL *****
BEGIN
  PROCEDURE NEXTCHAR; {*** GETS THE NEXT CHARACTER,
END OF NEXTCHAR;
$ $
DEFINE T1 = EXP #1 { USED AT 30178000
BEGIN
  DEFINE NENSEGMENT = HERE #;
END NEWSEGMENT;}
$ $
INTEGER EXPRLEVEL;
DEFINE PUTSYM(S) = PUTTEXT( (S) &1[41,516] ) #;
$ $
DEFINE PUTDUMMY = PUTTEXT("30000000") #;
$ $
PROCEDURE WRITEEXPR; {*** WRITE GENERATED ALGOL EXPRESSION
REAL SXJ, INTEGER TI, TX;
END OF WRITEEXPR;
PROCEDURE CHECKEXPR( LLLIM, ULLIM ); {*** WRITE CODE TO CHECK VALUE
VALUE LLLIM, ULLIM; INTEGER LLLIM, ULLIM;
BEGIN
  DEFINE CHECK = VALUE #1;
END OF CHECKEXPR;
INTEGER T1; T5; { USED ONCE EACH
T1=FIRSTWITHSYM; T5=LASTWITHSYM
FOR T1=TI STEP 1 UNTIL T5 DO PUTTEXT(WITHTAB(T1));
DEFINE T1 = T #1 { USED AT 40550000
$ $
PROCEDURE PARAMETER; {*** CHECK THAT THE FUNCTION HAS 1 PARAM.
BEGIN
  INSYMBOL;
  IF CURSY=LPAR
  THEN BEGIN
    PUTSYM("(") INSYMBOL EXPRESSION;
    IF TYPE=CHAR & FORM=NUMERIC THEN CURTYPE=INTTYPE;
    IF CURSY=RPAR THEN BEGIN ERROR(3); SKIP(RPAR) END
    $ $
PUTSYM("=") IF CURSZ=RPAR THEN INSYM; ELSE ERROR(3); % OR ERROR(58)
END OF PARAMETER;

BEGIN LABEL LABFOUND;
THISID=IOCLASS=CONST AND BOOLEAN(TTHISID, FORMAL) OR
THEN ASSIGNMENT ELSE

VALUE RECTABFIRSTADDR;
INTEGER RECTABFIRSTADDR_LASTADDR;

PROCEDURE TYPDECL(T TYPE, TSIZE);

BEGIN
INTEGER RECSINX, ARROSTRUCT,TX, SX, T, NJ REAL T1, T2, T3;
BOOLEAN FIRST, PACKED;

PROCEDURE SUBRANGE;

BEGIN
REAL VALX1, VALX2, T1;
INTEGER TYPE1, TYPE2;
CONSTANT(VALX1, TYPE1);
IF TYPE1=TYPE1ST THEN ERROR(11);
IF CURSZ=DOUBLEDOT THEN ERROR(53);
INSYM;
CONSTANT(VALX2, TYPE2);
IF TYPE1>0 AND TYPE2>0 THEN
IF TYPE1=TYPE2 THEN ERROR(11) ELSE
IF VALX1=VALX2 THEN ERROR(54);

IF (T1=TYPE1ST) THEN SYMBOLIC THEN T1=SUBTYPE;
NEWTYPE_T TYPE1=TYPEINDEX;
T1, SIZE=SIZE=1; T1, STRICT=0; T1, MAINTYPE=TYPE1;
TYPE1[TYPEINDEX1]=T1;
TYPE1[TYPEINDEX1]=VALX1; TYPE1[TYPEINDEX1]=VALX2;
END OF SUBRANGE;
DEFINE DEC = TYPE $1
DEFINE DEC = VAR $1
THEN BEGIN
DEFINE DEC = CODE $1
END OF SEGMENT FOR PROCEDURE DECLARATION$1

# PATCH 701 FOR PASCAL.XVI.0 CONTAINS 14 CARDS. REDUCE THRASHING BY ARRAY CUTS
# TO IMPROVE RUN TIME EFFICIENCY BY REDUCING ARRAY SIZES. THE MOST SIGNIFICANT
# CONTRIBUTION TO THE COMPILER'S THRASHERING BEHAVIOUR WAS THE EXCESSIVELY LARGE
# DATA ARRAYS. THIS PATCH SUCCEEDS IN DRAMATICALLY REDUCING THE CORE REQUIREMENT
# OF THE COMPILER BY MAKING MOST OF THE LARGE ARRAYS MUCH SMALLER WITHOUT
# IMPOSING UNREASONABLE RESTRICTIONS. IN PARTICULAR, THE THREE ARRAYS:
# NAMETAB1, NAMETAB2, NAMETAB3 WERE EACH [0150D0 011022], AND HAVE BEEN REDUCED
# TO [0130D0 011022]. THESE REDUCTIONS HAVE NOT PREVENTED THE COMPI LATION OF
# A LARGE PASCAL PROGRAM OF ABOUT 4000 LINES, NAMELY THE P4 PASCAL COMPIL E
# FROM ZURICH. IN FACT, PRIOR TO THE CHANGES INTRODUCED BY PATCHES 700 & 701,
# THE P4 PASCAL COMPILER TOOK 60 MINUTES ELAPSED TIME TO COMPIL E, WHICH WAS
# REDUCED TO 9 MINUTES BY THESE PATCHES, WHILE THE PROCESS TIME HAS REMAINED
# CONSTANT AT 9 MINUTES.
# *** NOTE THAT IF "MAXNAME" IS CHANGED THEN THERE ARE 7 DEFINES IN THE FILE
# PASCAL.PRELIM THAT MUST ALSO BE CHANGED.
# "MAXNAME" IS CHOSEN AS A PRIME NUMBER AS IT IS USED AS A MODULUS FOR A HASH
# FUNCTION. THE PASCAL IDENTIFIERS ARE TRANSLATED TO ALGOL NAMES USING LEVEL
# AND HASH INDEX, HENCE CHANGING "MAXNAME" CHANGES THE ALGOL NAMES FOR
# "INPUT", "OUTPUT", "PR25525".
# NILS A. OTTE, UNIVERSITY OF NATAL, DURBAN, AUG - NOV 1977.
# DEF I NEMENT MIXTABLES = 30 #
# MAX NAME = 307 #
# ONLY USED IN WITH STATEMENT TO TEST
# MAX CASES = 64 #
# MAX LABELS IN A CASE STATEMENT.
# MAX LABS = 50 #
# MAX NUMBER OF LABELS IN PROGRAM.
# MAX P A RAM S = 200 #
# MAX NUMBER OF PARAMETERS IN WHOLE PROGRAM.
# MAX TYPES = 250 #
# MAX NUMBER OF DIFFERENT TYPES.
# MAX CON S = 100 #
# MAX SIZE OF TABLE FOR CONSTANTS.
# MAX I NSYM S = 70 #
# MAX NUMBER OF SYMBOLS USED BY WITH STATEMENTS.
# MAX SYMS = 200 #
# MAX NUMBER OF SYMBOLS IN ONE EXPRESSION.
# LIST LENGTH = 100 #
# MAX LENGTH OF VAR & PARAM LISTS.
# MAX E X T FILES = 10 #
# MAX NUMBER OF EXTERNAL FILES.
# MAX FILES = 10 #
# MAX NUMBER OF FILES DEclared AT ONE TIME.
# MAX UNDECLARED POINTERS FOR DXD = 500 #
# MAX UNDECLARED POINTERS FOR DXD = 10 #
# MAX UNDECLARED POINTERS FOR DXD = 10 #
# PATCH 702 FOR PASCAL.XVI.0 CONTAINS 4 CARDS. BOOLEAN ARRAY "ERR" 120 TO 121
# TO EXTEND THE REDUCTIONS OF PATCH 701 TO THE BOOLEAN ARRAY "ERR" FOR NOTING
# THE SYNTAX ERRORS THAT HAVE OCCURRED. THIS PATCH COMpresses THE ARRAY FROM
# 120 WORDS TO 4 WORDs BY USING 32 BITS IN EACH WORD.
# IN ADDITION, THIS PATCH INSERTS THE ERROR COUNT ON THE LEFT OF THE LINE
# WHICH REPORTS THE SYNTAX ERRORS.
# NILS A. OTTE, UNIVERSITY OF NATAL, DURBAN, AUG - NOV 1977.
# ARRAY ERR[0131]; % HOLDS 128 BITS % RECORDS ERROR MESSAGES USED.
# 10156000
# ERR[0] = BOOLEAN & ERR[0], [01210] & ERR[1], [01451] #
# 10156010
# ERR[N][0] = BOOLEAN & ERR[N][0], [01451] #
# 10156110
# REPLACE POINTERS (ERR[L0][0] + 4) BY NUMBERS FOR 8 DIGITS.
# 20149000
# PATCH 703 FOR PASCAL.XVI.0 CONTAINS 6 CARDS. REDUCE THRASHERING BY SAVE CORE
# TO IMPROVE RUN TIME EFFICIENCY BY REDUCING NON-OVERLAPABLE AREAS.
# THIS PATCH REDUCES THE SAVE CORE REQUIREMENTS BY DECREASING THE FILE BLOCK
# SIZES AND ALSO THE NUMBER OF BUFFERS WITHOUT UNDULY RETARDING THE COMPILATION
# SPEED. THE SIZE OF THE DISK AREAS IS KEPT A MULTIPLE OF THE ORIGINAL BLOCK
$1 SIZE WHERE RELEVANT TO AVOID INCOMPATIBILITY PROBLEMS, COMPAREABLE REDUCTIONS
$1 IN BLOCK SIZES OF THE OBJECT PROGRAM ARE ALSO MADE.
$1 NILS A OTTE, UNIVERSITY OF NATAL, DURBAN, AUG - NOV 1977.
$1
FILE CARD "SOURCE" (1,10,30) X PASCAL SOURCE CODE INPUT FILE 10035000
FILE PASCALGO DISK SERIAL (20,300) (1,10,30,SAVE 0) X ALGOL CODE FILE 10037000
FILE XREFFILE DISK SERIAL (20,300) (1,3,30) X FOR CROSS REFERENCE 10137000
$1 IF REC_SIZE=1 OR REC_SIZE=10 THEN GENINT(30)
$1 8011900000
$1 GEN("x, SAVE", 6, 3)
$1 8012200000
$1 GEN("30", 4, 4)
$1 8012300000
$1 # PATCH 704 FOR PASCAL.X18.x0 HAS 8 CARDS. REDUCE OVERHEADS IN COPYING FILE
$1 TO REDUCE THE COMPILER'S OVERHEADS, FIRSTLY, THE ALGOL CODE FILE
$1 PASCRU/DISK IS RENAMED PASCAL/PRELUGE. ORIGINALLY, THE COMPILER COPIED
$1 THE PASCAL/PRELUGE FILE INTO THE GENERATED CODE FILE BEFORE STARTING TO
$1 TRANSLATE THE PASCAL PROGRAM. THIS PATCH SAVES THE 3 SECONDS OR SO REQUIRED
$1 FOR THIS BY SETTING THE "TAPE" OPTION FOR THE ALGOL COMPILER AND LABEL
$1 EQUATING THE TAPE FILE TO PASCAL/PRELUGE. THE OVERHEAD TO THE ALGOL COMPILER
$1 IS NEGLIGIBLE, THE ADVANTAGE IS EVEN GREATER IF THE PROGRAM FAILS TO
$1 COMPIL SYNTAX FREE. THE FILE PASCAL/PRELUGE IS NO LONGER REFERENCED
$1 DIRECTLY IN THE PASCAL COMPILER.
$1 SEE PATCH 711, THIS NEEDS PATCH 705.
$1 NILS A OTTE, UNIVERSITY OF NATAL, DURBAN, AUG - NOV 1977.
$1
$1 ERRORS (15:"ERRORS DETECTED "20("#")",
$1 10188000
$1 ALIST (""$ LIST "",
$1 10189000
$1 MERGE ("$ SET TAPE "$ RE "
$1 "/$ RESET TAPE" 73"99000000"
$1 10190000
$1 TERMMESS ("**** Compilation Terminated;"
$1 10192000
$1 WRITE(PASCALGO, MERGE) % ALGOL MUST COMPILE PRELUGE FIRST
$1 90022000
$1 $ SET VOID
$1 90023000
$1 $ POP VOID
$1 90032000
$1 "j ALGOL FILE TAPE= PASCAL/PRELUGE SERIALj ALGOL FILE CARD="90119000
$1 # PATCH 705 FOR PASCAL.X18.x0 CONTAINS 21 CARDS. GENERATE A BETTER ZIP
$1 THIS PATCH TIDIES UP THE CODE THAT GENERATES THE ZIP TO PASS CONTROL TO THE
$1 COMPATIBLE ALGOL COMPILER.
$1 NILS A OTTE, UNIVERSITY OF NATAL, DURBAN, AUG - NOV 1977.
$1
$1 $ PROGNAME = IF CURLENGTH < 7
$1 90013000
$1 THEN ""&CURNAME(1416xCURLENGTH+6xCURLENGTH)
$1 90042000
$1 ELSE CURNAME2(14135316)
$1 90042020
$1 $ ARRAY ZIPARRAY(01161)
$1 90092000
$1 DEFINE PROGNAME = 13 #
$1 90095000
$1 PALGONAME = 14 #
$1 90096000
$1 P(LIBRARY) = 15 #
$1 90097000
$1 PUSER = 16 #
$1 90097000
$1 P(P1) = $setter(ZIPARRAY(P1)) FOR 7 #
$1 90098000
$1 $ SET VOID
$1 90104000
$1 $ POP VOID
$1 90109000
$1 $ ZIPARRAY[PRGNAME] = PRGNAME
$1 90112000
$1 ZIPARRAY[ PALGONAME ] = PALGONAME
$1 90112000
$1 ZIPARRAY[P(LIBRARY)] = IF SAVEFACTOR > 0 THEN "LIBRARY" ELSE
$1 90113000
$1 IF SAVEFACTOR < 0 THEN "SYNTAX" ELSE " & RUN "
$1 90114000
$1 ZIPARRAY[PUSER] = USER
$1 90115000
$1 REPLACE $setter(ZIPARRAY[1]) BY "CC COMPIL 
$1 90116000
$1 P(PRGRAMNAME), "/", P(PUSER)
$1 90117000
$1 " XLGAL ", P(P(LIBRARY))
$1 90118000
$1 " ALGOL FILE TAPE= PASCAL/PRELUGE SERIALj ALGOL FILE CARD="90119000
$1 P(PALGONAME), "/", P(PUSER), " SERIALj END;"
$1 90120000
$1
SET VOID 90121000
SET POPOD 90128000
PATCH 708 FOR PASCAL.XVI.O CONTAINS 25 CARDS. LINE PRINT FILE MAY BE DISK
TO ENABLE THE COMPILER'S PRINT FILE TO BE LABEL EQUATED TO DISK AS FOR OTHER
BS700 COMPILERS, IN PARTICULAR, THIS PATCH CHANGES THE NAME TO LINE TO BE
CONSISTENT WITH ALL THE SYSTEM COMPILERS, THE ABILITY TO LABEL EQUATE FILE
"LINE" TO DISK IS NECESSARY IF THE COMPILER IS TO BE USED FROM A TERMINAL,
NOTE THAT A BLOCKED FILE SHOULD NOT HAVE VARIABLE LENGTH RECORDS IF IT IS
TO BE LABEL EQUATED TO A PRINTER. IF LESS THAN THE MAX NUMBER OF WORDS PER
RECORD IS WRITTEN, THE BALANCE OF THE RECORD REMAINS UNCHANGED FROM WHAT WAS
LAST IN THE FILE BUFFER, SO THAT ON BEING PRINTED, "GARBAGE" APPEARS AT THE
END OF SUCH LINES.
SAVE FILE OUT LINE DISK SERIAL (2012100) (1.17.90,SAVE 1) % PRINT FILE 10036000
% AVOID BLOCKING RECORDS OF VARIABLE LENGTH 10036001
ARRAY ICARD, ALGOLCARD(0199), LINES, XLINE(01163)
% AVOID BLOCKING VARIABLE LENGTH RECORDS 10130001
ARRAY HEADTEXT, ERRLINE(01163)
WRITE( LINES(01)+17,LINES(*)+)
WRITE( LINES(01)+17,XLINE(*)+)
WRITE(LINES, 17,LINES(*))
WRITE(LINES, 17,BFRELINE(*))
LINESP = POINTER(LINES(1))
REPLACE LINESP = 8 " " FOR 17 WORDS;
REPLACE XLINESP = 8 BY LINESP = 8 FOR 17 WORDS;
REPLACE POINTER(ERRLINE(*)) BY "**** " LINESP FOR 16 WORDS;
REPLACE ALGOLCARD PT BY LINESPP FOR 9 WORDS;
REPLACE POINTER(HEADTEXT(*)) BY LINESP FOR 10 WORDS, "PAGE 1 ",
LINESP FOR 6 WORDS;
WRITE(LINES, 17, XREFLINE(*))
LOCK LINES, * ); % & CRUNCH
WRITE(LINES, 17, XREFLINE(E)) LINESPT = LINESPT + 1
WRITE(LINES, 17, XREFLINE()) LINESPT = LINESPT + 1
WRITE(LINES, TERMMESS) 90084000
WRITE(LINES, NERRORS) 90111000
WRITE(LINES, ERRORS) 99100000
WRITE(LINES, ERRORMESS(1)) 91120000
WRITE(LINES, ERRORMESS(2) = 0)
$ PATCH 709 FOR PASCAL.XVI.O CONTAINS 17 CARDS. NO PRINT IF NO LIST & NUMRERS
$ TO OPEN THE PRINT FILE ONLY IF THE LIST OPTION IS SET OR IF SYNTAX ERRORS
$ ARE DETECTED. IF THE FIRST CARD IN THE PASCAL SOURCE RESETS THE LIST OPTION
$ (*L = 0) AND NO SYNTAX ERRORS ARE DETECTED, THEN THE PRINT FILE WILL NOT BE
$ CREATED SEVEN FOR THE HEADING) AS FOR OTHER COMPILERS, IN PARTICULAR, THIS
$ IMPLEMENTATION DOES NOT REQUIRE A TEST PRIOR TO PRINTING EACH LINE TO
$ DETERMINE WHETHER A HEADING HAS BEEN PRINTED. IT ONLY DOES THIS TEST WHEN
$ THE LIST OPTION IS SET AFTER THE FIRST CARD OR EXPLICITLY THEREAFTER, OR
$ IN THE "PRINTERROR" ROUTINE.
$ IF PAGECNT = 1 THEN WRITE(LINES(01)+17,HEADTEXT(*)) ELSE
WRITE(LINES(PAGE))
WRITE(LINES(0L1)+17,HEADTEXT(*))
IF NOT LISTOPTION THEN
BEGIN IF PAGECNT = 0 THEN HEADING; PRINTLINE END;
REPLACE POINTER(HEADTEXT(*)) = 45 BY TEXTPNT + 5 FOR 2; "/",
TEXTPNT + 1 FOR 2; " ", TEXTPNT + 3 FOR 2;
20330000
NEWCARDJ; LISTOPTION=CHECKOPTION=TRUE; % DEFAULT
INSYMBOL; % ANALYZING FIRST CARD MAY CHANGE DEFAULT LIST OPTN
IF LISTOPTION AND PAGECNT# THEN HEADING; % ON FIRST PAGE;
  IF LISTOPTION THEN IF PAGECNT=0 THEN HEADING; % ON FIRST PAGE
C I= " ";
  % TO INITIALIZE "INSYMBOL"
INITIALIZE; % COMPILED TABLES; NEWCARDJ INSYMBOL
$ 90034000
$ 90035000
$ 90036000
$ 90088000
THEN BEGIN WRITE( LINE(1)) ; WRITE( LINE(DBL) ) END;
$ 90089000
$ 90100000
$ 90110000
$# PATCH 710 FOR PASCAL.XVI.O CONTAINS 4 CARDS, NO OVERPRINTING WITH BLANK LINE
$ 1 TO PREVENT OVERPRINTING WITH BLANK LINES, IF THE OPTION FOR "BOLDFACE"
$ 1 PRINTING OF RESERVED WORDS IS SET (+AR=+K) THEN EACH LINE IS CONSTRUCTED BY
$ 1 2 OVERPRINTS FOR THE RESERVED WORDS ONLY, THEN ONE PRINT OF THE FULL TEXT.
$ 1 THE AIM OF THIS PATCH IS TO SKIP THE OVERPRINTING FOR ALL THOSE LINES IN
$ 1 WHICH NO RESERVED WORDS OCCUR.
$ 1 NILS A OTTE, UNIVERSITY OF NATAL, DURBAN, AUG = NOV 1977.
$ 1
$ 1 DEFINE RESWORDPRESENT = RESWORDOPTION,(111) ;
  IF REAL(RESWORDOPTION) = 3 THEN % RESERVED WORD IS PRESENT
  RESWORDOPTION = RESWORDOPTION AND TRUE; % RESET RESWORDPRESENT
  RESWORDOPTION = BOOLEAN(3); % SET RESWORDPRESENT BIT
  RESWORDPRESENT = 30179000
$ 1# PATCH 711 FOR PASCAL.XVI.O CONTAINS 10 CARDS, PASCALXXX.USERCODE UNIQUE NAME
$ 1 TO GENERATE A UNIQUE FILE NAME IN THE DISK DIRECTORY, THIS PATCH CHANGES THE
$ 1 METHOD FOR GENERATING A UNIQUE FILE NAME FOR THE ALGOL SOURCE CODE OUTPUT OF
$ 1 THE COMPILER. FORMERLY, THIS WAS DONE USING THE TIME FUNCTION TO OBTAIN
$ 1 SOME RANDOM DIGITS. THE METHOD USED IN PATCH/MERGE IS ADOPTED HERE, NAMELY
$ 1 STARTING WITH THE PREFIX ("MFID") "PASCALXXX", A SEARCH IS PERFORMED TO DETERMINE
$ 1 WHETHER SUCH A FILE NAME IS ALREADY CATALOGUED. IF SO, 1 IS ADDED AND THE
$ 1 SEARCH REPEATED. IN ADDITION, THE FILE IS CREATED WITH A SAVE FACTOR
$ 1 (RETENTION PERIOD) OF ZERO DAYS SO THAT A HALT-LOAD WILL REMOVE THE FILE
$ 1 AUTOMATICALLY.
$ 1 SEE PATCH 704.
$ 1 NILS A OTTE, UNIVERSITY OF NATAL, DURBAN, AUG = NOV 1977.
$ 1
$ 1 PROCEDURE SEARCHMIDIRECTORY(FNC, A); FILE FJ, ARRAY A0(J);
  SEARCH(FNC, A[+1]) ; % END OF SEARCHMIDIRECTORY
  SEARCHMIDIRECTORY(FNC, A[*])
  CHARPNT = POINTER(CH0(J)+*7); CH0(J) = " ";
  CH0(J) = "PASCALXXX"; CHARPNT = POINTER(CH0(J)+*5);
  PASCALGOL.FID = USER.* TIME[*+1];
  DO BEGIN C1=C+1; REPLACE CHARPNT BY C FOR 3 DIGITS;
  PASCALGOL.WFID = ALGOLNAME = CH0(J);
  SEARCHMIDIRECTORY PASCALGOL.LINES[*+1]) END UNTIL LINES(J)=+1; % FILE NOT ON DISK
  $# PATCH 712 FOR PASCAL.XVI.O CONTAINS 2 CARDS, MARK PROCEDURE LEVELS IN MARGIN
  $ 1 PATCH TO MARK THE START AND END OF PROCEDURES AND FUNCTIONS BY ANNOTATING THE
  $ 1 MARGIN WITH THE SYMBOLS "+P" & "-P" FOLLOWED BY THE LEVEL NUMBER.
  $ 1 NILS A OTTE, UNIVERSITY OF NATAL, DURBAN, AUG = NOV 1977.
  $ 1
  $ 1 MARGIN("+P", CURLEVEL) ; % MARK PROCEDURE LEVEL
  $ 1 MARGIN("-P", CURLEVEL) ; % MARK END OF PROCEDURE
  $ 1
  $ 1# PATCH 713 FOR PASCAL.XVI.O CONTAINS 14 CARDS, CORRECTS ERROR MESSAGES ETC.
  $ 1 CORRECTS THE DOUBLE "NO ERRORS" MESSAGE AND THE OUTPUT OF HEADINGS
  $ 1 WHEN LI IS SET AFTER L-.
  $ 1 ALSO CORRECTS THE SCANNING PROBLEM WHEN COMPILER OPTIONS ARE INCORRECT.
  $ 1 DAVID A COOPER, HERIOT-WATT UNIVERSITY ...... AUGUST 1978
ERROR102MESS("/"102 *** WARNING ONLY, ILLEGAL COMPILER OPTION.")10188750
* *
IF ERRNUM=100 OR ERRNUM=102
THEN NUMERRS = NUMERRS + 1
* ERROR NUMBER 102 IS ONLY AN ILLEGAL
* DOLLAR OPTION WARNING &
* ERROR NUMBER 100 ALONE SHOULD NOT
ELSE LISTOPTION = C="" ELSE
END
ELSE ERROR(102);
IF ERR(102) THEN
WRITE(LINE,ERROR102MESS));
# PATCH 800 FOR PASCAL.XVI.O CONTAINS 10 CARDS.
$ TO REMOVE CONFLICTS BETWEEN HERITI=MATT & NATAL EXISTING PATCHES.
$ MAXSYMS = 800#; % MAX NUMBER OF SYMBOLS IN ONE EXPRESSION, 10052000
MAXPNTS = 25#; % MAX NUMBER OF UNDECLARED POINTERSFORWD), 10056000
DEFINE ERRERR1) = BOOLEAN(0&ERRRI(111612);{0;{(ERRR1).145}111});
INTEGER EXPRLEVEL, EXPINVARCNT; % 10156100
INTEGER INDEX, CTYPER, NUMFORWARDS,TX, I ; % 4001800
INTEGER PROGNAMELENGTH; % 80403000
INTEGER PROGNAMELENGTH; % 90013900
IF ERR(100) % 90090000
"ALGOL FILE TAPE=PASCRUN/DISK SERIAL; ALGOL FILE CARD=",
P(PALGOLNAME)="/"P(PUSER)="/ SERIAL; % 90119000
"XALGOL STACK = 2048) END,); % 90120000
# PATCH 996 FOR PASCAL.XVI.O CONTAINS 10 CARDS. INSERT PAGE THROWS AT DESIRED
# PATCH TO INSERT PAGE THROWS AT DESIRED POINTS IN THE SOURCE TO PRODUCE A
# NICELY LAYED OUT LISTING.
$ PAGE 19000000
$ PAGE 20290000
$ PAGE 29000000
$ PAGE 39000000
$ PAGE 49000000
$ PAGE 59000000
$ PAGE 69000000
$ PAGE 79000000
$ PAGE 89000000
$ PAGE 90070999
$ NILS A OTTE, UNIVERSITY OF NATAL, DURBAN, AUG - NOV 1977.
# PATCH 999 FOR PASCAL.XVI.O CONTAINS 1 CARDS, VERSION NUMBER.
$ DEFINE EDITION = "4,4"#J/AUGUST 1978...DAVID A COOPER...
# LABEL 000000000OLINE 00178299CC EX 0/R;COMMON=1;FILE LINE=LINE PRINT;FILE S=PATCHES/PASCALJEND* 0 /R