```
001:  (*$U+ COMPILE TO 72 COLUMNS ONLY. *)                                    PASCREF     2
002:  (*$A+ USE ASCII CHARACTER SET. *)                                       PASCREF     3
003:  (*$T- COMPILE WITHOUT RUN-TIME TESTS. *)                                PASCREF     4
004:  (*$P0 COMPILE WITH ABSOLUTELY NO PMD INFO. *)                           PASCREF     5
005:  (*$R- DO NOT REDUCE FIELD LENGTH. *)                                    PASCREF     6
006:  (*$W20000B SET MINIMUM WORKSPACE SIZE. *)                               PASCREF     7
007:  (*$B8 USE LARGE FILE BUFFERS. *)                                        PASCREF     8
008:                                                                          PASCREF     9
009:                                                                          PASCREF    10
010:  PROGRAM PASCREF(INPUT,OUTPUT);                                          PASCREF    11
011:                                                                          PASCREF    12
012:  (* PASCREF - CROSS REFERENCE PASCAL PROGRAMS.                      *)   PASCREF    13
013:  (*                                                                 *)   PASCREF    14
014:  (* ORIGINAL PROGRAM (NAMED XREF).                                 *)   PASCREF    15
015:  (*    - N. WIRTH 71/01/15, 74/05/07, 75/07/02, 76/02/10.          *)   PASCREF    16
016:  (*      (SEE CHAPTER 4 IN "ALGORITHMS + DATA STRUCTURES = PROGRAMS") *)  PASCREF    17
017:  (* USE BETTER SORT, CASE STATEMENT IN SCANNER, PROCEDURE MAP,      *)   PASCREF    18
018:  (* CONTROL STATEMENT PROCESSING.                                  *)   PASCREF    19
019:  (*    - A. MICKEL 75/12/08.                                       *)   PASCREF    20
020:  (* USE RING STRUCTURE FOR REFERENCES.                             *)   PASCREF    21
021:  (*    - R. CICHELLI 76/11/14.                                     *)   PASCREF    22
022:  (* PROCESS LINE NUMBERS.                                          *)   PASCREF    23
023:  (*    - D. LALIBERTE 78/03/15.                                    *)   PASCREF    24
024:  (* PROCESS COMPILER TITLES, DIFFERENT PRINT DENSITIES,            *)   PASCREF    25
025:  (* USE VALUE PART, SORT CORRECTLY, ADD NESTING LEVELS.            *)   PASCREF    26
026:  (*   - J. STRAIT 78/12/28.                                        *)   PASCREF    27
027:  (* FIX NESTING-LEVEL INDICATORS, HANDLE PERCENT IN 64-CHAR SET.   *)   PASCREF1    1
028:  (*   - J.F. MINER 1982-01-06                                      *)   PASCREF1    2
029:  (*                                                                 *)   PASCREF    28
030:                                                                          PASCREF    29
031:  LABEL 99;                                                              PASCREF    30
032:                                                                          PASCREF    31
033:  CONST P = 1499;     (*SIZE OF HASH TABLE*)                             PASCREF    32
034:    NK = 33;          (*NO. OF KEYWORDS*)                                PASCREF    33
035:    KLN = 10;         (*KEYLENGTH*)                                      PASCREF    34
036:    LPPG6 = 63;       (*LINES PER PAGE AT 6 LINES PER INCH*)             PASCREF    35
037:    LPPG8 = 84;       (*LINES PER PAGE AT 8 LINES PER INCH*)             PASCREF    36
038:    LTPG6 = 3;        (*LINES AT TOP OF PAGE AT 6 LINES PER INCH*)       PASCREF    37
039:    LTPG8 = 4;        (*LINES AT TOP OF PAGE AT 8 LINES PER INCH*)       PASCREF    38
040:    LLMIN = 72;       (*MINIMUM LINE LENGTH*)                            PASCREF    39
041:    LLINMAX = 120;    (*MAXIMUM INPUT LINE LENGTH*)                      PASCREF    40
042:    LLOUTMAX = 132;   (*MAXIMUM OUTPUT LINE LENGTH*)                     PASCREF    41
043:    MAXN = 100000;    (*MAX NO. OF LINES*)                               PASCREF    42
044:    DGPN =  6;        (*NO. OF DIGITS PER NUMBER*)                       PASCREF    43
045:    LITL = 2;         (*NUMBER OF LINES IN COMPILER TITLE*)              PASCREF    44
046:    ADDRWIDTH = 6;    (*NUMBER OF DIGITS IN CODE ADDRESS*)               PASCREF    45
047:    STARS = ' *****';                                                    PASCREF    46
048:    C00 = COL;        (*CHR(00B)*)                                       PASCREF    47
049:                                                                         PASCREF    48
050:  TYPE INDEX = 0..P;                                                     PASCREF    49
051:    ALFA = PACKED ARRAY [1..KLN] OF CHAR;                                PASCREF    50
052:    REF = ^ITEM;                                                         PASCREF    51
053:    WORD = RECORD KEY: ALFA;                                             PASCREF    52
054:             LAST: REF;                                                  PASCREF    53
055:           END ;                                                         PASCREF    54
056:    ITEM = PACKED RECORD                                                 PASCREF    55
057:             LNO: 0..MAXN;                                               PASCREF    56
058:             NEXT: REF                                                   PASCREF    57
059:           END ;                                                         PASCREF    58
060:    NESTKIND = (NESTBEGIN, NESTOPEN, NESTCLOSE, NESTCLEAR, NESTNULL);    PASCREF1    3
061:    PROCREF = ^PROC;  (*PROCEDURE OR FUNCTION REFERENCE*)                PASCREF    59
062:    PROC = PACKED RECORD                                                 PASCREF    60
063:             NAME: ALFA;                                                 PASCREF    61
064:             LNO: 0..MAXN;                                               PASCREF    62
065:             NEXT: PROCREF                                               PASCREF    63
```

```
066:          END ;                                                                  PASCREF    64
067:                                                                                 PASCREF    65
068:  VAR I: INDEX;                                                                  PASCREF    66
069:    K: INTEGER;                                                                  PASCREF    67
070:    M: INTEGER;       (*NO. OF LINES ON PAGE*)                                   PASCREF    68
071:    N: INTEGER;       (*NO. OF LINES INPUT*)                                     PASCREF    69
072:    LPPG: INTEGER;    (*NUMBER OF LINES PER PAGE*)                               PASCREF    70
073:    LTPG: INTEGER;    (*NUMBER OF BLANK LINES AT TOP OF PAGE*)                   PASCREF    71
074:    LN: INTEGER;      (*CURRENT LINE NUMBER*)                                    PASCREF    72
075:    LLOUT: INTEGER;   (*LINE LENGTH FOR OUTPUT*)                                 PASCREF    73
076:    LLIN: INTEGER;    (*LINE LENGTH FOR INPUT*)                                  PASCREF    74
077:    CCOUNT: INTEGER;  (*CHARACTER COUNT IN LINE*)                                PASCREF    75
078:    NOPL: INTEGER;    (*NO. OF LINE-NUMBERS PER LINE*)                           PASCREF    76
079:    EMPTY: ALFA;                                                                 PASCREF    77
080:    ID: RECORD CASE BOOLEAN OF                                                   PASCREF    78
081:                FALSE: (A: ALFA);                                                PASCREF    79
082:                TRUE:  (ORD: INTEGER)                                            PASCREF    80
083:        END ;                                                                    PASCREF    81
084:    T: ARRAY [INDEX] OF WORD;    (*HASH TABLE*)                                  PASCREF    82
085:    KEYINDEX: 0..NK;                                                             PASCREF    83
086:    KEY: ARRAY[0..NK] OF RECORD                                                  PASCREF    84
087:                            K: ALFA;                                             PASCREF    85
088:                            NESTCHANGE: NESTKIND                                 PASCREF1    4
089:                          END;                                                   PASCREF    87
090:    INBODY: BOOLEAN;                                                             PASCREF    89
091:    NESTING, NESTINGATBOL: 0..MAXINT;                                            PASCREF1    5
092:    NESTCOUNT: 0..LLINMAX;  (* HIGHEST ELEMENT OF NEST BEING USED *)             PASCREF1    6
093:    NEST: ARRAY[1..LLINMAX] OF NESTBEGIN..NESTCLEAR;                             PASCREF1    7
094:    PROCORFUNC,                                                                  PASCREF    92
095:    PAGINATING,                                                                  PASCREF    93
096:    COMPILERLISTING,                                                             PASCREF    94
097:    LINENUMBERS: BOOLEAN;                                                        PASCREF    95
098:    FIRSTPROC,                                                                   PASCREF    96
099:    PROCPTR: PROCREF; (*POINTERS TO CHAIN OF PROCEDURES*)                        PASCREF    97
100:                                                                                 PASCREF    98
101:  VALUE EMPTY = (KLN OF C00);                                                    PASCREF    99
102:     T = (P OF ((KLN OF C00), NIL), ((KLN OF C00), NIL));                        PASCREF   100
103:     KEY = (((                                      10 OF C00),NESTNULL ),     PASCREF1    8
104:           (('A','N','D',                            7 OF C00),NESTNULL ),     PASCREF1    9
105:           (('A','R','R','A','Y',                    5 OF C00),NESTNULL ),     PASCREF1   10
106:           (('B','E','G','I','N',                    5 OF C00),NESTBEGIN),     PASCREF1   11
107:           (('C','A','S','E',                        6 OF C00),NESTOPEN ),     PASCREF1   12
108:           (('C','O','N','S','T',                    5 OF C00),NESTNULL ),     PASCREF1   13
109:           (('D','I','V',                            7 OF C00),NESTNULL ),     PASCREF1   14
110:           (('D','O',                                8 OF C00),NESTNULL ),     PASCREF1   15
111:           (('D','O','W','N','T','O',                4 OF C00),NESTNULL ),     PASCREF1   16
112:           (('E','L','S','E',                        6 OF C00),NESTNULL ),     PASCREF1   17
113:           (('E','N','D',                            7 OF C00),NESTCLOSE),     PASCREF1   18
114:           (('F','I','L','E',                        6 OF C00),NESTNULL ),     PASCREF1   19
115:           (('F','O','R',                            7 OF C00),NESTNULL ),     PASCREF1   20
116:           (('F','U','N','C','T','I','O','N',        2 OF C00),NESTCLEAR),     PASCREF1   21
117:           (('I','F',                                8 OF C00),NESTNULL ),     PASCREF1   22
118:           (('I','N',                                8 OF C00),NESTNULL ),     PASCREF1   23
119:           (('M','O','D',                            7 OF C00),NESTNULL ),     PASCREF1   24
120:           (('N','I','L',                            7 OF C00),NESTNULL ),     PASCREF1   25
121:           (('N','O','T',                            7 OF C00),NESTNULL ),     PASCREF1   26
122:           (('O','F',                                8 OF C00),NESTNULL ),     PASCREF1   27
123:           (('O','R',                                8 OF C00),NESTNULL ),     PASCREF1   28
124:           (('P','A','C','K','E','D',                4 OF C00),NESTNULL ),     PASCREF1   29
125:           (('P','R','O','C','E','D','U','R','E',    1 OF C00),NESTCLEAR),     PASCREF1   30
126:           (('P','R','O','G','R','A','M',            3 OF C00),NESTCLEAR),     PASCREF1   31
127:           (('R','E','C','O','R','D',                4 OF C00),NESTNULL ),     PASCREF1   32
128:           (('R','E','P','E','A','T',                4 OF C00),NESTOPEN ),     PASCREF1   33
129:           (('S','E','T',                            7 OF C00),NESTNULL ),     PASCREF1   34
130:           (('T','H','E','N',                        6 OF C00),NESTNULL ),     PASCREF1   35
```

```
131:           (('T','O',                          8 OF C00),NESTNULL ),  PASCREF1   36
132:           (('T','Y','P','E',                  6 OF C00),NESTNULL ),  PASCREF1   37
133:           (('U','N','T','I','L',              5 OF C00),NESTCLOSE),  PASCREF1   38
134:           (('V','A','R',                      7 OF C00),NESTNULL ),  PASCREF1   39
135:           (('W','H','I','L','E',              5 OF C00),NESTNULL ),  PASCREF1   40
136:           (('W','I','T','H',                  6 OF C00),NESTNULL )); PASCREF1   41
137:     N = 0;                                                             PASCREF   138
138:     M = 0;                                                             PASCREF   139
139:     LLIN = LLINMAX;                                                    PASCREF   140
140:     LLOUT = LLOUTMAX;                                                  PASCREF   141
141:     PROCORFUNC = TRUE;                                                 PASCREF   142
142:     LPPG = LPPG6;                                                      PASCREF   143
143:     LTPG = LTPG6;                                                      PASCREF   144
144:     COMPILERLISTING = TRUE;                                            PASCREF   145
145:     LINENUMBERS = TRUE;                                                PASCREF   146
146:     NESTING = 0;                                                       PASCREF   147
147:     INBODY = FALSE;                                                    PASCREF   148
148:                                                                        PASCREF   149
149:  PROCEDURE CLASSIFY;                                                  PASCREF   150
150:    VAR I,J,K: INTEGER;                                                PASCREF   151
151:  BEGIN I := 1; J := NK;   (*BINARY SEARCH*)                           PASCREF   152
152:    KEYINDEX := 0;                                                     PASCREF   153
153:    REPEAT K := (I+J) DIV 2;                                           PASCREF   154
154:      IF KEY[K].K <= ID.A THEN I := K+1 ELSE J := K-1                  PASCREF   155
155:    UNTIL I > J;                                                       PASCREF   156
156:    IF KEY[J].K = ID.A THEN KEYINDEX := J                              PASCREF   157
157:  END (*CLASSIFY*) ;                                                   PASCREF   158
158:                                                                       PASCREF   159
159:  PROCEDURE BEGINLINE;                                                PASCREF   160
160:   VAR I: INTEGER;                                                     PASCREF   161
161:  BEGIN                                                               PASCREF   162
162:    IF PAGINATING THEN                                                 PASCREF   163
163:      BEGIN                                                            PASCREF   164
164:        IF M >= LPPG THEN                                              PASCREF   165
165:          BEGIN WRITE('1');                                            PASCREF   166
166:            FOR I := 1 TO LTPG DO WRITELN;                             PASCREF   167
167:            M := LTPG                                                  PASCREF   168
168:          END;                                                         PASCREF   169
169:        M := M + 1                                                     PASCREF   170
170:      END                                                              PASCREF   171
171:  END (*BEGINLINE*) ;                                                  PASCREF   172
172:                                                                       PASCREF   173
173:  PROCEDURE ADVANCE;                                                  PASCREF   174
174:  BEGIN                                                               PASCREF   175
175:    IF NOT EOLN THEN                                                   PASCREF   176
176:      BEGIN WRITE(INPUT^); GET(INPUT);                                 PASCREF   177
177:        CCOUNT := CCOUNT + 1;                                          PASCREF   178
178:        IF CCOUNT = LLIN THEN                                          PASCREF   179
179:          WHILE NOT EOLN(INPUT) DO                                     PASCREF   180
180:            BEGIN WRITE(INPUT^); GET(INPUT); CCOUNT := CCOUNT + 1 END  PASCREF   181
181:      END                                                              PASCREF   182
182:  END (*ADVANCE*);                                                     PASCREF   183
183:                                                                       PASCREF   184
184:  PROCEDURE SKIP(J: INTEGER);                                          PASCREF   185
185:  BEGIN                                                               PASCREF   186
186:    IF M+J >= LPPG THEN M := LPPG                                      PASCREF   187
187:    ELSE                                                               PASCREF   188
188:      FOR J := 1 TO J DO BEGIN BEGINLINE; WRITELN END                  PASCREF   189
189:  END (*SKIP*) ;                                                       PASCREF   190
190:                                                                       PASCREF   191
191:  PROCEDURE NEWLINE;                                                  PASCREF   192
192:                                                                       PASCREF   193
193:    PROCEDURE COPYCOMPILERTITLE;                                       PASCREF   194
194:      VAR I: INTEGER;                                                  PASCREF   195
195:    BEGIN                                                              PASCREF   196
```

```
196:      I := 0;                                                                PASCREF   197
197:      WHILE (I < LITL) AND NOT EOF DO                                        PASCREF   198
198:        BEGIN I := I + 1;                                                    PASCREF   199
199:          BEGINLINE;                                                       PASCREF   200
200:          WHILE NOT EOLN(INPUT) DO                                         PASCREF   201
201:            ADVANCE;                                                       PASCREF   202
202:          READLN; WRITELN                                                  PASCREF   203
203:        END                                                                PASCREF   204
204:    END (*COPYCOMPILERTITLE*) ;                                            PASCREF   205
205:                                                                           PASCREF   206
206:  BEGIN CCOUNT := 0;                                                       PASCREF   207
207:    NESTCOUNT := 0;  NESTINGATBOL := NESTING;                              PASCREF1   42
208:    IF N < MAXN THEN                                                       PASCREF   209
209:    BEGIN BEGINLINE;  N := N + 1;                                          PASCREF   210
210:      IF COMPILERLISTING THEN                                              PASCREF   211
211:        BEGIN IF INPUT^ = '1' THEN COPYCOMPILERTITLE;                      PASCREF   212
212:          ADVANCE;                                                         PASCREF   213
213:          IF NOT (INPUT^ IN ['0'..'7']) THEN  (* ERRORS *)                 PASCREF   214
214:            WHILE NOT EOLN DO                                              PASCREF   215
215:              ADVANCE                                                      PASCREF   216
216:          ELSE                                                             PASCREF   217
217:            BEGIN                                                          PASCREF   218
218:              FOR I := 1 TO ADDRWIDTH + 1 DO                               PASCREF   219
219:                ADVANCE;                                                   PASCREF   220
220:              WHILE (INPUT^ = ' ') AND NOT EOLN DO                         PASCREF   221
221:                ADVANCE                                                    PASCREF   222
222:            END                                                            PASCREF   223
223:        END                                                                PASCREF   224
224:      ELSE WRITE(' ');                                                     PASCREF   225
225:      IF LINENUMBERS THEN                                                  PASCREF   226
226:        BEGIN LN := 0;                                                     PASCREF   227
227:          WHILE INPUT^ IN ['0'..'9'] DO                                    PASCREF   228
228:            BEGIN LN := 10*LN MOD 100000 + ORD(INPUT^) - ORD('0');         PASCREF   229
229:              ADVANCE                                                      PASCREF   230
230:            END;                                                           PASCREF   231
231:          ADVANCE;                                                         PASCREF   232
232:          IF COMPILERLISTING THEN CCOUNT := 0                              PASCREF   233
233:        END                                                                PASCREF   234
234:      ELSE BEGIN                                                           PASCREF   235
235:        LN := N;  WRITE(LN:6, ' ')                                         PASCREF   236
236:        END                                                                PASCREF   237
237:      END                                                                  PASCREF   238
238:    ELSE BEGIN                                                             PASCREF   239
239:      WRITELN(STARS, ' TEXT TOO LONG', STARS);                             PASCREF   240
240:      GOTO 99                                                              PASCREF   241
241:      END                                                                  PASCREF   242
242:  END (*NEWLINE*) ;                                                       PASCREF   243
243:                                                                           PASCREF   244
244:  PROCEDURE SEARCH;   (*MODULO P HASH SEARCH*)                             PASCREF   245
245:    VAR H,D: INDEX;                                                        PASCREF   246
246:        X,Y: REF; F: BOOLEAN;                                              PASCREF   247
247:  BEGIN H := (ID.ORD DIV 4096) MOD P;                                      PASCREF   248
248:    F := FALSE; D := 1;                                                    PASCREF   249
249:    NEW(X); X^.LNO := LN;                                                  PASCREF   250
250:    REPEAT                                                                 PASCREF   251
251:      IF T[H].KEY = ID.A THEN                                              PASCREF   252
252:      BEGIN (*FOUND*) F := TRUE;                                           PASCREF   253
253:        Y := T[H].LAST; X^.NEXT := Y^.NEXT;                                PASCREF   254
254:        Y^.NEXT := X; T[H].LAST := X                                       PASCREF   255
255:      END ELSE                                                             PASCREF   256
256:      IF T[H].KEY = EMPTY THEN                                             PASCREF   257
257:      BEGIN (*NEW ENTRY*) F := TRUE;                                       PASCREF   258
258:        T[H].KEY := ID.A;                                                  PASCREF   259
259:        T[H].LAST := X; X^.NEXT := X                                       PASCREF   260
260:      END ELSE                                                             PASCREF   261
```

```
261:      BEGIN (*COLLISION*) H := H+D; D := D+2;                                PASCREF   262
262:        IF H >= P THEN H := H-P;                                             PASCREF   263
263:        IF D = P THEN                                                        PASCREF   264
264:          BEGIN WRITELN; WRITELN(STARS,' TABLE FULL',STARS); GOTO 99         PASCREF   265
265:          END                                                                PASCREF   266
266:      END                                                                    PASCREF   267
267:    UNTIL F                                                                  PASCREF   268
268:  END (*SEARCH*) ;                                                          PASCREF   269
269:                                                                           PASCREF   270
270:  PROCEDURE SORT(MIN, MAX: INTEGER);                                        PASCREF   271
271:                                                                           PASCREF   272
272:  (* QUICKSORT WITH BOUNDED RECURSION DEPTH *)                             PASCREF   273
273:  (* REQUIRES MIN <= MAX *)                                                 PASCREF   274
274:                                                                           PASCREF   275
275:     VAR                                                                   PASCREF   276
276:           LOW,                                                            PASCREF   277
277:         HIGH: INDEX;                                                      PASCREF   278
278:        MIDKEY: ALFA;                                                      PASCREF   279
279:         TEMP: WORD;                                                       PASCREF   280
280:                                                                           PASCREF   281
281:     BEGIN                                                                 PASCREF   282
282:        REPEAT (*PICK SPLIT POINT*)                                        PASCREF   283
283:           MIDKEY := T[(MIN + MAX) DIV 2].KEY;                             PASCREF   284
284:           LOW := MIN;                                                     PASCREF   285
285:           HIGH := MAX;                                                    PASCREF   286
286:           REPEAT (*PARTITION*)                                            PASCREF   287
287:              WHILE T[LOW].KEY < MIDKEY DO                                 PASCREF   288
288:                 LOW := LOW + 1;                                           PASCREF   289
289:              WHILE T[HIGH].KEY > MIDKEY DO                                PASCREF   290
290:                 HIGH := HIGH - 1;                                         PASCREF   291
291:              IF LOW <= HIGH THEN                                          PASCREF   292
292:                 BEGIN                                                     PASCREF   293
293:                    TEMP := T[LOW];                                        PASCREF   294
294:                    T[LOW] := T[HIGH];                                     PASCREF   295
295:                    T[HIGH] := TEMP;                                       PASCREF   296
296:                    LOW := LOW + 1;                                        PASCREF   297
297:                    HIGH := HIGH - 1                                       PASCREF   298
298:                 END;                                                      PASCREF   299
299:           UNTIL LOW > HIGH;                                               PASCREF   300
300:                                                                           PASCREF   301
301:           (*RECURSIVELY SORT SHORTER SUB-SEGMENT*)                        PASCREF   302
302:           IF HIGH - MIN < MAX - LOW                                       PASCREF   303
303:           THEN                                                            PASCREF   304
304:              BEGIN                                                        PASCREF   305
305:                 IF MIN < HIGH THEN                                        PASCREF   306
306:                    SORT(MIN, HIGH);                                       PASCREF   307
307:                 MIN := LOW                                                PASCREF   308
308:              END                                                          PASCREF   309
309:           ELSE                                                            PASCREF   310
310:              BEGIN                                                        PASCREF   311
311:                 IF LOW < MAX THEN                                         PASCREF   312
312:                    SORT(LOW, MAX);                                        PASCREF   313
313:                 MAX := HIGH                                               PASCREF   314
314:              END                                                          PASCREF   315
315:        UNTIL MAX <= MIN                                                   PASCREF   316
316:     END (*SORT*);                                                         PASCREF   317
317:                                                                           PASCREF   318
318:                                                                           PASCREF   319
319:  PROCEDURE NOTEPROC;   (*NOTE INSTANCE OF PROCEDURE OR FUNCTION*)         PASCREF   320
320:    VAR P: PROCREF;                                                       PASCREF   321
321:  BEGIN PROCORFUNC := FALSE;                                              PASCREF   322
322:    NEW(P); PROCPTR^.NEXT := P;                                           PASCREF   323
323:    P^.NAME := ID.A; P^.LNO := LN; P^.NEXT := NIL;                        PASCREF   324
324:    PROCPTR := P                                                          PASCREF   325
325:  END (*NOTEPROC*) ;                                                      PASCREF   326
```

```
326:                                                                    PASCREF   327
327:  PROCEDURE PRINTWORD(W: WORD);                                     PASCREF   328
328:    VAR L: INTEGER; X,Y: REF;                                       PASCREF   329
329:        K: ALFA;                                                    PASCREF   330
330:  BEGIN K := W.KEY; L := KLN;                                       PASCREF   331
331:    WHILE K[L] = C00 DO BEGIN K[L] := ' '; L := L - 1 END;          PASCREF   332
332:    BEGINLINE; WRITE(' ', K);                                       PASCREF   333
333:    X := W.LAST^.NEXT; Y := X;                                      PASCREF   334
334:    L := 0;                                                         PASCREF   335
335:    REPEAT                                                          PASCREF   336
336:      IF L = NOPL THEN                                              PASCREF   337
337:        BEGIN L := 0; WRITELN; BEGINLINE; WRITE(' ':KLN+1)          PASCREF   338
338:        END;                                                        PASCREF   339
339:      L := L+1; WRITE(X^.LNO: DGPN); X := X^.NEXT                   PASCREF   340
340:    UNTIL X = Y;                                                    PASCREF   341
341:    WRITELN                                                         PASCREF   342
342:  END (*PRINTWORD*) ;                                               PASCREF   343
343:                                                                    PASCREF   344
344:  PROCEDURE PRINTTABLE;                                            PASCREF   345
345:    VAR I,M: INDEX;                                                 PASCREF   346
346:  BEGIN M := 0;    (*COMPRESS TABLE*)                              PASCREF   347
347:    FOR I := 0 TO P-1 DO                                            PASCREF   348
348:      IF T[I].KEY <> EMPTY THEN                                     PASCREF   349
349:        BEGIN T[M] := T[I]; M := M+1                                PASCREF   350
350:        END ;                                                       PASCREF   351
351:    IF M > 0 THEN SORT(0,M-1);                                      PASCREF   352
352:    NOPL := (LLOUT-KLN-1) DIV DGPN;                                 PASCREF   353
353:    SKIP(2);                                                        PASCREF   354
354:    BEGINLINE;                                                      PASCREF   355
355:    WRITELN(' CROSS REFERENCE OF IDENTIFIERS,',                     PASCREF   356
356:            ' LABEL DECLARATIONS AND GOTO STATEMENTS:');            PASCREF   357
357:    SKIP(1);                                                        PASCREF   358
358:    FOR I := 0 TO M-1 DO PRINTWORD(T[I])                            PASCREF   359
359:  END (*PRINTTABLE*) ;                                              PASCREF   360
360:                                                                    PASCREF   361
361:  PROCEDURE PRINTPROCS;                                            PASCREF   362
362:    VAR N: ALFA; L: INTEGER;                                        PASCREF   363
363:  BEGIN SKIP(2); BEGINLINE;                                         PASCREF   364
364:    WRITELN(' LIST OF PROCEDURES AND FUNCTIONS:');                  PASCREF   365
365:    SKIP(1);                                                        PASCREF   366
366:    PROCPTR := FIRSTPROC^.NEXT;                                     PASCREF   367
367:    WHILE PROCPTR <> NIL DO                                         PASCREF   368
368:      WITH PROCPTR^ DO                                              PASCREF   369
369:        BEGIN BEGINLINE;                                            PASCREF   370
370:          N := NAME; L := KLN;                                      PASCREF   371
371:          WHILE N[L] = C00 DO BEGIN N[L] := ' '; L := L - 1 END;    PASCREF   372
372:          WRITELN(N:24, LNO:10);                                    PASCREF   373
373:          PROCPTR := NEXT                                           PASCREF   374
374:        END                                                         PASCREF   375
375:  END (*PRINTPROCS*) ;                                             PASCREF   376
376:                                                                    PASCREF   377
377:  PROCEDURE INITIALIZE;                                            PASCREF   378
378:    TYPE SETTING = PACKED RECORD                                    PASCREF   379
379:                     CASE SWITCH: BOOLEAN OF                        PASCREF   380
380:                       TRUE: (ONOFF: CHAR);                         PASCREF   381
381:                       FALSE: (SIZE: 0..999999)                     PASCREF   382
382:                     END;                                           PASCREF   383
383:    VAR S: SETTING;                                                 PASCREF   384
384:                                                                    PASCREF   385
385:    FUNCTION OPTION(NAME: CHAR; VAR S: SETTING): BOOLEAN;           PASCREF   386
386:      EXTERN;                                                       PASCREF   387
387:                                                                    PASCREF   388
388:  BEGIN                                                            PASCREF   389
389:    IF OPTION('U',S) THEN                                           PASCREF   390
390:      IF S.SWITCH AND (S.ONOFF = '+')                               PASCREF   391
```

```
391:        THEN LLIN := LLMIN;                                                    PASCREF   392
392:    IF OPTION('W',S) THEN                                                      PASCREF   393
393:      IF S.SWITCH AND (S.ONOFF = '+')                                          PASCREF   394
394:        THEN LLOUT := LLMIN;                                                   PASCREF   395
395:    NEW(PROCPTR); FIRSTPROC := PROCPTR; PROCPTR^.NEXT := NIL;                  PASCREF   396
396:    PAGINATING := INPUT^ <> 'Q';                                               PASCREF   397
397:    IF INPUT^ = 'Q' THEN READLN;                                               PASCREF   398
398:    WRITELN('Q');                                                              PASCREF   399
399:    IF INPUT^ = 'T' THEN                                                       PASCREF   400
400:      BEGIN LPPG := LPPG8; LTPG := LTPG8; READLN; WRITELN('T') END;            PASCREF   401
401:    IF INPUT^ <> '1' THEN                                                      PASCREF   402
402:      BEGIN COMPILERLISTING := FALSE;                                          PASCREF   403
403:        M := LPPG;                                                             PASCREF   404
404:        LINENUMBERS := INPUT^ IN ['0'..'9']                                    PASCREF   405
405:      END                                                                      PASCREF   406
406:  END (*INITIALIZE*) ;                                                        PASCREF   407
407:                                                                             PASCREF   408
408:  PROCEDURE SCANANDLISTINPUT;                                                PASCREF   409
409:    VAR I: 1..LLINMAX;  NC, NCLAST: NESTKIND;                                PASCREF1   43
410:  BEGIN                                                                       PASCREF   411
411:    WHILE NOT EOF(INPUT) DO                                                  PASCREF   412
412:    BEGIN NEWLINE;                                                           PASCREF   413
413:      WHILE NOT EOLN(INPUT) DO                                               PASCREF   414
414:      CASE INPUT^ OF                                                         PASCREF   415
415:       'A','B','C','D','E','F','G','H','I','J','K','L','M',                 PASCREF   416
416:       'N','O','P','Q','R','S','T','U','V','W','X','Y','Z':                 PASCREF   417
417:        BEGIN K := 0; ID.A := EMPTY;                                         PASCREF   418
418:          REPEAT                                                             PASCREF   419
419:            IF K < KLN THEN                                                  PASCREF   420
420:              BEGIN K := K+1; ID.A[K] := INPUT^                              PASCREF   421
421:              END;                                                           PASCREF   422
422:            ADVANCE                                                          PASCREF   423
423:          UNTIL NOT(INPUT^ IN ['A'..'Z', '0'..'9']);                         PASCREF   424
424:          CLASSIFY;                                                          PASCREF   425
425:          IF KEYINDEX = 0 THEN                                               PASCREF   426
426:            BEGIN SEARCH;                                                    PASCREF   427
427:              IF PROCORFUNC THEN NOTEPROC                                    PASCREF   428
428:            END                                                              PASCREF   429
429:          ELSE                                                               PASCREF   430
430:            BEGIN  NC := KEY[KEYINDEX].NESTCHANGE;                           PASCREF1   44
431:              CASE NC OF                                                     PASCREF1   45
432:                NESTBEGIN:                                                   PASCREF1   46
433:                  BEGIN NESTING := NESTING + 1; INBODY := TRUE END;          PASCREF1   47
434:                NESTOPEN:                                                    PASCREF1   48
435:                  IF INBODY THEN NESTING := NESTING + 1                      PASCREF1   49
436:                  ELSE NC := NESTNULL;                                       PASCREF1   50
437:                NESTCLOSE:                                                   PASCREF1   51
438:                  IF INBODY THEN NESTING := NESTING - 1                      PASCREF1   52
439:                  ELSE NC := NESTNULL;                                       PASCREF1   53
440:                NESTCLEAR:                                                   PASCREF1   54
441:                  BEGIN  PROCORFUNC := TRUE;  INBODY := FALSE;               PASCREF1   55
442:                     IF NESTING = 0 THEN NC := NESTNULL ELSE NESTING := 0    PASCREF1   56
443:                  END;                                                       PASCREF1   57
444:                NESTNULL: ;                                                  PASCREF1   58
445:              END;                                                           PASCREF1   59
446:              IF NC <> NESTNULL THEN                                         PASCREF1   60
447:                BEGIN  NESTCOUNT := NESTCOUNT + 1;                           PASCREF1   61
448:                  NEST[NESTCOUNT] := NC                                      PASCREF1   62
449:                END                                                          PASCREF1   63
450:            END                                                              PASCREF1   64
451:        END;                                                                 PASCREF   451
452:       '0','1','2','3','4','5','6','7','8','9':                             PASCREF   452
453:          REPEAT ADVANCE;                                                    PASCREF   453
454:          UNTIL NOT (INPUT^ IN ['B','E','0'..'9']);                          PASCREF   454
455:       '''':                                                                 PASCREF   455
```

```
456:        BEGIN (*STRING*)                                                    PASCREF   456
457:          REPEAT ADVANCE;                                                   PASCREF   457
458:          UNTIL (INPUT^ = '''') OR EOLN(INPUT);                             PASCREF   458
459:          IF NOT EOLN(INPUT) THEN                                           PASCREF   459
460:            ADVANCE                                                         PASCREF   460
461:        END;                                                                PASCREF   461
462:       '_':                                                                 PASCREF   462
463:        BEGIN (*COMMENT*)                                                   PASCREF   463
464:          REPEAT ADVANCE;                                                   PASCREF   464
465:            WHILE EOLN(INPUT) DO                                            PASCREF   465
466:              BEGIN WRITELN; GET(INPUT); NEWLINE                            PASCREF   466
467:              END                                                           PASCREF   467
468:          UNTIL INPUT^ = '?';                                               PASCREF   468
469:          ADVANCE                                                           PASCREF   469
470:        END;                                                                PASCREF   470
471:       '(':                                                                 PASCREF   471
472:        BEGIN ADVANCE;                                                      PASCREF   472
473:          IF INPUT^ = '*' THEN                                              PASCREF   473
474:          BEGIN (*COMMENT*) ADVANCE;                                        PASCREF   474
475:            REPEAT                                                          PASCREF   475
476:              WHILE INPUT^ <> '*' DO                                        PASCREF   476
477:              BEGIN                                                         PASCREF   477
478:                IF EOLN(INPUT) THEN                                         PASCREF   478
479:                  BEGIN GET(INPUT); WRITELN; NEWLINE                        PASCREF   479
480:                  END ELSE                                                  PASCREF   480
481:                  ADVANCE                                                   PASCREF   481
482:              END ;                                                         PASCREF   482
483:              ADVANCE                                                       PASCREF   483
484:            UNTIL INPUT^ = ')';                                             PASCREF   484
485:            ADVANCE                                                         PASCREF   485
486:          END                                                               PASCREF   486
487:        END;                                                                PASCREF   487
488:       '+','-','*','/',')','$','=',' ',',','.','[',']',                    PASCREF   488
489:       '"','!','&','#','?','<','>','@','\','^',';',COL,PER:                PASCREF1   65
490:        ADVANCE                                                           PASCREF   490
491:      END (*CASE*) ;                                                      PASCREF   491
492:      IF (LLOUT = LLOUTMAX) AND (NESTCOUNT > 0) THEN                      PASCREF1   66
493:        BEGIN                                                             PASCREF   493
494:          IF CCOUNT >= 100 THEN WRITE('  ')                               PASCREF   494
495:          ELSE WRITE(' ':100-CCOUNT);                                     PASCREF   495
496:          NESTING := NESTINGATBOL;                                        PASCREF1   67
497:          NCLAST := NESTNULL;                                             PASCREF1   68
498:          IF NEST[1] = NESTCLOSE THEN WRITE(' ');                         PASCREF1   69
499:          FOR I := 1 TO NESTCOUNT DO                                      PASCREF1   70
500:            BEGIN NC := NEST[I];                                          PASCREF1   71
501:              CASE NC OF                                                  PASCREF1   72
502:                NESTBEGIN, NESTOPEN:                                      PASCREF1   73
503:                  BEGIN  NESTING := NESTING + 1;                          PASCREF1   74
504:                    WRITE('[', NESTING:1)                                 PASCREF1   75
505:                  END;                                                    PASCREF1   76
506:                NESTCLOSE:                                                PASCREF1   77
507:                  IF NESTING = 0 THEN WRITE('*]')                         PASCREF1   78
508:                  ELSE                                                    PASCREF1   79
509:                    BEGIN                                                 PASCREF1   80
510:                      IF NOT (NCLAST IN [NESTBEGIN,NESTOPEN]) THEN        PASCREF1   81
511:                        WRITE(NESTING:1);                                 PASCREF1   82
512:                      WRITE(']');  NESTING := NESTING - 1                 PASCREF1   83
513:                    END;                                                  PASCREF1   84
514:                NESTCLEAR:                                                PASCREF1   85
515:                  BEGIN WRITE(' * ');  NESTING := 0 END                   PASCREF1   86
516:              END;                                                        PASCREF1   87
517:              NCLAST := NC                                                PASCREF1   88
518:            END                                                           PASCREF   510
519:        END;                                                              PASCREF   511
520:      WRITELN; READLN                                                     PASCREF   512
```

```
521:    END ;                                                                    PASCREF   513
522:  END (*SCANANDLISTINPUT*) ;                                                 PASCREF   514
523:                                                                             PASCREF   515
524:  BEGIN (*CROSSREF*)                                                         PASCREF   516
525:    IF NOT EOF(INPUT) THEN                                                   PASCREF   517
526:    BEGIN LINELIMIT(OUTPUT, MAXN); INITIALIZE;                               PASCREF   518
527:      SCANANDLISTINPUT; LINELIMIT(OUTPUT, MAXN);                             PASCREF   519
528:      IF NOT PAGINATING THEN                                                 PASCREF   520
529:        BEGIN PAGINATING := TRUE; M := LPPG END;                             PASCREF   521
530:      PRINTTABLE; PRINTPROCS                                                 PASCREF   522
531:    END ELSE WRITELN(STARS,' NO PROGRAM FOUND TO CROSS REFERENCE',STARS);    PASCREF   523
532:  99:END .                                                                   PASCREF   524

THAT'S ALL FOLKS!      LINES: 532   CHARACTERS: 45752
```