

```

let FillProcVec[] be
§10
    ProcVec↓0 := P_b
5    ProcVec↓'p' := P_b
    ProcVec↓'4' := P_b
    ProcVec↓'s' := P_b
    ProcVec↓'_' := P_b
    for i = '0' to '9' do ProcVec↓i := P_d
10   for i = 'a' to 'z' do ProcVec↓i := P_l
    ProcVec↓'*' := CharacterRt
    ProcVec↓148 := CharacterRt
    ProcVec↓20 := StringRt
    ProcVec↓'*' := StringRt
15   ProcVec↓'[' := StringRt
    for i = 'A' to 'Z' do ProcVec↓i := P_u
    for i = 'A' to 'Z' do ProcVec↓i := P_v
    for i = 'a' to 'z' do ProcVec↓i := P_v
    ProcVec↓'*n' := NLRt
20   ProcVec↓'↗' := P_22
    ProcVec↓'√' := P_26
    ProcVec↓'×' := P_27
    ProcVec↓'≠' := P_29
    ProcVec↓'↓' := P_31
25   ProcVec↓'(' := P_40
    ProcVec↓')' := P_41
    ProcVec↓'**' := P_42
    ProcVec↓'+' := P_43
    ProcVec↓',' := P_44
30   ProcVec↓'-' := P_45
    ProcVec↓'/' := P_47
    ProcVec↓':' := P_58
    ProcVec↓';' := P_59
    ProcVec↓'<' := P_60
35   ProcVec↓'=' := P_61
    ProcVec↓'>' := P_62
    ProcVec↓'[' := P_91
    ProcVec↓']' := P_93
    ProcVec↓'^' := P_94
40   ProcVec↓'§' := P_123
    ProcVec↓'|' := P_124
    ProcVec↓'§' := P_125
    ProcVec↓' ' := P_126
    ProcVec↓'≠' := P_157
45   ProcVec↓'8' := P_184
    ProcVec↓'≤' := P_188
    ProcVec↓'=' := P_189
    ProcVec↓'>' := P_190
    ProcVec↓'a' := P_225
50   ProcVec↓'b' := P_226
    ProcVec↓'c' := P_227
    ProcVec↓'d' := P_228
    ProcVec↓'e' := P_229
    ProcVec↓'f' := P_230
55   ProcVec↓'g' := P_231
    ProcVec↓'i' := P_233
    ProcVec↓'l' := P_236
    ProcVec↓'m' := P_237
    ProcVec↓'o' := P_239
60   ProcVec↓'r' := P_242
    ProcVec↓'s' := P_243

```

```

ProcVec↓‘t’ := P_244
ProcVec↓‘u’ := P_245
ProcVec↓‘v’ := P_246
65 ProcVec↓‘w’ := P_247
$10

and P_b[] be || Ch is CAT_b
70 $20
NextCh[]
while Is[Ch, CAT_b] do NextCh[]
$20

75
and P_d[] be || Ch is CAT_d
$30
unless DecimalFn[] return
Send[NUMBER, Adjoin_to[NumTable] + 500]
80 $30

and P_l[] be || Ch is CAT_l
$40
85 Send[NAME, Ch - (‘a’ - 1)]
NextCh[]
$40

90 and P_u[] be || Ch is CAT_u
$70
PushCh_NextCh[]
while Is[Ch, CAT_a] do PushCh_NextCh[]
if TooBig[] do
95 $ Error[61, HARD, DUMMY, NullProgram]
Send[ERROR, 0]
ResetStack[]
return
$
100 Send[NAME, InsertName[]]
$70

and P_v[] be || Ch is CAT_v
105 $80
Error[56, HARD, Ch, Delete, CAT_v]
Send[ERROR, 0]
$80

110
and P_22[] be || Ch is ‘↗’
$90
NextCh[]
Send[COND, 0]
115 $90

and P_26[] be || Ch is ‘√’
$100
120 NextCh[]
Send[LOGOR, 0]
$100

125 and P_27[] be || Ch is ‘×’
$110

```

```

        NextCh[]
        Send[MULT, 0]
    §110
130

    and P_29[] be                || Ch is '≠'
    §120
        NextCh[]
135    Send[NE, 0]
    §120

    and P_31[] be                || Ch is '↓'
140 §130
        NextCh[]
        Send[VECAP, 0]
    §130

145
    and P_40[] be                || Ch is '('
    §140
        NextCh[]
        Send[RBRA, 0]
150 §140

    and P_41[] be                || Ch is ')'
    §150
155    NextCh[]
        Send[RKET, 0]
    §150

160 and P_42[] be                || Ch is '**'
    §160
        NextCh[]
        if Ch = '[' do
    §161
165    NextCh[]
        Send[VECAP, 0]
        return
    §161
        Error[78, SOFT, DUMMY, NullProgram]
170    Send[MULT, 0]
    §160

    and P_43[] be                || Ch is '+'
175 §170
        NextCh[]
        Send[PLUS, 0]
    §170

180
    and P_44[] be                || Ch is ', '
    §180
        NextCh[]
        Send[COMMA, 0]
185 §180

    and P_45[] be                || Ch is '-'
    §190
190    NextCh[]
        Send[MINUS, 0]

```

```

§190

195 and P_47[] be                                || Ch is '/'
    §200
        NextCh[]
        Send[DIV, 0]
    §200
200

    and P_58[] be                                || Ch is ':'
    §210
        NextCh[]
205    switchon Ch into
        §211
            case '=':
                NextCh[]
                Send[ASS, 0]
210                return

                default:Send[COLON, 0]
                return
        §211
215 §210

    and P_59[] be                                || Ch is ';'
    §220
220    NextCh[]
        Send[SEMICOLON, 0]
    §220

225 and P_60[] be                                || Ch is '<'
    §230
        NextCh[]
        Send[LT, 0]
    §230
230

    and P_61[] be                                || Ch is '='
    §240
235    NextCh[]
        Send[EQ, 0]
    §240

    and P_62[] be                                || Ch is '>'
240 §250
        NextCh[]
        Send[GT, 0]
    §250

245

    and P_91[] be                                || Ch is '['
    §260
        NextCh[]
        Send[SBRA, 0]
250 §260

    and P_93[] be                                || Ch is '['
    §270
255    NextCh[]
        Send[SKET, 0]

```

```

$270

260 and P_94[] be                                     || Ch is '^'
    §280
        NextCh[]
        Send[LOGAND, 0]
    §280
265

    and P_123[] be                                     || Ch is '§'
    §290
        NextCh[]
270    while Is[Ch, CAT_a] do PushCh_NextCh[]
        MSCRt[]
        if TooBig[] do
            §    Error[62, HARD, DUMMY, NullProgram]
            Send[ERROR, 0]
275        ResetStack[]
            return
        §
        Send[SECTBRA, InsertName[]]
    §290
280

    and P_124[] be                                     || Ch is '|'
    §300
        NextCh[]
285    if Wrong['|'] do
            §    Error[63, HARD, DUMMY, NullProgram]
            Send[ERROR, 0]
            return
        §
290    while Is[Ch, CAT_c] do NextCh[]
    §300

    and P_125[] be                                     || Ch is '$'
295 §310
        NextCh[]
        while Is[Ch, CAT_a] do PushCh_NextCh[]
        if TooBig[] do
            §    Error[62, HARD, DUMMY, NullProgram]
300        Send[ERROR, 0]
            ResetStack[]
            return
        §
        Send[SECTKET, InsertName[]]
305 §310

    and P_126[] be                                     || Ch is ' '
    §320
310    NextCh[]
        Send[NOT, 0]
    §320

315 and P_157[] be                                     || Ch is '≠'
    §330
        NextCh[]
        Send[NEQV, 0]
    §330
320

```

```

    and P_184[] be                                     || Ch is '8'
§340
    NextCh[]
325    while Is[Ch, CAT_b] do NextCh[]
        unless Is[Ch, CAT_o] do
            §    Error[64, HARD, Ch, NullProgram]
                Send[ERROR, 0]
            return
330    §
        unless OctalFn[] return
        Send[NUMBER, Adjoin_to[NumTable] + 500]
§340

335
    and P_188[] be                                     || Ch is '≤'
§350
    NextCh[]
    Send[LE, 0]
340 §350

    and P_189[] be                                     || Ch is '≡'
§360
345    NextCh[]
        Send[EQV, 0]
§360

350 and P_190[] be                                     || Ch is '≥'
§370
    NextCh[]
    Send[GE, 0]
§370
355

****

```