

Print Reports 2
*** Compilers 16/2/77

```
manifest
$m
    Write2[Ch, S] is
5    §    OutputSymbol[Ch]
        WriteS[S]
    $

    Write3[S1, Ch, S2] is
10    §    WriteS[S1]
        OutputSymbol[Ch]
        WriteS[S2]
    $

    Write4[Ch1, Ch2] is
15    §    OutputSymbol[Ch1]
        WriteS[' found where ']
        OutputSymbol[Ch2]
        WriteS[' expected']
20    $

    WriteTM[S] is
        WriteS['Too many *S', S]

    WriteSR[S] is
25    WriteS['Soft report: *S', S]

    WriteGF[S] is
        WriteS['get file *S', S]
30

    WriteSubs[n, S] is
    §    WriteS[n = 1 → '*n*4*4second *S ', '*n*4*4subsequent *Ss ', S]
        Room := (n = 1 → 44, 39)
    $
35

    Times[n] is
        WriteS[n = 2 → 'twice', '*N times', n]
$m

40 let ErrorString[i, n] be
    §ES
        let Ch = ChVal[i]
        WriteS['*n*4']
        switchon ErrorVal[i] into
45    §S
        case 1: WriteS['Too many pass *N error reports', PassVal[i]]
            endcase

        case 2: WriteS['Structure Size exceeded']
50        endcase

        case 3: WriteTM['newlines in text']
            endcase

        case 5: WriteS['End of text found but not expected']
55        endcase

        case 6: WriteS['End of text expected but not found']
            endcase
60

        case 7: WriteS['Too much code']
```

```

        endcase

65    case 8: WriteS['Subsidiary vector overflow']
        endcase

        case 9: WriteS['Not enough free store']
        endcase

70    case 10: WriteS['Too much data']
        endcase

        case 11: WriteS['Too many pass *N reports - soft reports
75                *signed, starting', PassVal[i]]
        endcase

        case 20: WriteSR['too many globals set for list']
        endcase

80    case 21: WriteSR['item with ' ]
        Write2[Ch, ' sets system global']
        endcase

        case 22: WriteS['global *N set ', Ch]
85        Times[n + 1]
        return

        case 23: Write3['global with ', Ch, ' set ' ]
        Times[n + 1]
90        WriteSubs[n, 'time']
        return

        case 51: WriteTM['names']
        endcase

95    case 52: WriteTM['numbers']
        endcase

        case 53: WriteTM['strings']
100    endcase

        case 55: WriteS['Invalid character *'*_C*',', Ch]
        endcase

105    case 56: WriteS['Invalid symbol, starting *'*_C*',', Ch]
        endcase

        case 57: WriteS['Invalid symbol *'***_C*' in string', Ch]
        endcase

110    case 58: WriteS['String too long']
        endcase

        case 59: WriteS['End of text in string']
115    endcase

        case 60: WriteTM['characters in names']
        endcase

120    case 61: WriteS['Name too long']
        endcase

        case 62: WriteS['Tag too long']
        endcase

125    case 63: WriteS['Unpaired comment bar']

```

```

        endcase

130    case 64:WriteS['Invalid character *'*C*' after 8', Ch]
        endcase

        case 65:WriteTM['gets']
        endcase

135    case 66:WriteS['Invalid get param']
        endcase

        case 67:WriteTM['get params']
        endcase

140    case 68:WriteS['No get params']
        endcase

        case 69:WriteTM['nested gets']
145    endcase

        case 70:WriteSR['get file empty']
        endcase

150    case 71:WriteS['get index not found']
        endcase

        case 72:WriteGF['not found']
        endcase

155    case 73:WriteS['String in character quotes']
        endcase

        case 74:WriteS['Invalid char after **8']
160    endcase

        case 75:WriteS['StringVec full']
        endcase

165    case 76:WriteGF['deleted']
        endcase

        case 77:WriteGF['archived']
        endcase

170    case 78:WriteSR['star used for multiplication']
        endcase

        case 101:
175    Write2[Ch, ' found out of context in expression']
        endcase

        case 102:
        Write2[Ch, ' not expected to delimit expression']
180    endcase

        case 103:
        Write2[Ch, ' not expected to start command']
        endcase

185    case 104:
        Write2[Ch, ' not expected to delimit command']
        endcase

190    case 105:
        Write2[Ch, ' found out of context']

```

```

                endcase

195      case 106:
          Write2[Ch, ' out of context in manifest expression']
          endcase

200      case 107:
          WriteSR['closing section bracket *']
          Write2[Ch, '*' inserted']
          endcase

205      case 111:
          Write4[Ch, SECTBRA]
          endcase

210      case 112:
          Write4[Ch, ASS]
          endcase

215      case 113:
          Write4[Ch, COLON]
          endcase

220      case 114:
          Write4[Ch, COMMA]
          endcase

225      case 115:
          Write4[Ch, RKET]
          endcase

230      case 116:
          Write4[Ch, SKET]
          endcase

235      case 118:
          Write4[Ch, SBRA]
          endcase

240      case 122:
          Write4[Ch, EQ]
          endcase

245      case 130:
          Write4[Ch, DO]
          endcase

250      case 131:
          Write4[Ch, INTO]
          endcase

255      case 132:
          Write4[Ch, TO]
          endcase

260      case 142:
          Write2[Ch, ' found where ; or $ expected']
          endcase

265      case 144:
          Write4[Ch, THEN]
          endcase

270      case 145:
          Write4[Ch, OR]

```

```

                endcase

260    case 181:
        WriteS['Invalid L-mode expression']
        endcase

265    case 191:
        Write2[Ch, ' found where operator expected']
        endcase

        case 192:
        Write2[Ch, ' found where operand expected']
        endcase
270

        case 201:
        Write2[Ch, ' found where name expected']
        endcase

275    case 202:
        Write2[Ch, ' declared more than once on the same level']
        Nce[n]
        WriteSubs[n, 'time']
        return
280

        case 203:
        Write2[Ch, ' used but not declared']
        endcase

285    case 204:
        Write3['manifest ', Ch, ' used in l-mode']
        endcase

        case 205:
290    WriteS['Invalid constant expression']
        endcase

        case 206:
295    WriteSR['scope clash for items with ']
        OutputSymbol[Ch]
        Nce[n]
        WriteSubs[n, 'defn']
        return

300    case 251:
        WriteS['case not in switch']
        endcase

        case 252:
305    WriteS['endcase not in switch']
        endcase

        case 253:
310    WriteS['break not in loop']
        endcase

        case 254:
        WriteS['resultis not in valof block']
        endcase
315

        case 255:
        WriteS['default not in switch']
        endcase

320    case 256:
        WriteS['*_N identical case constants in switch', n + 1]

```

```

        WriteSubs[n, 'time']
        return

325     case 257:
        WriteS['loop not in loop']
        endcase

        case 258:
330     WriteS['*N defaults found in switch', n + 1]
        WriteSubs[n, 'time']
        return

        case 259:
335     WriteS['return used in function']
        endcase

        case 260:
340     WriteS['No resultis in valof block*C ',
        n = 1 → DUMMY, 's']
        Room := 21
        return

        case 261:
345     WriteS['Global number *N out of range', Ch]
        endcase

        case 262:
350     WriteS['vec parameter negative']
        endcase

        case 270:
        WriteS['Manifest function used as routine']
        endcase
355

        case 271:
        WriteS['Manifest routine used as function']
        endcase

360     case 301:
        WriteS['Unequal sides in local declaration']
        endcase

        case 302:
365     WriteS['Unequal sides in assignment command']
        endcase

        case 303:
370     WriteS['More than one name in vec declaration']
        endcase

        case 350:
        WriteSR['manifest fn called with']
        WriteS[' *N too *S params', Ch > 0 → Ch, -Ch,
375     Ch > 0 → 'many', 'few']
        endcase

        case 351:
380     WriteSR['manifest definition buffer full']
        endcase

        case 352:
        WriteSR['manifest procedure with ']
        Write2[Ch, '*n*4*4too big for open code']
385     endcase

```

```

        case 353:
            WriteSR['manifest procedure with ']
            Write2[Ch, '*n*4*4used (not applied)']
390         endcase

        default:WriteS['Unknown report *N ', ErrorVal[i]]
            OutputSymbol[Ch]
            Write[' ','']
395         endcase

        $S
        Nce[n]
        WriteS[n = 1 → ' ', '*n*4*4']
        Room := 56
400 $ES

```
