```
0001:  (*$A+  USE ASCII CHARACTER SET. *)
0002:                                                                                  INT      2
0003:  (*ASSEMBLER AND INTERPRETER OF PASCAL CODE*)                                    I        1
0004:  (*K. JENSEN, N. WIRTH, CH. JACOBI, ETH MAY 76*)                                 I        2
0005:                                                                                  I        3
0006:  PROGRAM PCODE(INPUT,OUTPUT,PRD,PRR);                                            INT      4
0007:                                                                                  INT      5
0008:  (* NOTE FOR THE IMPLEMENTATION.                                                 INT      6
0009:     ==========================                                                   INT      7
0010:  THIS INTERPRETER IS WRITTEN FOR THE CASE WHERE ALL THE FUNDAMENTAL              INT      8
0011:  TYPES TAKE ONE STORAGE UNIT.                                                    INT      9
0012:  IN AN IMPLEMENTATION ALL THE HANDLING OF THE SP POINTER HAS TO TAKE             INT     10
0013:  INTO ACCOUNT THE FACT TAHT THE TYPES MAY HAVE A LENGTH DIFFERENT FROM           INT     11
0014:  ONE. SO IN PUSH AND POP OPERATIONS THE IMPLEMENTOR HAS TO INCREASE              INT     12
0015:  AND DECREASE THE SP NOT BY 1 BUT BY A NUMBER DEPENDING ON THE TYPE              INT     13
0016:  CONCERNED.                                                                      INT     14
0017:  WHERE THE NUMBER OF UNITS OF STORAGE IS USED EXPLICITELY,                       I        4
0018:  THIS VALUE MUST NOT BE CORRECTED, BECAUSE                                       I        5
0019:  THE COMPILER HAS COMPUTED IT TAKING INTO ACCOUNT THE LENGTHS OF THE             INT     17
0020:  TYPES INVOLVED.                                                                 INT     18
0021:  THE SAME HOLDS FOR THE HANDLING OF THE NP POINTER (WHICH MUST NOT BE            INT     19
0022:  CORRECTED)                                                          *)          INT     20
0023:                                                                                  INT     21
0024:  LABEL 1;                                                                        INT     22
0025:  CONST CODEMAX =8650;                                                            BIG      1
0026:        PCMAX = 17500;                                                            BIG      2
0027:         MAXSTK      = 13650; (* SIZE OF VARIABLE STORE *)                       INT     25
0028:         OVERI       = 13655; (* SIZE OF INTEGER CONSTANT TABLE = 5 *)           INT     26
0029:         OVERR       = 13660; (* SIZE OF REAL CONSTANT TABLE = 5 *)              INT     27
0030:         OVERS       = 13730; (* SIZE OF SET CONSTANT TABLE = 70 *)              INT     28
0031:        OVERB = 13820;                                                            BIG      3
0032:         OVERM = 18000;                                                           I        9
0033:         MAXSTR = 18001;                                                          I       10
0034:         LARGEINT    = 26144;                                                     I       11
0035:         BEGINCODE   = 3;                                                         INT     33
0036:         INPUTADR   = 5;                                                          N        1
0037:         OUTPUTADR  = 6;                                                          N        2
0038:         PRDADR     = 7;                                                          N        3
0039:         PRRADR     = 8;                                                          N        4
0040:        DUMINST     = 62;                                                         N        5
0041:                                                                                  INT     38
0042:  TYPE  BIT4         = 0..15;                                                     INT     39
0043:        BIT6        = 0..127;                                                     I       13
0044:        BIT20       = -26143..26143;                                              I       14
0045:        DATATYPE    = (UNDEF,INT,REEL,BOOL,SETT,ADR,MARK,CAR);                    I       15
0046:        ADDRESS      = -1..MAXSTR;                                                INT     43
0047:        BETA         = PACKED ARRAY[1..25] OF CHAR; (*ERROR MESSAGE*)            INT     44
0048:                                                                                  INT     45
0049:  VAR  CODE          : ARRAY[0..CODEMAX] OF   (* THE PROGRAM *)                   INT     46
0050:                        PACKED RECORD  OP1    :BIT6;                             INT     47
0051:                                       P1     :BIT4;                             INT     48
0052:                                       Q1     :BIT20;                            INT     49
0053:                                       OP2    :BIT6;                             INT     50
0054:                                       P2     :BIT4;                             INT     51
0055:                                       Q2     :BIT20                             INT     52
0056:                               END;                                               INT     53
0057:       PC            : 0..PCMAX;  (*PROGRAM ADDRESS REGISTER*)                    INT     54
0058:       OP : BIT6;  P : BIT4;  Q : BIT20;  (*INSTRUCTION REGISTER*)                INT     55
0059:                                                                                  INT     56
0060:       STORE         : ARRAY [0..OVERM] OF                                        INT     57
0061:                          RECORD CASE DATATYPE OF                                 I       16
0062:                                   INT        :(VI :INTEGER);                    INT     59
0063:                                   REEL       :(VR :REAL);                       INT     60
0064:                                   BOOL       :(VB :BOOLEAN);                    INT     61
0065:                                   SETT       :(VS :SET OF 0..58);               INT     62
```

```
0066:                                CAR        :(VC: CHAR);                          I      17
0067:                                ADR        :(VA : ADDRESS);                      I      18
0068:                                             (*ADDRESS IN STORE*)                 I      19
0069:                                MARK       :(VM :INTEGER);                       INT    64
0070:                           END;                                                   INT    65
0071:       MP,SP,NP,EP  : ADDRESS;  (* ADDRESS REGISTERS *)                          I      20
0072:      (*MP  POINTS TO BEGINNING OF A DATA SEGMENT                                 INT    67
0073:       SP  POINTS TO TOP OF THE STACK                                             INT    68
0074:       EP  POINTS TO THE STACK WENN IT IS GROWTH THE MAXIMUM                      I      21
0075:       NP  POINTS TO TOP OF DYNAMICLY ALLOCATED AREA*)                            INT    69
0076:                                                                                  INT    70
0077:      INTERPRETING  : BOOLEAN;                                                    INT    71
0078:      PRD,PRR: TEXT;(*PRD FOR READ ONLY, PRR FOR WRITE ONLY *)                    INT    72
0079:                                                                                  INT    73
0080:      INSTR         : ARRAY[BIT6] OF ALFA; (* MNEMONIC INSTRUCTION CODES *)      INT    74
0081:      COP           : ARRAY[BIT6] OF INTEGER;                                     I      22
0082:      SPTABLE       : ARRAY[0..20] OF ALFA;                                       I      23
0083:        (*STANDARD FUNCTIONS AND PROCEDURES*)                                     I      24
0084:                                                                                  INT    76
0085:      (*LOCALY USED FOR INTERPRETING ONE INSTRUCTION*)                            I      25
0086:      AD,AD1: ADDRESS;                                                            I      26
0087:      B: BOOLEAN;                                                                 I      27
0088:      I,J,I1,I2: INTEGER;                                                         I      28
0089:       C: CHAR;                                                                   I      29
0090:                                                                                  INT    77
0091:  (*-----------------------------------------------------------------*)           I      30
0092:                                                                                  INT    79
0093:  PROCEDURE LOAD;                                                                 INT    80
0094:     CONST MAXLABEL = 1850;                                                       I      31
0095:     TYPE  LABELST  = (ENTERED,DEFINED); (*LABEL SITUATION*)                      INT    82
0096:           LABELRG  = 0..MAXLABEL;   (*LABEL RANGE*)                              INT    83
0097:           LABELREC = RECORD                                                      INT    84
0098:                            VAL: ADDRESS;                                         INT    85
0099:                             ST: LABELST                                          INT    86
0100:                      END;                                                        INT    87
0101:     VAR  ICP,RCP,SCP,BCP,MCP  : ADDRESS;  (*POINTERS TO NEXT FREE POSITION*)    INT    88
0102:          WORD : ARRAY[1..10] OF CHAR; I  : INTEGER;  CH  : CHAR;               INT    89
0103:          LABELTAB: ARRAY[LABELRG] OF LABELREC;                                 INT    90
0104:          LABELVALUE: ADDRESS;                                                  INT    91
0105:                                                                                INT    92
0106:     PROCEDURE INIT;                                                            INT    93
0107:        VAR I: INTEGER;                                                         INT    94
0108:     BEGIN INSTR[ 0]:='LOD       '; INSTR[ 1]:='LDO       ';                   INT    95
0109:           INSTR[ 2]:='STR       '; INSTR[ 3]:='SRO       ';                   INT    96
0110:           INSTR[ 4]:='LDA       '; INSTR[ 5]:='LAO       ';                   INT    97
0111:           INSTR[ 6]:='STO       '; INSTR[ 7]:='LDC       ';                   INT    98
0112:           INSTR[ 8]:='...       '; INSTR[ 9]:='IND       ';                   INT    99
0113:           INSTR[10]:='INC       '; INSTR[11]:='MST       ';                   INT   100
0114:           INSTR[12]:='CUP       '; INSTR[13]:='ENT       ';                   INT   101
0115:           INSTR[14]:='RET       '; INSTR[15]:='CSP       ';                   INT   102
0116:           INSTR[16]:='IXA       '; INSTR[17]:='EQU       ';                   INT   103
0117:           INSTR[18]:='NEQ       '; INSTR[19]:='GEQ       ';                   INT   104
0118:           INSTR[20]:='GRT       '; INSTR[21]:='LEQ       ';                   INT   105
0119:           INSTR[22]:='LES       '; INSTR[23]:='UJP       ';                   INT   106
0120:           INSTR[24]:='FJP       '; INSTR[25]:='XJP       ';                   INT   107
0121:           INSTR[26]:='CHK       '; INSTR[27]:='EOF       ';                   INT   108
0122:           INSTR[28]:='ADI       '; INSTR[29]:='ADR       ';                   INT   109
0123:           INSTR[30]:='SBI       '; INSTR[31]:='SBR       ';                   INT   110
0124:           INSTR[32]:='SGS       '; INSTR[33]:='FLT       ';                   INT   111
0125:           INSTR[34]:='FLO       '; INSTR[35]:='TRC       ';                   INT   112
0126:           INSTR[36]:='NGI       '; INSTR[37]:='NGR       ';                   INT   113
0127:           INSTR[38]:='SQI       '; INSTR[39]:='SQR       ';                   INT   114
0128:           INSTR[40]:='ABI       '; INSTR[41]:='ABR       ';                   INT   115
0129:           INSTR[42]:='NOT       '; INSTR[43]:='AND       ';                   INT   116
0130:           INSTR[44]:='IOR       '; INSTR[45]:='DIF       ';                   INT   117
```

```
0131:          INSTR[46]:='INT       '; INSTR[47]:='UNI       ';                    INT    118
0132:          INSTR[48]:='INN       '; INSTR[49]:='MOD       ';                    INT    119
0133:          INSTR[50]:='ODD       '; INSTR[51]:='MPI       ';                    INT    120
0134:          INSTR[52]:='MPR       '; INSTR[53]:='DVI       ';                    INT    121
0135:          INSTR[54]:='DVR       '; INSTR[55]:='MOV       ';                    INT    122
0136:          INSTR[56]:='LCA       '; INSTR[57]:='DEC       ';                    INT    123
0137:          INSTR[58] := 'STP       '; INSTR[59] := 'ORD       ';               I      32
0138:          INSTR[60] := 'CHR       '; INSTR[61] := 'UJC       ';               I      33
0139:                                                                               INT    125
0140:          SPTABLE[ 0]:='GET       '; SPTABLE[ 1]:='PUT       ';               INT    126
0141:          SPTABLE[ 2]:='RST       '; SPTABLE[ 3]:='RLN       ';               INT    127
0142:          SPTABLE[ 4]:='NEW       '; SPTABLE[ 5]:='WLN       ';               INT    128
0143:          SPTABLE[ 6]:='WRS       '; SPTABLE[ 7]:='ELN       ';               INT    129
0144:          SPTABLE[ 8]:='WRI       '; SPTABLE[ 9]:='WRR       ';               INT    130
0145:          SPTABLE[10]:='WRC       '; SPTABLE[11]:='RDI       ';               INT    131
0146:          SPTABLE[12]:='RDR       '; SPTABLE[13]:='RDC       ';               INT    132
0147:          SPTABLE[14]:='SIN       '; SPTABLE[15]:='COS       ';               INT    133
0148:          SPTABLE[16]:='EXP       '; SPTABLE[17]:='LOG       ';               INT    134
0149:          SPTABLE[18]:='SQT       '; SPTABLE[19]:='ATN       ';               INT    135
0150:          SPTABLE[20]:='SAV       ';                                           INT    136
0151:           COP[ 0] := 105;  COP[ 1] :=  65;                                    I      34
0152:           COP[ 2] :=  70;  COP[ 3] :=  75;                                    I      35
0153:           COP[ 6] :=  80;  COP[ 9] :=  85;                                    I      36
0154:           COP[10] :=  90;  COP[26] :=  95;                                    I      37
0155:           COP[57] := 100;                                                     I      38
0156:          PC:= BEGINCODE;                                                      INT    137
0157:          ICP := MAXSTK + 1;                                                   I      39
0158:          RCP := OVERI + 1;                                                    I      40
0159:          SCP := OVERR + 1;                                                    I      41
0160:          BCP := OVERS + 2;                                                    I      42
0161:          MCP := OVERB + 1;                                                    I      43
0162:          FOR I:= 1 TO 10 DO WORD[I]:= ' ';                                    INT    143
0163:          FOR I:= 0 TO MAXLABEL DO                                             INT    144
0164:              WITH LABELTAB[I] DO BEGIN VAL:=-1; ST:= ENTERED END;             INT    145
0165:          RESET(PRD);                                                          INT    146
0166:     END;(*INIT*)                                                              INT    147
0167:                                                                               INT    148
0168:     PROCEDURE ERRORL(STRING: BETA); (*ERROR IN LOADING*)                      INT    149
0169:     BEGIN WRITELN;                                                            INT    150
0170:        WRITE(STRING);                                                         I      44
0171:        HALT;                                                                  I      45
0172:     END; (*ERRORL*)                                                           INT    152
0173:                                                                               INT    153
0174:     PROCEDURE UPDATE(X: LABELRG); (*WHEN A LABEL DEFINITION LX IS FOUND*)     INT    154
0175:        VAR CURR,SUCC: -1..PCMAX; (*RESP. CURRENT ELEMENT AND SUCCESSOR ELEMENT INT    155
0176:                                     OF A LIST OF FUTURE REFERENCE*)           INT    156
0177:            ENDLIST: BOOLEAN;                                                  INT    157
0178:     BEGIN                                                                     INT    158
0179:        IF LABELTAB[X].ST=DEFINED THEN ERRORL(' DUPLICATED LABEL       ')      INT    159
0180:        ELSE BEGIN                                                             INT    160
0181:               IF LABELTAB[X].VAL<>-1 THEN (*FORWARD REFERENCE(S)*)            INT    161
0182:               BEGIN CURR:= LABELTAB[X].VAL; ENDLIST:= FALSE;                  INT    162
0183:                  WHILE NOT ENDLIST DO                                         INT    163
0184:                        WITH CODE[CURR DIV 2] DO                               INT    164
0185:                        BEGIN                                                  INT    165
0186:                           IF ODD(CURR) THEN BEGIN SUCC:= Q2;                  INT    166
0187:                                             Q2:= LABELVALUE                   INT    167
0188:                                       END                                     INT    168
0189:                                     ELSE BEGIN SUCC:= Q1;                     INT    169
0190:                                                Q1:= LABELVALUE                INT    170
0191:                                           END;                                INT    171
0192:                           IF SUCC=-1 THEN ENDLIST:= TRUE                      INT    172
0193:                                      ELSE CURR:= SUCC                         INT    173
0194:                        END;                                                   INT    174
0195:                END;                                                          INT    175
```

```
0196:               LABELTAB[X].ST:= DEFINED;                                          INT   176
0197:               LABELTAB[X].VAL:= LABELVALUE;                                      INT   177
0198:             END                                                                  INT   178
0199:     END;(*UPDATE*)                                                             INT   179
0200:                                                                                INT   180
0201:     PROCEDURE ASSEMBLE; FORWARD;                                               INT   181
0202:                                                                                INT   182
0203:     PROCEDURE GENERATE;(*GENERATE SEGMENT OF CODE*)                             INT   183
0204:        VAR X: INTEGER; (* LABEL NUMMER *)                                      INT   184
0205:            AGAIN: BOOLEAN;                                                     I      46
0206:     BEGIN                                                                      INT   185
0207:        AGAIN := TRUE;                                                          I      47
0208:        WHILE AGAIN DO                                                          I      48
0209:             BEGIN READ(PRD,CH);(* FIRST LINE OF CHARACTER*)                    INT   187
0210:                   CASE CH OF                                                   INT   188
0211:                        'I': READLN(PRD);                                       INT   189
0212:                        'L': BEGIN READ(PRD,X);                                 INT   190
0213:                                   IF NOT EOLN(PRD) THEN READ(PRD,CH);          INT   191
0214:                                   IF CH='=' THEN READ(PRD,LABELVALUE)          INT   192
0215:                                             ELSE LABELVALUE:= PC;              INT   193
0216:                                   UPDATE(X); READLN(PRD);                      INT   194
0217:                              END;                                              INT   195
0218:                        'Q': BEGIN AGAIN := FALSE; READLN(PRD) END;             I      49
0219:                        ' ': BEGIN READ(PRD,CH); ASSEMBLE END                   INT   196
0220:                   END;                                                         INT   197
0221:             END                                                                INT   198
0222:     END; (*GENERATE*)                                                          INT   199
0223:                                                                                INT   200
0224:     PROCEDURE ASSEMBLE; (*TRANSLATE SYMBOLIC CODE INTO MACHINE CODE AND STORE*) INT   201
0225:           LABEL 1; (*GOTO 1 FOR INSTRUCTIONS WITHOUT CODE GENERATION*)         I      50
0226:        VAR NAME :ALFA;  B :BOOLEAN;  R :REAL;  S :SET OF 0..58;                INT   202
0227:            C1 :CHAR;  I,S1,LB,UB :INTEGER;                                     INT   203
0228:                                                                                INT   204
0229:        PROCEDURE LOOKUP(X: LABELRG); (* SEARCH IN LABEL TABLE*)                INT   205
0230:        BEGIN CASE LABELTAB[X].ST OF                                            INT   206
0231:                  ENTERED: BEGIN Q := LABELTAB[X].VAL;                          I      51
0232:                             LABELTAB[X].VAL := PC                              I      52
0233:                           END;                                                 I      53
0234:             DEFINED: Q:= LABELTAB[X].VAL                                       INT   213
0235:             END(*CASE LABEL..*)                                                INT   214
0236:        END;(*LOOKUP*)                                                          INT   215
0237:                                                                                INT   216
0238:        PROCEDURE LABELSEARCH;                                                  INT   217
0239:           VAR X: LABELRG;                                                      INT   218
0240:        BEGIN WHILE (CH<>'L') AND NOT EOLN(PRD) DO READ(PRD,CH);                INT   219
0241:              READ(PRD,X); LOOKUP(X)                                            INT   220
0242:        END;(*LABELSEARCH*)                                                     INT   221
0243:                                                                                INT   222
0244:        PROCEDURE GETNAME;                                                      INT   223
0245:        BEGIN  WORD[1] := CH;                                                   INT   224
0246:           READ(PRD,WORD[2],WORD[3]);                                           INT   225
0247:           IF NOT EOLN(PRD) THEN READ(PRD,CH) (*NEXT CHARACTER*);               INT   226
0248:           PACK(WORD,1,NAME)                                                    INT   227
0249:        END; (*GETNAME*)                                                        INT   228
0250:                                                                                I      54
0251:             PROCEDURE TYPESYMBOL;                                              I      55
0252:               VAR I: INTEGER;                                                  I      56
0253:             BEGIN                                                              I      57
0254:               IF CH <> 'I' THEN                                                I      58
0255:                 BEGIN                                                          I      59
0256:                   CASE CH OF                                                   I      60
0257:                     'A' : I := 0;                                              I      61
0258:                     'R' : I := 1;                                              I      62
0259:                     'S' : I := 2;                                              I      63
0260:                     'B' : I := 3;                                              I      64
```

```
0261:                    'C' : I := 4;                                                  I      65
0262:                  END;                                                             I      66
0263:                  OP := COP[OP]+I;                                                 I      67
0264:               END;                                                                I      68
0265:            END (*TYPESYMBOL*) ;                                                   I      69
0266:                                                                                   INT   229
0267:     BEGIN  P := 0;  Q := 0;  OP := 0;                                             INT   230
0268:        GETNAME;                                                                   INT   231
0269:        INSTR[DUMINST] := NAME;                                                    I      70
0270:        WHILE INSTR[OP]<>NAME DO OP := OP+1;                                       INT   232
0271:        IF OP = DUMINST THEN ERRORL(' ILLEGEAL INSTRUCTION    ');                  I      71
0272:                                                                                   INT   233
0273:        CASE OP OF  (* GET PARAMETERS P,Q *)                                       INT   234
0274:                                                                                   INT   235
0275:            (*EQU,NEQ,GEQ,GRT,LEQ,LES*)                                            INT   236
0276:            17,18,19,                                                              INT   237
0277:            20,21,22 :    BEGIN CASE CH OF                                         INT   238
0278:                                'A': ; (*P = 0*)                                   INT   239
0279:                                'I': P := 1;                                       INT   240
0280:                                'R': P := 2;                                       INT   241
0281:                                'B': P := 3;                                       INT   242
0282:                                'S': P := 4;                                       INT   243
0283:                                'C' : P := 6;                                      I      72
0284:                                'M' :BEGIN P := 5;                                 INT   244
0285:                                       READ(PRD,Q)                                 INT   245
0286:                                    END                                            INT   246
0287:                               END                                                 INT   247
0288:                          END;                                                     INT   248
0289:                                                                                   INT   249
0290:             (*LOD,STR*)                                                           I      73
0291:             0,2: BEGIN TYPESYMBOL; READ(PRD,P,Q)                                  I      74
0292:                  END;                                                             I      75
0293:             4 (*LDA*) : READ(PRD,P,Q);                                            I      76
0294:            12 (*CUP*): BEGIN READ(PRD,P); LABELSEARCH END;                        INT   252
0295:                                                                                   INT   253
0296:            11 (*MST*) :    READ(PRD,P);                                          INT   254
0297:                                                                                   INT   255
0298:            14 (*RET*) : CASE CH OF                                                INT   256
0299:                               'P': P:=0;                                          INT   257
0300:                               'I': P:=1;                                          INT   258
0301:                               'R': P:=2;                                          INT   259
0302:                               'C': P:=3;                                          INT   260
0303:                               'B': P:=4;                                          INT   261
0304:                               'A': P:= 5                                          INT   262
0305:                         END;                                                      INT   263
0306:                                                                                   INT   264
0307:             (*LAO,IXA,MOV*)                                                       I      77
0308:             5,16,55: READ(PRD,Q);                                                 I      78
0309:            (*LDO,SRO,IND,INC,DEC*)                                                I      79
0310:            1,3,9,10,57: BEGIN TYPESYMBOL; READ(PRD,Q)                             I      80
0311:                         END;                                                      I      81
0312:                                                                                   I      82
0313:            (*ENT,UJP,FJP,XJP*)                                                    INT   268
0314:            23,24,25: LABELSEARCH;                                                 I      83
0315:                                                                                   I      84
0316:            13 (*ENT*) : BEGIN READ(PRD,P); LABELSEARCH END;                       I      85
0317:                                                                                   INT   271
0318:            15 (*CSP*) :    BEGIN FOR I:=1 TO 9 DO READ(PRD,CH); GETNAME;          INT   272
0319:                             WHILE NAME<>SPTABLE[Q] DO  Q := Q+1                   INT   273
0320:                            END;                                                   INT   274
0321:                                                                                   INT   275
0322:            7 (*LDC*) :     BEGIN CASE CH OF  (*GET Q*)                            INT   276
0323:                             'I' :BEGIN  P := 1;  READ(PRD,I);                     INT   277
0324:                                    IF ABS(I)>=LARGEINT THEN                       INT   278
0325:                                    BEGIN  OP := 8;                                INT   279
```

```
0326:                                   STORE[ICP].VI := I;  Q := MAXSTK;          INT    280
0327:                                   REPEAT  Q := Q+1  UNTIL STORE[Q].VI=I;     INT    281
0328:                                   IF Q=ICP THEN                              INT    282
0329:                                   BEGIN  ICP := ICP+1;                       INT    283
0330:                                     IF ICP=OVERI THEN                        I       86
0331:                                       ERRORL(' INTEGER TABLE OVERFLOW  ');   I       87
0332:                                   END                                        INT    285
0333:                                 END  ELSE Q := I                             INT    286
0334:                               END;                                           INT    287
0335:                                                                              INT    288
0336:                             'R' :BEGIN  OP := 8; P := 2;                     INT    289
0337:                                    READ(PRD,R);                              INT    290
0338:                                    STORE[RCP].VR := R;  Q := OVERI;          INT    291
0339:                                    REPEAT  Q := Q+1  UNTIL STORE[Q].VR=R;    INT    292
0340:                                    IF Q=RCP THEN                             INT    293
0341:                                       BEGIN  RCP := RCP+1;                   INT    294
0342:                                       IF RCP = OVERR THEN                    I       88
0343:                                         ERRORL(' REAL TABLE OVERFLOW     '); I       89
0344:                                    END                                       INT    296
0345:                               END;                                           INT    297
0346:                                                                              INT    298
0347:                             'N' :; (*P,Q = 0*)                               INT    299
0348:                                                                              INT    300
0349:                             'B' :BEGIN  P := 3;  READ(PRD,Q)  END;           INT    301
0350:                                                                              INT    302
0351:                             'C': BEGIN P := 6;                               I       90
0352:                                    REPEAT READ(PRD,CH); UNTIL CH <> ' ';     I       91
0353:                                    IF CH <> '''' THEN                        I       92
0354:                                      ERRORL(' ILLEGAL CHARACTER       ');   I       93
0355:                                    READ(PRD,CH);  Q := ORD(CH);              I       94
0356:                                    READ(PRD,CH);                             I       95
0357:                                    IF CH <> '''' THEN                        I       96
0358:                                      ERRORL(' ILLEGAL CHARACTER       ');   I       97
0359:                               END;                                           I       98
0360:                             '(' :BEGIN  OP := 8;  P := 4;                    INT    303
0361:                                    S := [ ];  READ(PRD,CH);                  INT    304
0362:                                    WHILE CH<>')' DO                          INT    305
0363:                                    BEGIN READ(PRD,S1,CH); S := S + [S1]      INT    306
0364:                                    END;                                      INT    307
0365:                                    STORE[SCP].VS := S;  Q := OVERR;          INT    308
0366:                                    REPEAT  Q := Q+1  UNTIL STORE[Q].VS=S;    INT    309
0367:                                    IF Q=SCP THEN                             INT    310
0368:                                    BEGIN  SCP := SCP+1;                      INT    311
0369:                                       IF SCP=OVERS THEN                      I       99
0370:                                         ERRORL(' SET TABLE OVERFLOW      '); I      100
0371:                                    END                                       INT    313
0372:                               END                                            INT    314
0373:                           END (*CASE*)                                       INT    315
0374:                       END;                                                   INT    316
0375:                                                                              INT    317
0376:            26 (*CHK*):  BEGIN TYPESYMBOL;                                    I      101
0377:                          READ(PRD,LB,UB);                                    I      102
0378:                          IF OP = 95 THEN Q := LB                             I      103
0379:                          ELSE                                                I      104
0380:                          BEGIN                                               I      105
0381:                           STORE[BCP-1].VI := LB; STORE[BCP].VI := UB;        INT    319
0382:                           Q := OVERS;                                        INT    320
0383:                           REPEAT  Q := Q+2                                   INT    321
0384:                           UNTIL (STORE[Q-1].VI=LB)AND (STORE[Q].VI=UB);      INT    322
0385:                           IF Q=BCP THEN                                      INT    323
0386:                           BEGIN  BCP := BCP+2;                               INT    324
0387:                              IF BCP=OVERB THEN                               I      106
0388:                                ERRORL(' BOUNDARY TABLE OVERFLOW ');          I      107
0389:                           END                                                INT    326
0390:                          END                                                 I      108
```

```
0391:                          END;                                                    INT    327
0392:                                                                                  INT    328
0393:             56 (*LCA*): BEGIN                                                    I      109
0394:                           IF MCP + 16 >= OVERM THEN                            I      110
0395:                             ERRORL(' MULTIPLE TABLE OVERFLOW ');               I      111
0396:                           MCP := MCP+16;                                       I      112
0397:                           Q := MCP;                                            I      113
0398:                           FOR I := 0 TO 15 (*STRINGLGTH*) DO                   I      114
0399:                             BEGIN READ(PRD,CH);                                I      115
0400:                               STORE[Q+I].VC := CH                              I      116
0401:                             END;                                               I      117
0402:                          END;                                                  I      118
0403:                                                                                INT    336
0404:             6 (*STO*) : TYPESYMBOL;                                            I      119
0405:              27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,  I      120
0406:             48,49,50,51,52,53,54,58  :  ;                                      INT    338
0407:                                                                                INT    339
0408:             (*ORD,CHR*)                                                        I      121
0409:             59,60: GOTO 1;                                                     I      122
0410:                                                                                I      123
0411:             61 (*UJC*): ; (*MUST HAVE SAME LENGTH AS UJP*)                     I      124
0412:                                                                                I      125
0413:          END; (*CASE*)                                                         INT    340
0414:                                                                                INT    341
0415:          (* STORE INSTRUCTION *)                                              INT    343
0416:          WITH CODE[PC DIV 2] DO                                                INT    344
0417:             IF ODD(PC) THEN                                                    INT    345
0418:             BEGIN  OP2 := OP; P2 := P; Q2 := Q                                 INT    346
0419:             END  ELSE                                                          INT    347
0420:             BEGIN  OP1 := OP; P1 := P; Q1 := Q                                 INT    348
0421:             END;                                                               INT    349
0422:          PC := PC+1;                                                           INT    350
0423:          1: READLN(PRD);                                                       I      126
0424:       END; (*ASSEMBLE*)                                                        INT    351
0425:                                                                                INT    352
0426:    BEGIN (*LOAD*)                                                              INT    353
0427:       INIT;                                                                    INT    354
0428:       GENERATE;                                                                INT    355
0429:       PC := 0;                                                                 I      127
0430:       GENERATE;                                                                INT    357
0431:                                                                                INT    358
0432:    END; (*LOAD*)                                                               INT    359
0433:                                                                                INT    360
0434:    (*--------------------------------------------------------------------*)    INT    361
0435:                                                                                INT    362
0436:                                                                                INT    363
0437:    PROCEDURE PMD;                                                              INT    364
0438:       VAR S :INTEGER; I: INTEGER;                                              INT    365
0439:                                                                                INT    366
0440:       PROCEDURE PT;                                                            INT    367
0441:       BEGIN  WRITE(S:6);                                                       INT    368
0442:          IF ABS(STORE[S].VI) < MAXINT THEN WRITE(STORE[S].VI)                  I      128
0443:          ELSE WRITE('TOO BIG ');                                               I      129
0444:          S := S - 1;                                                           INT    381
0445:          I := I + 1;                                                           INT    382
0446:          IF I = 4 THEN                                                         INT    383
0447:             BEGIN WRITELN(OUTPUT); I := 0 END;                                 INT    384
0448:       END; (*PT*)                                                              INT    385
0449:                                                                                INT    386
0450:    BEGIN                                                                       I      130
0451:      WRITE(' PC =',PC-1:5,' OP =',OP:3,'  SP =',SP:5,'  MP =',MP:5,            I      131
0452:            '  NP =',NP:5);                                                    I      132
0453:       WRITELN; WRITELN('-----------------------------------');                 INT    388
0454:                                                                                INT    389
0455:       S := SP; I := 0;                                                         INT    390
```

```
0456:     WHILE S>=0 DO PT;                                                    INT    391
0457:     S := MAXSTK;                                                         INT    392
0458:     WHILE S>=NP DO PT;                                                   INT    393
0459:  END; (*PMD*)                                                            INT    394
0460:                                                                          INT    395
0461:  PROCEDURE ERRORI(STRING: BETA);                                         INT    396
0462:  BEGIN WRITELN; WRITELN(STRING);                                         INT    397
0463:        PMD; GOTO 1                                                       INT    398
0464:  END;(*ERRORI*)                                                         INT    399
0465:                                                                          INT    400
0466:  FUNCTION BASE(LD :INTEGER):ADDRESS;                                     INT    401
0467:     VAR AD :ADDRESS;                                                     INT    402
0468:  BEGIN  AD := MP;                                                       INT    403
0469:     WHILE LD>0 DO                                                        INT    404
0470:     BEGIN  AD := STORE[AD+1].VM;  LD := LD-1                             INT    405
0471:     END;                                                                 INT    406
0472:     BASE := AD                                                           INT    407
0473:  END; (*BASE*)                                                          INT    408
0474:                                                                          INT    409
0475:  PROCEDURE COMPARE;                                                     I      133
0476:  (*COMPARING IS ONLY CORRECT IF RESULT BY COMPARING INTEGERS WILL BE*)   I      134
0477:  BEGIN                                                                  I      135
0478:    I1 := STORE[SP].VA;                                                 I      136
0479:    I2 := STORE[SP+1].VA;                                               I      137
0480:    I := 0; B := TRUE;                                                  I      138
0481:    WHILE B AND (I<>Q) DO                                               I      139
0482:      IF STORE[I1+I].VI = STORE[I2+I].VI THEN I := I+1                  I      140
0483:      ELSE B := FALSE                                                   I      141
0484:  END; (*COMPARE*)                                                      I      142
0485:                                                                          I      143
0486:     PROCEDURE CALLSP;                                                    INT    419
0487:        VAR LINE: BOOLEAN; ADPTR,ADELNT: ADDRESS;                         INT    420
0488:            I: INTEGER;                                                   INT    421
0489:                                                                          INT    422
0490:        PROCEDURE READI(VAR F:TEXT);                                      INT    423
0491:           VAR AD: ADDRESS;                                               INT    424
0492:        BEGIN AD:= STORE[SP-1].VA;                                        INT    425
0493:              READ(F,STORE[AD].VI);                                       I      144
0494:              STORE[STORE[SP].VA].VC := F^;                               I      145
0495:              SP:= SP-2                                                   INT    428
0496:        END;(*READI*)                                                     INT    429
0497:                                                                          INT    430
0498:        PROCEDURE READR(VAR F: TEXT);                                     INT    431
0499:           VAR AD: ADDRESS;                                               INT    432
0500:        BEGIN AD:= STORE[SP-1].VA;                                        INT    433
0501:              READ(F,STORE[AD].VR);                                       I      146
0502:              STORE[STORE[SP].VA].VC := F^;                               I      147
0503:              SP:= SP-2                                                   INT    436
0504:        END;(*READR*)                                                     INT    437
0505:                                                                          INT    438
0506:        PROCEDURE READC(VAR F: TEXT);                                     INT    439
0507:           VAR C: CHAR; AD: ADDRESS;                                      INT    440
0508:        BEGIN READ(F,C);                                                  INT    441
0509:              AD:= STORE[SP-1].VA;                                        INT    442
0510:              STORE[AD].VC := C;                                          I      148
0511:              STORE[STORE[SP].VA].VC := F^;                               I      149
0512:              STORE[STORE[SP].VA].VI:= ORD(F^);                           INT    444
0513:              SP:= SP-2                                                   INT    445
0514:        END;(*READC*)                                                     INT    446
0515:                                                                          INT    447
0516:        PROCEDURE WRITESTR(VAR F: TEXT);                                  INT    448
0517:           VAR I,J,K: INTEGER;                                            INT    449
0518:               AD: ADDRESS;                                               INT    450
0519:        BEGIN AD:= STORE[SP-3].VA;                                        INT    451
0520:              K := STORE[SP-2].VI; J := STORE[SP-1].VI;                   I      150
```

```
0521:            (* J AND K ARE NUMBERS OF CHARACTERS *)                                  INT    453
0522:             IF K>J THEN FOR I:=1 TO K-J DO WRITE(F,' ')                           INT    454
0523:                    ELSE J:= K;                                                    INT    455
0524:             FOR I := 0 TO J-1 DO WRITE(F,STORE[AD+I].VC);                         I      151
0525:             SP:= SP-4                                                             INT    459
0526:       END;(*WRITESTR*)                                                            INT    460
0527:                                                                                   INT    461
0528:                                                                                   INT    462
0529:       PROCEDURE GETFILE(VAR F: TEXT);                                             INT    463
0530:          VAR AD: ADDRESS;                                                         INT    464
0531:       BEGIN AD:=STORE[SP].VA;                                                     INT    465
0532:            GET(F); STORE[AD].VC := F^;                                            I      152
0533:             SP:=SP-1                                                              INT    467
0534:       END;(*GETFILE*)                                                             INT    468
0535:                                                                                   INT    469
0536:       PROCEDURE PUTFILE(VAR F: TEXT);                                             INT    470
0537:          VAR AD: ADDRESS;                                                         INT    471
0538:       BEGIN AD:= STORE[SP].VA;                                                    INT    472
0539:             F^ := STORE[AD].VC; PUT(F);                                           I      153
0540:             SP:= SP-1;                                                            INT    475
0541:       END;(*PUTFILE*)                                                             INT    476
0542:                                                                                   INT    477
0543:    BEGIN (*CALLSP*)                                                               INT    478
0544:          CASE Q OF                                                                INT    479
0545:               0 (*GET*): CASE STORE[SP].VA OF                                     INT    480
0546:                               5: GETFILE(INPUT);                                  N       6
0547:                               6: ERRORI(' GET ON OUTPUT FILE      ');              N       7
0548:                               7: GETFILE(PRD);                                    N       8
0549:                               8: ERRORI(' GET ON PRR FILE         ')              N       9
0550:                          END;                                                     N      10
0551:               1 (*PUT*): CASE STORE[SP].VA OF                                     N      11
0552:                               5: ERRORI(' PUT ON READ FILE        ');             N      12
0553:                               6: PUTFILE(OUTPUT);                                 N      13
0554:                               7: ERRORI(' PUT ON PRD FILE         ');             N      14
0555:                               8: PUTFILE(PRR)                                     N      15
0556:                          END;                                                     INT    491
0557:               2 (*RST*): BEGIN                                                    I      154
0558:                            (*FOR TESTPHASE*)                                      I      155
0559:                            NP := STORE[SP].VA; SP := SP-1                         I      156
0560:                          END;                                                     INT    493
0561:               3 (*RLN*) : BEGIN CASE STORE[SP].VA OF                              INT    494
0562:                                    5: BEGIN READLN(INPUT);                        N      16
0563:                                         STORE[INPUTADR].VC := INPUT^              N      17
0564:                                       END;                                        N      18
0565:                                    6: ERRORI(' READLN ON OUTPUT FILE   ');        N      19
0566:                                    7: BEGIN READLN(INPUT);                        N      20
0567:                                         STORE[INPUTADR].VC := INPUT^              N      21
0568:                                       END;                                        N      22
0569:                                    8: ERRORI(' READLN ON PRR FILE      ')        N      23
0570:                                 END;                                              INT    499
0571:                                 SP:= SP-1                                         INT    500
0572:                            END;                                                   INT    501
0573:                4 (*NEW*): BEGIN AD:= NP-STORE[SP].VA;                             INT    502
0574:                           (*TOP OF STACK GIVES THE LENGTH IN UNITS OF STORAGE *)INT    503
0575:                                   IF AD <= EP THEN                               I      163
0576:                                     ERRORI(' STORE OVERFLOW          ');         I      164
0577:                                 NP:= AD; AD:= STORE[SP-1].VA;                     INT    506
0578:                                 STORE[AD].VA := NP;                               I      165
0579:                                 SP:=SP-2                                          INT    508
0580:                            END;                                                   INT    509
0581:                                                                                   INT    510
0582:                5 (*WLN*) : BEGIN CASE STORE[SP].VA OF                             INT    511
0583:                                    5: ERRORI(' WRITELN ON INPUT FILE   ');        N      24
0584:                                    6: WRITELN(OUTPUT);                            N      25
0585:                                    7: ERRORI(' WRITELN ON PRD FILE     ');        N      26
```

```
0586:                                   8: WRITELN(PRR)                                  N      27
0587:                                END;                                                N      28
0588:                                SP:= SP-1                                           N      29
0589:                            END;                                                    N      30
0590:            6 (*WRS*): CASE STORE[SP].VA OF                                         N      31
0591:                           5: ERRORI(' WRITE ON INPUT FILE     ');                 N      32
0592:                           6: WRITESTR(OUTPUT);                                    N      33
0593:                           7: ERRORI(' WRITE ON PRD FILE       ');                 N      34
0594:                           8: WRITESTR(PRR)                                        N      35
0595:                       END;                                                        N      36
0596:            7 (*ELN*) : BEGIN CASE STORE[SP].VA OF                                  N      37
0597:                                   5: LINE:= EOLN(INPUT);                          N      38
0598:                                   6: ERRORI(' EOLN OUTPUT FILE        ');          N      39
0599:                                   7: LINE:=EOLN(PRD);                             N      40
0600:                                   8: ERRORI(' EOLN ON PRR FILE        ')          N      41
0601:                                END;                                               N      42
0602:                               STORE[SP].VB := LINE                                N      43
0603:                            END;                                                   N      44
0604:            8 (*WRI*) : BEGIN CASE STORE[SP].VA OF                                 N      45
0605:                                   5: ERRORI(' WRITE ON INPUT FILE     ');          N      46
0606:                                   6: WRITE(OUTPUT,                                N      47
0607:                                        STORE[SP-2].VI: STORE[SP-1].VI);           N      48
0608:                                   7: ERRORI(' WRITE ON PRD FILE       ');          N      49
0609:                                   8: WRITE(PRR,STORE[SP-2].VI:                    N      50
0610:                                            STORE[SP-1].VI)                        N      51
0611:                                 END;                                              N      52
0612:                                SP:=SP-3                                           N      53
0613:                            END;                                                   N      54
0614:            9 (*WRR*) : BEGIN CASE STORE[SP].VA OF                                 N      55
0615:                                   5: ERRORI(' WRITE ON INPUT FILE     ');          N      56
0616:                                   6: WRITE(OUTPUT,                                N      57
0617:                                        STORE[SP-2].VR: STORE[SP-1].VI);           N      58
0618:                                   7: ERRORI(' WRITE ON PRD FILE       ');          N      59
0619:                                   8: WRITE(PRR,STORE[SP-2].VR:                    N      60
0620:                                            STORE[SP-1].VI)                        N      61
0621:                                END;                                               N      62
0622:                                SP:=SP-3                                           N      63
0623:                       END;                                                        N      64
0624:            10 (*WRC*):BEGIN CASE STORE[SP].VA OF                                  N      65
0625:                                  5: ERRORI(' WRITE ON INPUT FILE     ');          N      66
0626:                                  6: WRITE(OUTPUT,STORE[SP-2].VC:                  N      67
0627:                                       STORE[SP-1].VI);                            N      68
0628:                                  7: ERRORI(' WRITE ON PRD FILE       ');          N      69
0629:                                  8: WRITE(PRR,CHR(STORE[SP-2].VI):                N      70
0630:                                       STORE[SP-1].VI);                            N      71
0631:                               END;                                                N      72
0632:                               SP:=SP-3                                            N      73
0633:                       END;                                                        N      74
0634:            11(*RDI*) : CASE STORE[SP].VA OF                                       N      75
0635:                           5: READI(INPUT);                                        N      76
0636:                           6: ERRORI(' READ ON OUTPUT FILE     ');                 N      77
0637:                           7: READI(PRD);                                          N      78
0638:                           8: ERRORI(' READ ON PRR FILE        ')                  N      79
0639:                       END;                                                        N      80
0640:            12(*RDR*) : CASE STORE[SP].VA OF                                       N      81
0641:                           5: READR(INPUT);                                        N      82
0642:                           6: ERRORI(' READ ON OUTPUT FILE     ');                 N      83
0643:                           7: READR(PRD);                                          N      84
0644:                           8: ERRORI(' READ ON PRR FILE        ')                  N      85
0645:                        END;                                                       N      86
0646:            13(*RDC*):  CASE STORE[SP].VA OF                                       N      87
0647:                           5: READC(INPUT);                                        N      88
0648:                           6: ERRORI(' READ ON OUTPUT FILE     ');                 N      89
0649:                           7: READC(PRD);                                          N      90
0650:                           8: ERRORI(' READ ON PRR FILE        ')                  N      91
```

```
0651:                            END;                                                    INT    574
0652:                14(*SIN*): STORE[SP].VR:= SIN(STORE[SP].VR);                        INT    575
0653:                15(*COS*): STORE[SP].VR:= COS(STORE[SP].VR);                        INT    576
0654:                16(*EXP*): STORE[SP].VR:= EXP(STORE[SP].VR);                        INT    577
0655:                17(*LOG*): STORE[SP].VR:= LN(STORE[SP].VR);                         INT    578
0656:                18(*SQT*): STORE[SP].VR:= SQRT(STORE[SP].VR);                       INT    579
0657:                19(*ATN*): STORE[SP].VR:= ARCTAN(STORE[SP].VR);                     INT    580
0658:                20(*SAV*): BEGIN AD:=STORE[SP].VA;                                  INT    581
0659:                              STORE[AD].VA := NP;                                   I     179
0660:                              SP:= SP-1                                           INT    583
0661:                           END;                                                   INT    584
0662:           END;(*CASE Q*)                                                         INT    585
0663:     END;(*CALLSP*)                                                               INT    586
0664:                                                                                  INT    587
0665:  BEGIN (* MAIN *)                                                                 I     180
0666:    REWRITE(PRR);                                                                 I     181
0667:    LOAD; (* ASSEMBLES AND STORES CODE *)                                         I     182
0668:    WRITELN(OUTPUT); (* FOR TESTING *)                                           I     183
0669:    PC := 0; SP := -1; MP := 0; NP := MAXSTK+1; EP := 5;                          N      92
0670:    STORE[INPUTADR].VC := INPUT^;                                                 I     185
0671:    STORE[PRDADR].VC := PRD^;                                                     I     186
0672:    INTERPRETING := TRUE;                                                         I     187
0673:    WHILE INTERPRETING DO                                                         I     188
0674:    BEGIN                                                                         I     189
0675:                                                                                  I     190
0676:      (*FETCH*)                                                                   I     191
0677:      WITH CODE[PC DIV 2] DO                                                      I     192
0678:        IF ODD(PC) THEN                                                           I     193
0679:        BEGIN OP := OP2; P := P2; Q := Q2                                         I     194
0680:        END ELSE                                                                  I     195
0681:        BEGIN OP := OP1; P := P1; Q := Q1                                         I     196
0682:        END;                                                                      I     197
0683:      PC := PC+1;                                                                 I     198
0684:                                                                                  I     199
0685:      (*EXECUTE*)                                                                 I     200
0686:      CASE OP OF                                                                  I     201
0687:                                                                                 INT    589
0688:           105,106,107,108,109,                                                   I     202
0689:           0 (*LOD*): BEGIN  AD := BASE(P) + Q;                                  INT    590
0690:                      SP := SP+1;                                                 I     203
0691:                      STORE[SP] := STORE[AD]                                     INT    593
0692:                   END;                                                          INT    594
0693:                                                                                 INT    595
0694:           65,66,67,68,69,                                                        I     204
0695:           1 (*LDO*): BEGIN                                                      INT    596
0696:                      SP := SP+1;                                                 I     205
0697:                      STORE[SP] := STORE[Q]                                      INT    599
0698:                   END;                                                          INT    600
0699:                                                                                 INT    601
0700:           70,71,72,73,74,                                                        I     206
0701:           2 (*STR*): BEGIN  STORE[BASE(P)+Q] := STORE[SP];  SP := SP-1           INT    602
0702:                   END;                                                          INT    603
0703:                                                                                 INT    604
0704:           75,76,77,78,79,                                                        I     207
0705:           3 (*SRO*): BEGIN  STORE[Q] := STORE[SP];  SP := SP-1                  INT    605
0706:                   END;                                                          INT    606
0707:                                                                                 INT    607
0708:           4 (*LDA*): BEGIN SP := SP+1;                                           I     208
0709:                      STORE[SP].VA := BASE(P) + Q                                 I     209
0710:                   END;                                                          INT    610
0711:                                                                                 INT    611
0712:           5 (*LAO*): BEGIN SP := SP+1;                                           I     210
0713:                      STORE[SP].VA := Q                                           I     211
0714:                   END;                                                          INT    614
0715:                                                                                 INT    615
```

```
0716:           80,81,82,83,84,                                                  I     212
0717:           6 (*STO*): BEGIN                                                 I     213
0718:                        STORE[STORE[SP-1].VA] := STORE[SP];                 I     214
0719:                        SP := SP-2;                                         I     215
0720:                      END;                                                  I     216
0721:                                                                            INT   618
0722:           7 (*LDC*): BEGIN SP := SP+1;                                     I     217
0723:                       IF P=1 THEN                                          INT   620
0724:                       BEGIN STORE[SP].VI := Q;                             I     218
0725:                       END ELSE                                             INT   622
0726:                           IF P = 6 THEN STORE[SP].VC := CHR(Q)             I     219
0727:                           ELSE                                             I     220
0728:                             IF P = 3 THEN STORE[SP].VB := Q = 1            I     221
0729:                             ELSE (* LOAD NIL *) STORE[SP].VA := MAXSTR     I     222
0730:                  END;                                                      INT   628
0731:                                                                            INT   629
0732:           8 (*LCI*): BEGIN SP := SP+1;                                     I     223
0733:                        STORE[SP] := STORE[Q]                               I     224
0734:                  END;                                                      INT   631
0735:                                                                            INT   632
0736:           85,86,87,88,89,                                                  I     225
0737:           9 (*IND*): BEGIN AD := STORE[SP].VA + Q;                         I     226
0738:                      (* Q IS A NUMBER OF STORAGE UNITS *)                  I     227
0739:                       STORE[SP] := STORE[AD]                               INT   635
0740:                  END;                                                      INT   636
0741:                                                                            INT   637
0742:           90,91,92,93,94,                                                  I     228
0743:           10 (*INC*): BEGIN                                                I     229
0744:                         STORE[SP].VI := STORE[SP].VI+Q;                    I     230
0745:                       END;                                                 I     231
0746:                                                                            INT   639
0747:           11 (*MST*):BEGIN (*P=LEVEL OF CALLING PROCEDURE MINUS LEVEL OF CALLED INT   640
0748:                             PROCEDURE + 1;  SET DL AND SL, INCREMENT SP*)  INT   641
0749:                      (* THEN LENTH OF THIS ELEMENT IS                     INT   643
0750:                        MAX(INTSIZE,REALSIZE,BOOLSIZE,CHARSIZE,PTRSIZE *) INT   644
0751:                      STORE[SP+2].VM := BASE(P);                           I     232
0752:                      (* THE LENGTH OF THIS ELEMENT IS PTRSIZE *)           INT   646
0753:                      STORE[SP+3].VM := MP;                                I     233
0754:                      (* IDEM *)                                           INT   648
0755:                      STORE[SP+4].VM := EP;                                I     234
0756:                      (* IDEM *)                                           INT   650
0757:                      SP := SP+5                                           I     235
0758:                  END;                                                      INT   652
0759:                                                                            INT   653
0760:           12 (*CUP*):BEGIN  (*P=NO OF LOCATIONS FOR PARAMETERS, Q=ENTRY POINT*) INT   654
0761:                      MP := SP-(P+4);                                       N      93
0762:                      STORE[MP+4].VM := PC;                                 N      94
0763:                      PC := Q                                               INT   657
0764:                  END;                                                      INT   658
0765:                                                                            INT   659
0766:           13 (*ENT*): IF P = 1 THEN                                        I     237
0767:                         BEGIN SP := MP + Q; (*Q = LENGTH OF DATASEG*)       I     238
0768:                           IF SP > NP THEN ERRORI(' STORE OVERFLOW         '); I     239
0769:                         END                                                I     240
0770:                      ELSE                                                  I     241
0771:                        BEGIN EP := SP+Q;                                   I     242
0772:                          IF EP > NP THEN ERRORI(' STORE OVERFLOW         '); I     243
0773:                        END;                                                I     244
0774:                        (*Q = MAX SPACE REQUIRED ON STACK*)                 I     245
0775:                                                                            INT   667
0776:           14 (*RET*):BEGIN  CASE P OF                                      INT   668
0777:                               0: SP:= MP-1;                                INT   669
0778:                               1,2,3,4,5: SP:= MP                           INT   670
0779:                             END;                                           INT   671
0780:                             PC := STORE[MP+4].VM;                          N      95
```

```
0781:                          EP := STORE[MP+3].VM;                          N      96
0782:                          MP:= STORE[MP+2].VM;                           INT    673
0783:                    END;                                                 INT    674
0784:                                                                         INT    675
0785:           15 (*CSP*): CALLSP;                                           INT    676
0786:                                                                         INT    677
0787:           16 (*IXA*): BEGIN                                             I      247
0788:                         I := STORE[SP].VI;                              I      248
0789:                         SP := SP-1;                                     I      249
0790:                         STORE[SP].VA := Q*I+STORE[SP].VA;               I      250
0791:                       END;                                              I      251
0792:                                                                         INT    697
0793:           17 (*EQU*):BEGIN  SP := SP-1;                                 INT    698
0794:                       CASE P OF                                         INT    699
0795:                        1: STORE[SP].VB := STORE[SP].VI =  STORE[SP+1].VI;  I      252
0796:                        0: STORE[SP].VB := STORE[SP].VA =  STORE[SP+1].VA;  I      253
0797:                        6: STORE[SP].VB := STORE[SP].VC =  STORE[SP+1].VC;  I      254
0798:                         2: STORE[SP].VB := STORE[SP].VR=STORE[SP+1].VR;   INT    701
0799:                         3: STORE[SP].VB := STORE[SP].VB=STORE[SP+1].VB;   INT    702
0800:                         4: STORE[SP].VB := STORE[SP].VS=STORE[SP+1].VS;   INT    703
0801:                         5: BEGIN  COMPARE;                                INT    704
0802:                               STORE[SP].VB := B;                          INT    705
0803:                            END;                                           INT    706
0804:                       END; (*CASE P*)                                   INT    707
0805:                  END;                                                   INT    709
0806:                                                                         INT    710
0807:           18 (*NEQ*):BEGIN  SP := SP-1;                                 INT    711
0808:                       CASE P OF                                         INT    712
0809:                        0: STORE[SP].VB := STORE[SP].VA <> STORE[SP+1].VA;  I      255
0810:                        1: STORE[SP].VB := STORE[SP].VI <> STORE[SP+1].VI;  I      256
0811:                        6: STORE[SP].VB := STORE[SP].VC <> STORE[SP+1].VC;  I      257
0812:                         2: STORE[SP].VB := STORE[SP].VR<>STORE[SP+1].VR;  INT    714
0813:                         3: STORE[SP].VB := STORE[SP].VB<>STORE[SP+1].VB;  INT    715
0814:                         4: STORE[SP].VB := STORE[SP].VS<>STORE[SP+1].VS;  INT    716
0815:                         5: BEGIN  COMPARE;                                INT    717
0816:                               STORE[SP].VB := NOT B;                      INT    718
0817:                            END                                            INT    719
0818:                       END; (*CASE P*)                                   INT    720
0819:                  END;                                                   INT    722
0820:                                                                         INT    723
0821:           19 (*GEQ*):BEGIN  SP := SP-1;                                 INT    724
0822:                       CASE P OF                                         INT    725
0823:                        0: ERRORI(' <,<=,>,>= FOR ADDRESS   ');           I      258
0824:                        1: STORE[SP].VB := STORE[SP].VI >= STORE[SP+1].VI;  I      259
0825:                        6: STORE[SP].VB := STORE[SP].VC >= STORE[SP+1].VC;  I      260
0826:                         2: STORE[SP].VB := STORE[SP].VR>=STORE[SP+1].VR;  INT    727
0827:                         3: STORE[SP].VB := STORE[SP].VB>=STORE[SP+1].VB;  INT    728
0828:                         4: STORE[SP].VB := STORE[SP].VS>=STORE[SP+1].VS;  INT    729
0829:                         5: BEGIN COMPARE;                                 INT    730
0830:                              STORE[SP].VB := B OR                         I      261
0831:                                (STORE[I1+I].VI >= STORE[I2+I].VI)         I      262
0832:                            END                                            INT    732
0833:                       END; (*CASE P*)                                   INT    733
0834:                  END;                                                   INT    735
0835:                                                                         INT    736
0836:           20 (*GRT*):BEGIN  SP := SP-1;                                 INT    737
0837:                       CASE P OF                                         INT    738
0838:                        0: ERRORI(' <,<=,>,>= FOR ADDRESS   ');           I      263
0839:                        1: STORE[SP].VB := STORE[SP].VI >  STORE[SP+1].VI;  I      264
0840:                        6: STORE[SP].VB := STORE[SP].VC >  STORE[SP+1].VC;  I      265
0841:                         2: STORE[SP].VB := STORE[SP].VR>STORE[SP+1].VR;   INT    740
0842:                         3: STORE[SP].VB := STORE[SP].VB>STORE[SP+1].VB;   INT    741
0843:                         4: ERRORI(' SET INCLUSION          ');           INT    742
0844:                         5: BEGIN  COMPARE;                                INT    743
0845:                              STORE[SP].VB := NOT B AND                    I      266
```

```
0846:                              (STORE[I1+I].VI > STORE[I2+I].VI)                I     267
0847:                           END                                                 INT   745
0848:                      END; (*CASEP*)                                           INT   746
0849:                 END;                                                          INT   748
0850:                                                                               INT   749
0851:           21 (*LEQ*):BEGIN  SP := SP-1;                                       INT   750
0852:                      CASE P OF                                                INT   751
0853:                       0: ERRORI(' <,<=,>,>= FOR ADDRESS   ');                 I     268
0854:                       1: STORE[SP].VB := STORE[SP].VI <= STORE[SP+1].VI;      I     269
0855:                       6: STORE[SP].VB := STORE[SP].VC <= STORE[SP+1].VC;      I     270
0856:                        2: STORE[SP].VB := STORE[SP].VR<=STORE[SP+1].VR;       INT   753
0857:                        3: STORE[SP].VB := STORE[SP].VB<=STORE[SP+1].VB;       INT   754
0858:                        4: STORE[SP].VB := STORE[SP].VS<=STORE[SP+1].VS;       INT   755
0859:                        5: BEGIN  COMPARE;                                     INT   756
0860:                             STORE[SP].VB := B OR                              I     271
0861:                               (STORE[I1+I].VI <= STORE[I2+I].VI)              I     272
0862:                           END;                                                INT   758
0863:                      END; (*CASE P*)                                          INT   759
0864:                 END;                                                          INT   761
0865:                                                                               INT   762
0866:           22 (*LES*):BEGIN  SP := SP-1;                                       INT   763
0867:                      CASE P OF                                                INT   764
0868:                       0: ERRORI(' <,<=,>,>= FOR ADDRESS   ');                 I     273
0869:                       1: STORE[SP].VB := STORE[SP].VI <  STORE[SP+1].VI;      I     274
0870:                       6: STORE[SP].VB := STORE[SP].VC <  STORE[SP+1].VC;      I     275
0871:                        2: STORE[SP].VB := STORE[SP].VR<STORE[SP+1].VR;        INT   766
0872:                        3: STORE[SP].VB := STORE[SP].VB<STORE[SP+1].VB;        INT   767
0873:                        5: BEGIN  COMPARE;                                     INT   768
0874:                             STORE[SP].VB := NOT B AND                         I     276
0875:                               (STORE[I1+I].VI < STORE[I2+I].VI)               I     277
0876:                           END                                                 INT   770
0877:                      END; (*CASE P*)                                          INT   771
0878:                 END;                                                          INT   773
0879:                                                                               INT   774
0880:                                                                               INT   775
0881:           23 (*UJP*):PC := Q;                                                 INT   776
0882:                                                                               INT   777
0883:           24 (*FJP*):BEGIN  IF NOT STORE[SP].VB THEN PC := Q;  SP := SP-1     INT   778
0884:                 END;                                                          INT   779
0885:                                                                               INT   780
0886:          25 (*XJP*): BEGIN                                                    I     278
0887:                       PC := STORE[SP].VI + Q;                                 I     279
0888:                       SP := SP-1                                              I     280
0889:                     END;                                                      I     281
0890:                                                                               INT   783
0891:           95 (*CHKA*) : IF (STORE[SP].VA<NP) OR                               I     282
0892:                            (STORE[SP].VA>(MAXSTR-Q)) THEN                     I     283
0893:                          ERRORI(' BAD POINTER VALUE       ');                I     284
0894:           96,97,98,99,                                                        I     285
0895:           26 (*CHK*):IF (STORE[SP].VI < STORE[Q-1].VI) OR                     I     286
0896:                         (STORE[SP].VI > STORE[Q].VI) THEN                     I     287
0897:                      ERRORI(' VALUE OUT OF RANGE      ');                    INT   785
0898:                                                                               INT   786
0899:           27 (*EOF*):BEGIN  I := STORE[SP].VI;                                INT   787
0900:                       IF I=INPUTADR THEN                                      INT   788
0901:                       BEGIN STORE[SP].VB := EOF(INPUT);                       INT   789
0902:                       END ELSE ERRORI(' CODE IN ERROR          ')             INT   791
0903:                 END;                                                          INT   792
0904:                                                                               INT   793
0905:           28 (*ADI*):BEGIN  SP := SP-1;                                       INT   794
0906:                      STORE[SP].VI := STORE[SP].VI + STORE[SP+1].VI            INT   795
0907:                 END;                                                          INT   796
0908:                                                                               INT   797
0909:           29 (*ADR*):BEGIN  SP := SP-1;                                       INT   798
0910:                      STORE[SP].VR := STORE[SP].VR + STORE[SP+1].VR            INT   799
```

```
0911:                   END;                                                    INT    800
0912:                                                                           INT    801
0913:            30 (*SBI*):BEGIN SP := SP-1;                                   INT    802
0914:                       STORE[SP].VI := STORE[SP].VI - STORE[SP+1].VI       INT    803
0915:                   END;                                                    INT    804
0916:                                                                           INT    805
0917:            31 (*SBR*):BEGIN  SP := SP-1;                                  INT    806
0918:                       STORE[SP].VR := STORE[SP].VR - STORE[SP+1].VR       INT    807
0919:                   END;                                                    INT    808
0920:                                                                           INT    809
0921:           32 (*SGS*): BEGIN                                               I      288
0922:                           STORE[SP].VS := [STORE[SP].VI]                  I      289
0923:                     END;                                                  I      290
0924:                                                                           INT    819
0925:            33 (*FLT*):BEGIN  STORE[SP].VR := STORE[SP].VI;                INT    820
0926:                   END;                                                    INT    822
0927:                                                                           INT    823
0928:            34 (*FLO*):BEGIN  STORE[SP-1].VR := STORE[SP-1].VI;            INT    824
0929:                   END;                                                    INT    826
0930:                                                                           INT    827
0931:            35 (*TRC*):BEGIN STORE[SP].VI := TRUNC(STORE[SP].VR);          INT    828
0932:                   END;                                                    INT    830
0933:                                                                           INT    831
0934:            36 (*NGI*):STORE[SP].VI := -STORE[SP].VI;                      INT    832
0935:                                                                           INT    833
0936:            37 (*NGR*):STORE[SP].VR := -STORE[SP].VR;                      INT    834
0937:                                                                           INT    835
0938:            38 (*SQI*):STORE[SP].VI := SQR(STORE[SP].VI);                  INT    836
0939:                                                                           INT    837
0940:            39 (*SQR*):STORE[SP].VR := SQR(STORE[SP].VR);                  INT    838
0941:                                                                           INT    839
0942:            40 (*ABI*):STORE[SP].VI := ABS(STORE[SP].VI);                  INT    840
0943:                                                                           INT    841
0944:            41 (*ABR*):STORE[SP].VR := ABS(STORE[SP].VR);                  INT    842
0945:                                                                           INT    843
0946:            42 (*NOT*):STORE[SP].VB := NOT STORE[SP].VB;                   INT    844
0947:                                                                           INT    845
0948:            43 (*AND*):BEGIN  SP := SP-1;                                  INT    846
0949:                       STORE[SP].VB := STORE[SP].VB AND STORE[SP+1].VB     INT    847
0950:                   END;                                                    INT    848
0951:                                                                           INT    849
0952:            44 (*IOR*):BEGIN  SP := SP-1;                                  INT    850
0953:                       STORE[SP].VB := STORE[SP].VB OR STORE[SP+1].VB      INT    851
0954:                   END;                                                    INT    852
0955:                                                                           INT    853
0956:            45 (*DIF*):BEGIN  SP := SP-1;                                  INT    854
0957:                       STORE[SP].VS := STORE[SP].VS - STORE[SP+1].VS       INT    855
0958:                   END;                                                    INT    856
0959:                                                                           INT    857
0960:            46 (*INT*):BEGIN  SP := SP-1;                                  INT    858
0961:                       STORE[SP].VS := STORE[SP].VS *  STORE[SP+1].VS      INT    859
0962:                   END;                                                    INT    860
0963:                                                                           INT    861
0964:            47 (*UNI*):BEGIN  SP := SP-1;                                  INT    862
0965:                       STORE[SP].VS := STORE[SP].VS +  STORE[SP+1].VS      INT    863
0966:                   END;                                                    INT    864
0967:                                                                           INT    865
0968:            48 (*INN*): BEGIN                                              I      291
0969:                         SP := SP - 1; I := STORE[SP].VI;                  I      292
0970:                         STORE[SP].VB := I IN STORE[SP+1].VS;              I      293
0971:                   END;                                                    INT    876
0972:                                                                           INT    877
0973:            49 (*MOD*):BEGIN  SP := SP-1;                                  INT    878
0974:                       STORE[SP].VI := STORE[SP].VI MOD STORE[SP+1].VI     INT    879
0975:                   END;                                                    INT    880
```

```
0976:                                                                          INT    881
0977:           50 (*ODD*):BEGIN  STORE[SP].VB := ODD(STORE[SP].VI);           INT    882
0978:                      END;                                                INT    884
0979:                                                                          INT    885
0980:           51 (*MPI*):BEGIN  SP := SP-1;                                  INT    886
0981:                        STORE[SP].VI := STORE[SP].VI * STORE[SP+1].VI     INT    887
0982:                      END;                                                INT    888
0983:                                                                          INT    889
0984:           52 (*MPR*):BEGIN  SP := SP-1;                                  INT    890
0985:                        STORE[SP].VR := STORE[SP].VR * STORE[SP+1].VR     INT    891
0986:                      END;                                                INT    892
0987:                                                                          INT    893
0988:           53 (*DVI*):BEGIN  SP := SP-1;                                  INT    894
0989:                        STORE[SP].VI := STORE[SP].VI DIV STORE[SP+1].VI   INT    895
0990:                      END;                                                INT    896
0991:                                                                          INT    897
0992:           54 (*DVR*):BEGIN  SP := SP-1;                                  INT    898
0993:                        STORE[SP].VR := STORE[SP].VR/STORE[SP+1].VR       INT    899
0994:                      END;                                                INT    900
0995:                                                                          INT    901
0996:           55 (*MOV*):BEGIN I1 := STORE[SP-1].VA;                         I      294
0997:                        I2 := STORE[SP].VA; SP := SP-2;                   I      295
0998:                         FOR I := 0 TO Q-1 DO STORE[I1+I] := STORE[I2+I]  INT    903
0999:                        (* Q IS A NUMBER OF STORAGE UNITS *)              INT    904
1000:                      END;                                                INT    905
1001:                                                                          INT    906
1002:           56 (*LCA*):BEGIN SP := SP+1;                                   I      296
1003:                        STORE[SP].VA := Q;                                I      297
1004:                      END;                                                INT    910
1005:                                                                          INT    911
1006:           100,101,102,103,104,                                           I      298
1007:           57 (*DEC*): BEGIN                                              I      299
1008:                          STORE[SP].VI := STORE[SP].VI-Q;                 I      300
1009:                       END;                                               I      301
1010:                                                                          INT    913
1011:           58 (*STP*):INTERPRETING := FALSE;                              INT    914
1012:                                                                          INT    915
1013:           59(*ORD*): (*ONLY USED TO CHANGE THE TAGFIELD*)                 I      302
1014:                      BEGIN                                               I      303
1015:                      END;                                                I      304
1016:                                                                          I      305
1017:           60(*CHR*): BEGIN                                               I      306
1018:                      END;                                                I      307
1019:                                                                          I      308
1020:           61(*UJC*): ERRORI(' CASE - ERROR           ');                 I      309
1021:      END                                                                 I      310
1022:    END; (*WHILE INTERPRETING*)                                           I      311
1023:                                                                          I      312
1024:  1 :                                                                     INT    949
1025:  END.                                                                    INT    950

THAT'S ALL FOLKS!      LINES: 1025   CHARACTERS: 92194
```