

which this program originally came had not yet introduced subroutines when the example was presented.)

And what about its readability? Here is a letter to the editor of *Datamation*, published shortly after the original article.

In the October issue, [...] told us "How to Write a Readable Fortran Program." I wish they had followed their own advice in the example that they gave. Said example has so many comments in it that it is unreadable. I agree that program documentation is a long-neglected and important problem. And placing a comment card before each statement in a program *does* document it. It also makes the program unreadable. Grouping Fortran comments and program statements into logical blocks makes *both* of them readable with very little loss of clarity.

Neal Paris
Durham, North Carolina

Mr. Paris has a point. There are more comments than program; some of them convey little information. We use few comments in our programs — most of the programs are short enough to speak for themselves. And when a program cannot speak for itself, it is seldom the case that greater reliability or understanding will result by interposing yet another insulating layer of documentation between the code and reader.

Don't over-comment.

A second letter on the same subject also appeared in *Datamation*:

The example given in the article, "How to Write a Readable Fortran Program" (Oct., p 73), illustrates one of the most common faults of comments in programs — that the comments don't agree with the program. The program itself illustrates one of the commonest programming mistakes — the failure to check controlling parameters for limits.

Specifically, the last line of comment in the heading states reassuringly: "All input is checked for validity." So what happens? The very first READ statement reads four controlling parameters which are checked only by the field width in the format, not a very good way to do it. In particular, N is not checked for its limit of 80. The unchecked N is used to "check" the values of I and J. Hence, a too-large N may result in storing of data beyond the array bounds. The unchecked N also limits several DO loops in the program.

N. M. Taylor
Washington, D.C.

Ms. Taylor is also right. The comment, like the input checking, is slightly wrong.

As a final observation, not worth a letter to *Datamation*, try the program on the equations

$$\begin{aligned} Y &= 1.0 \\ X + Y &= 2.0 \end{aligned}$$

The solution is obviously $X=1.0$, $Y=1.0$. What does the program do? Following the code, when I equals one in the first pass through the inner loop, we evaluate (one line after statement 50)

$$TEMP = (A(I, NPLUS1) - SUM) / A(I, I)$$