

```

103 Q = Q + (A(I,J) + A(I,K))*(A(I,J) - A(I,K))
    P = HALF P + HALF P
    ABSQ = ABS(Q)
    IF (ABSQ - EPS) 104, 105, 105
104 IF (Q) 105, 106, 106
105 ABSQ = ABS(Q)
    IF (ABSQ - ABSQ) 107, 107, 108
107 TAN = ABSQ/ABSQ
    COS = 1.0/SQRT(1.0 + TAN*TAN)
    SIN = TAN*COS
    GO TO 109
108 CTN = ABSQ/ABSQ
    SIN = 1.0/SQRT(1.0 + CTN*CTN)
    COS = CTN*SIN
109 COS = SQRT((1.0 + COS)/2.0)
    SIN = SIN/(COS + COS)
    IF(Q) 110, 111, 111
110 TEMP = COS
    COS = SIN
    SIN = TEMP
111 IF(P) 112, 113, 113
112 SIN = -SIN

```

```

113 DO 114 I = 1, N
    TEMP = A(I,J)
    A(I,J) = TEMP*COS + A(I,K)*SIN
114 A(I,K) = -TEMP*SIN + A(I,K)*COS
    NCOUNT = NN
    GO TO 102
106 NCOUNT = NCOUNT - 1
    IF(NCOUNT) 115, 115, 102
102 CONTINUE
    GO TO 116
115 DO 117 J = 1, N
    VALUE(J) = 0.0
    DO 118 I = 1, N
118 VALUE(J) = VALUE(J) + A(I,J)*A(I,J)
117 VALUE(J) = SQRT(VALUE(J))
    DO 119 J = 1, N
    DO 119 I = 1, N
119 A(I,J) = A(I,J)/VALUE(J)

```

Output the matrix $\underline{A}(I,J)$ of the unit-length column eigenvectors \underline{T} and the vector (diagonal matrix) $\underline{VALUE}(J)$ of eigenvalues λ .

END

Correspondence

To the Editor
The Computer Journal

Sir,
The FORTRAN complainants in February's Journal (letters of Palme, Hill, Finn, and Flavell), prompt me to ask all those who want FORTRAN to look like ALGOL to please stick to ALGOL; those who want more freedom of implementation, read on . . .

My feeling on the question of what FORTRAN should look like in the future is that we ought to concentrate on standardising implementations rather than the language.

Every major manufacturer likes to embellish his compiler with exclusive features, and in fact some of the complaints of Palme *et al.* are answered in this way. For example, ICL 1900 users have great freedom on DO loop expressions; UNIVAC 1100 users are warned of variables never assigned a value; IBM 360 users can suffix suffixes to their hearts content, and so on. To try to foster a FORTRAN V (say) incorporating all these features is dangerous—the extensions chosen by a manufacturer may use characteristic hardware features not generally available.

What industry and commerce needs is true compatibility of implementation of a series of FORTRAN versions of increasing ambition.

LEVEL A would be A.S.A. standard—no more, no less.

LEVEL B would introduce mixed-mode arithmetic, general suffixes.

LEVEL C might include completely general expressions at all points.

LEVEL D may include character and bit manipulation, perhaps interactive features, and so on. The details of each implementation must be the result of international agreement—the BCS could give an adequate lead.

The essential thing is to have complete machine interchangeability of useful FORTRAN compilers, and a level of implementation to suit every size and power of computer, so that none are held back

and conversely, no one over-extended to the point of inefficiency of object programs.

Nobody yet seems to have discussed what the compiler is supposed to do, apart from compile. Equally as important as the basic function are the additional tasks such as cross-listing, error detection and error reporting. There definitely ought to be a standard for ancillary compiler tasks.

Anything other than a LEVEL A compiler as defined above is sure to be a multi-pass job, so it is reasonable for all higher levels to conform at least to the following:

- All diagnostics written out above or below the offending line, in words, not just an error number.
- No limit on the number of diagnostics per line, and the compiler not to give up too easily!
- Warnings to be given on the use of non-initiated variables, ambiguous constructions (*vide* Palme) and similar fertile sources of logical errors, in addition to the usual syntactic errors.
- All variables, arrays and functions to be listed in alphabetical order, with type, length (if applicable) and place of first reference.
- All labels to be listed in numerical order, with an indication of use as a format or a statement number.

Consideration will show that these suggestions for ancillary tasks answer other complaints of Palme *et al.* Nor are they original; many having been incorporated in UNIVAC 1100 series machines for years.

Yours faithfully,
D. T. HALL

21 Bandywood Crescent
Kingstanding
Birmingham
B44 9JX
31 March 1972