

LETTER TO THE EDITOR

Sir—

The Appendix in a recent article by J. O. L. Roberts (*Bird-Banding*, 42(3): 181-184, 1971) prompts me to raise an issue that has festered in a corner of my cerebrum for some time: namely, in what *form* should computer programs be communicated? The issue is not *whether* they should be communicated: I find them an integral part of the methods of a paper, and also useful and timesaving when I tackle a related problem. The question is *how* to communicate their essence with greatest generality and efficiency.

The former convention was that all programs were published in the high-level language ALGOL, whether used by the author in that language or not. This convention recognized the widespread use of ALGOL, especially in Europe, where it was a sort of lingua franca of computing. ALGOL is a convenient language for science, in that it is more mathematically-oriented than FORTRAN, and is equally adaptable to interactive (teletype) and batch (card) uses in its various forms. The population explosion of IBM machines, however, led to the displacement of ALGOL by IBM's FORTRAN, especially in America. The influence is felt beyond IBM's own machines, moreover, as the newer generations of larger computers (such as the UNIVAC 1108 system) generally encourage a supported version of FORTRAN while merely retaining unsupported compilers for ALGOL and other high-level languages. This situation has, in turn, led to the recent trend of publishing programs in some version of FORTRAN.

The problems associated with publication in FORTRAN are several. First, there are many versions of FORTRAN, two common ones being the FORTRAN II of smaller IBM computers and the FORTRAN V of larger computers such as the UNIVAC 1108. The V version is so much freer than the II that it would be very misleading to publish in V; yet those using V may never have learned II and could not publish their programs in it even if "required" to do so.

Second, every version and subversion of a high-level language is slightly different, particularly in input/output commands, which are somewhat machine-dependent. Roberts (*op. cit.*) wisely stated that "In order to make the program as machine independent as possible, the input/output statements were simplified and no string manipulations were used." However, that procedure alone does not cure all the ills of general communication.

A third point to be made is that many shorter programs in particular (e.g., Buckley and Hancock, *Bird-Banding*, 39: 123-129, 1968) are more parsimoniously run interactively by those who have the facilities to do so. While there are limited possibilities for using FORTRAN V interactively (as CFOR in the 1108 executive system, for instance), most persons who have teletypes available will likely use General Electric's BASIC on one of their telephone-communicated, time-sharing computer systems available everywhere in the U. S. The essence of a FORTRAN algorithm may be unduly difficult to glean for convenient conversion to BASIC.

A fourth point is that a program in any language (e.g., Roberts' FORTRAN program) may contain a very large number of language-dependent statements that are really irrelevant to the basic algorithm, such as format statements, counters and incremental devices, initializing of variables and so on. If these and the comment statements were omitted from Roberts' program, for instance, it would probably be half its length, and yet still waste more than half of each page with blank paper.

Putting these problems together seems to lead inevitably to the conclusion that publication of a program in any high-level language is at best an awkward and inefficient means of communication. Newer texts in computer science avoid using any real language: some contrive a simple language (e.g., Forsythe *et al.*, *Computer Science: A First Course*, Wiley, N. Y., 553 pp.), while others rely solely on flowcharts to introduce computing.

My suggestion, then, is this: algorithms in published papers should be presented as flowcharts, using the standard IBM template flowcharting symbols and the real algebraic variables used in the paper. Flowcharts should be accompanied by statement (a) of what exact version of which language the program was written for which machine, and (b) that a print-out of the program will be made available to any reader writing the author. It would be far easier for me, for instance, to write Roberts' program in BASIC from a flowchart of its essence than from his actual program in an old version of FORTRAN, and it might even be easier to write it in FORTRAN V from such a flowchart. The flowchart has the further advantage of communicating the essential idea of the program to someone who has little familiarity with computing, and in any case is the most universal method for communicating the algorithm used.

While I do not call for such a policy to be enforced by editorial fiat before there is a general discussion of this problem, I do think that now is the time to come to some agreement on this issue. The frequency of published programs is increasing in *Bird-Banding* and many other journals, and the time to decide upon a uniform method of presentation is now, before chaos prevails.—Jack P. Hailman, Department of Zoology, University of Wisconsin, Madison 53706.

Ed. note: We agree that it would be helpful for authors to know in advance what format would be most useful to readers. Flowcharts could well be a major element, but we prefer to show the algorithms first. We agree that the offer of copies of the complete program would be useful for those able to handle or translate the particular language used.