

Print Reports 1

\*\*\* Compilers 20/12/75

static Room = 0

```
5 let PrintReports[] be
  §PR
    SetupTables[]
    DoReports[WorkFile['Error Reports']]
    unless (Mode ^ 2) = 0 do PrintGlobList[]
10  ClosedownTables[]
    §PR

  and SetupTables[N] be
15  §ST
    Output := ReportStream
    NameTable := SetUpTable['Names', 0]
    StringTable := SetUpTable['Strings', 0]
    GetTable := SetUpTable['Gets', 1]
20  NumberTable := VecfromFile[WorkFile['Numbers']] + 1
    §ST

  and SetUpTable[WF, a] = valof
25  §ST
    let v = VecfromFile[WorkFile[WF]] + 1
    let Ptr = v + 1
    and n = v↓0
    let Tab = ProperVec[n]
30  for i = 1 to n do
    §  Tab↓i := Ptr
      Ptr := NextString[Ptr + a]
    §
    resultis Tab
35  §ST

  and ClosedownTables[] be
    §CT
40  ReturnTable[NumberTable]
    ReturnTable[StringTable]
    ReturnTable[GetTable]
    ReturnTable[NameTable]
    §CT
45

  and DoReports[E] be
    §DR
    SetupVecs[E]
50  OutputReports[]
    §DR

  and SetupVecs[E] be
55  §SV
    let v = VecfromFile[E]
    NoReports := v↓0 / 5
    ErrorVec := ProperVec[NoReports]
    LineVec := ProperVec[ReportSize]
60  for i = 1 to NoReports do ErrorVec↓i := v + (5 × i) - 4
    Quicksort[ErrorVec, 1, NoReports, Gtr]
```

\$SV

65 and Gtr[p, q] =  $p \downarrow 1 \neq q \downarrow 1 \rightarrow p \downarrow 1 > q \downarrow 1$ ,  $p \downarrow 2 \neq q \downarrow 2 \rightarrow p \downarrow 2 > q \downarrow 2$ ,  $p > q$

and OutputReports[] be

§OR

```
70   let ReportsPrinted = 0
      if NoReports = 0 return
      WriteS['*n*n*N REPORT*C*n', NoReports, NoReports = 1 → ' ', 'S']
      §R
        let i = ReportsPrinted + 1
75     let G = GetVal[i]
        test G = 0
          ifso WriteS['*nIn Main Text:*n']
          ifnot WriteS['*nIn get file *'*S*':*n', FileTitle[G]]
        SetupGIn[G]
80     while i ≤ NoReports ∧ GetVal[i] = G do
      §   unless AlreadyOutput[ErrorVal[i]] do PrintError[i]
          i := i + 1
      §
        WriteS['*n*n']
85     i := ReportsPrinted + 1
        while i ≤ NoReports ∧ GetVal[i] = G do
      §   OutputLine[LineVal[i]]
          i := i + 1
      §
90     ReportsPrinted := i - 1
        Close[GIn]
      §R repeatwhile ReportsPrinted < NoReports
```

§OR

```
95   and PrintError[i] be
      §MES
        let p, q = 1, 1
        and G, E, C, L = GetVal[i], ErrorVal[i], ChVal[i], LineVal[i]
100    and MultLine = false
        LineVec↓1 := L
        for j = i + 1 to NoReports do
          if GetVal[j] = G ∧ ErrorVal[j] = E ∧ ChVal[j] = C do
            §   p := p + 1
105            LineVec↓p := LineVal[j]
                UpdateErrorVal[j, 0]
                unless LineVal[j] = L do MultLine := true
            §
110            LineVec↓(p + 1) := 0
                ErrorString[i, p]
            if L = 0 return
            WriteS['on line']
            if MultLine do Write['s']
            while q ≤ p do
115          §   let l = LineVec↓q
                and r = 0
                while LineVec↓q = l do q, r := q + 1, r + 1
                if Room < (r = 1 → 5, 15) do
                  §   WriteS['*n*4*4']
120                  Room := 64
                §
                WriteS[' *N', l]
                Nce[r]
                if q ≤ p do Write[' ','']
125          Room := Room - 5
```

§

\$MES

```
130 and Nce[r] be
    §N
        switchon r into
        §S
            case 1: return
135
            case 2: WriteS[' (twice)']
                Room := Room - 8
                return

140        default: WriteS[' (*N times)', r]
            Room := Room - 10

        §S
    §N

145 and SetupGIn[G] be
    §SI
        let Ch = 0
        GIn := G = 0 → In, BytesfromWords[InfromFile[FileNo[G]]]
150    Ch := Next[GIn] repeatwhile Ch = '*n'
        PutBack[GIn, Ch]
        LineNo := 1
    §SI

155 and OutputLine[L] be
    §OL
        let Ch = 0
        until LineNo > L do
160    §    Ch := Next[GIn]
            if Ch = '*n' do LineNo := LineNo + 1
            if Endof[GIn] do
                §    WriteS['End of text at line *N', LineNo]
                    LineNo := ENORMOUS
165    §
    §
        if LineNo > L return
        WriteS['*I4. ', LineNo]
        §    Ch := Next[GIn]
170        Write[Ch]
        § repeatuntil Ch = '*n' ∨ Endof[GIn]
        LineNo := LineNo + 1
    §OL

175 and VecfromFile[f] = valof
    §V
        let h = FindHeading[f]
        if Empty[h] do
180    §    ReturnHead[h]
        resultis ProperVec[0]
    §
        DisctoCore[DiscPage, LastPage[h]]
        §    let Unused = ENDOFDATAAREA - LastUsedWordPtr[DiscPage]
185        let n = (DATASIZE × NumberofPages[h]) - Unused
            ReturnHead[h]
        § let v = ProperVec[n]
            let S = InfromFile[f]
            TransferIn[S, v + 1, n]
190        Close[S]
        resultis v
```

```

$V

195 and ProperVec[n] = valof
    $P
        let m = MaxVecSize[]
        if m < n do
            § ReportS['Not enough free store for reports']
200         Finish[]
        §
        § let v = NewVec[n]
        v↓0 := n
        resultis v
205 $P

and PrintGlobList[] be
    $P
210     let n = GlobList↓0
        let Sys = GlobList↓(-1)
        let m = n > GlobSize → GlobSize, n
        ReportS['*n*N global*S set', n, AddS[n]]
        unless Sys = 0 do
215         ReportS['*4including *N system global*S', Sys, AddS[Sys]]
        for i = 1 to 2 × m by 2 do
            § let v = GlobList↓i || the sign bit is set for system globals
                ReportS['*I3: *S*S', (v ∧ 877777), NameTable↓(GlobList↓(i + 1)),
                    v < 0 → ' *****', '']
220         §
        if m < n do ReportS['*n etc.']
    $P

225 and AddS[n] = n = 1 → '', 's'
****

```