

Pass 3.4

\*\*\* Compilers 26/6/76

```

    let SetUpRA[] be
    §SRA
        SaveIn := In
5        BuffStr := ConstructStream[FASTBILATERAL]
        SetNextFn[BuffStr, ErrorNext]
        SetOutBlockRt[BuffStr, NullProgram]
        SetCloseRt[BuffStr, DestroyStream]
        SetUpVector[BuffStr, BUFFERSIZE]
10    § let Buffer = Vector[BuffStr] + 1
        SetInBuff[BuffStr, Buffer, Buffer]
        SetOutBuff[BuffStr, Buffer, Buffer + BUFFERSIZE]
    §SRA

15
    and CloseRA[] be
    §CRA
        Close[BuffStr]
    §CRA
20

    and ErrorNext[S] = valof
    §    In := SaveIn
        CompilerError[3401]
25    resultis 0
    §

    and RestoreIn[S] = valof
30 §RI
        SwapIn[]
        SetNextFn[S, ErrorNext]
        resultis Next[In]
    §RI
35

    and SwapIn[] be
    §S
        let IP, IE = InPtr, InEnd
40    test In = SaveIn
        ifso InPtr, In := 0, BuffStr
        ifnot InPtr, InEnd := InBuffPtr[BuffStr], InBuffEnd[BuffStr]
        test IP = 0
        ifso In := SaveIn
45    ifnot SetInBuff[BuffStr, IP, IE]
    §S

    and DealwithAss[] be
50 §DA
        let IP, IE = InPtr, InEnd
        and OBP, OBE = OutBuffPtr[BuffStr], OutBuffEnd[BuffStr]
        and Count = ValPart[Next[In]]
        and LeftLN, RightLN = LineNo, LineNo
55    and LeftGet, RightGet = GetNo, GetNo
        and LASSCt = 1
        until OutBuffPtr[BuffStr] ≥ OBE ∨ LASSCt = 0 do
        §u
            Ch := Next[In]
60    switchon TypePart[Ch] into
        §s
```

```

        case NEWLINE:
            RightLN := ValPart[Ch]
            endcase
65
        case GET:
            RightGet := ValPart[Ch]
            endcase

70
        case LASS:
            LASSCt := LASSCt + 1
            endcase

        case ASS:
75            LASSCt := LASSCt - 1
            endcase

        $s
        Out[BuffStr, Ch]

    $u
80    InPtr, InEnd := OBP, OutBuffPtr[BuffStr]
    test Ch = ASS
        ifso §    let Delim = 0
                for i = 1 to Count do
                    §f
85                    LineNo, GetNo := RightLN, RightGet
                    Expression[]
                    test i = Count
                    ifso Delim := Ch
                    ifnot unless Ch = COMMA do CompilerError[3402]
90                    RightLN, RightGet := LineNo, GetNo
                    SwapIn[]
                    LineNo, GetNo := LeftLN, LeftGet
                    LeftExpression[]
                    unless Ch = (i = Count → ASS, COMMA) do
95                    CompilerError[3403]
                    LeftLN, LeftGet := LineNo, GetNo
                    SwapIn[]

                §f
                LineNo, GetNo := RightLN, RightGet
100                Ch := Delim
                ChType := TypePart[Delim]

        §
    ifnot §    let AssStart = HN
            and RSSP = SSP
105            and AN = HN
            SetNextFn[BuffStr, RestoreIn]
            SwapIn[]
            HN := HN + 2 × Count + 1
            §r
110            ForHop[GenHopNo[lv AN]]
            BackHopPt[GenHopNo[lv AN]]
            UpdateS[RSSP + 1]
            LeftExpression[]
            §r repeatwhile Ch = COMMA
115            unless Ch = ASS do CompilerError[3404]
            ForHop[GenHopNo[lv AN]]
            AN := AssStart
            ForHopPt[GenHopNo[lv AN]]
            §r
120            UpdateS[RSSP]
            Expression[]
            BackHop[GenHopNo[lv AN]]
            ForHopPt[GenHopNo[lv AN]]
            §r repeatwhile Ch = COMMA
125            UpdateS[RSSP]
            SetNextFn[BuffStr, ErrorNext]

```

```

        $
        InPtr, InEnd := IP, IE
        SetOutBuff[BuffStr, OBP, OBE]
130 $DA
```

```

        and GenHopNo[Loc] = valof
        §GH
135      rv Loc := rv Loc + 1
        resultis rv Loc
        §GH
```

```
****
```