

on "personal computing-1966" as this book already promises to do for computing and the basics of programming with respect to the older techniques.

One of the strong points of this volume is its pleasant pedagogical approach. When words alone do not suffice, a concrete example is used "to fix ideas." These examples illustrate the points being made quite adequately without exhausting the intelligent reader's patience or endurance. The authors manage quite nicely to increase the reader's cultural background through the use of anecdotes and historical sidelights that go well with the lesson. This is often characteristic of gifted authors, perhaps a little more frequently with British writers than with others. One wonders why this is so.

For style and lucidity of exposition and for its skill in pleasing, useful communication, as well as its content, this book is highly recommended.

The plea of G. H. Stearman [1] for the improvement of technical writing, with which this review strongly concurs, would be unnecessary if more of our colleagues wrote like the present authors.

HERBERT M. ERNST

Applied Mathematics Laboratory  
David Taylor Model Basin  
Washington, D. C.

1. G. H. STEARMAN, "Is switching theory mathematics or engineering?," *IEEE Trans. on Electronic Computers*, v. EC-15, 1966, p. 124.

112[Z].—SEYMOUR V. POLLACK, *A Guide to Fortran IV*, Columbia University Press, New York, 1965, 260 pp., 24 cm. Price \$5.00.

The text is written in a clear and lucid manner and contains a glossary of terms (the lack of which sometimes occurs in the best of texts of this type). The problem exercises are interesting but rather limited in scope (many having evolved from medical applications). The illustrations and flow charts are clear and well coordinated with the text, and a complete index is included. However, there is no discussion of the use of disc storage appropriate to some machine configurations and the material is scanty on the use and advantages of binary tape, an important medium for handling large amounts of data conveniently. The most serious criticism of the book concerns the complete lack of material concerned with the basic numerical problems associated with computing hardware. For example, when discussing the arithmetic IF statement, no material is presented on the use of a tolerance when comparing two floating point numbers. This presentation of the Fortran IV language appears to be geared toward researchers in the life sciences.

HOWARD ROBINSON

Courant Institute of Mathematical Sciences  
New York University  
New York, New York

113[Z].—CHARLES PHILIP LECHT, *The Programmer's Fortran II and IV*, McGraw-Hill Book Co., New York, 1966, xx + 162 pp., 28 cm. Price \$7.95.

This concise, compact book is a catalogue of the main features of and contrasts between Fortran II and Fortran IV. Other forms of Fortran such as that for the CDC 3600, for instance, are not included. It is characterized by a simple and uniform

arrangement, clear type, sharp explanations, and freedom from extraneous material. Although an index would enhance its value as a reference, a very well organized table of contents serves almost the same purpose. Its compactness is achieved in part by relegating definitions to a glossary, with some words in the glossary referred to other definitions in the glossary. However, a great deal of the substance of the book is thus partially buried.

Although the author carefully points out that this book is intended for programmers and is not a "self-teaching device," it is not clear for whom the book is written. The omission or scanty coverage of a number of fine, technical details hardly supports the author's claim to a complete reference for programmers. For instance, in some systems, such as the Fortran II compiler for the IBM 7094 used until recently at New York University, the value of the index of a *DO* loop after a normal exit is equal to one plus the upper limit. In other systems it is equal to the upper limit. This author disposes of this point by stating that, upon a normal exit, the index of a *DO* loop is not available, which is not even correct. A number of other technical details, especially concerning formats and *DO* loops, are left unexplained. If it is indeed intended for programmers, these details are the only real essentials. One uses a reference to look up the obscure, not the elementary and obvious.

This book is probably most useful for a beginning student of Fortran, to be used with other texts and manuals, much as a student of French uses a bilingual dictionary.

THAN PORTER

Courant Institute of Mathematical Sciences  
New York University  
New York, New York

114[Z].—J. ARSAC, A. LENTIN, M. NIVAT & L. NOLIN, *Algol, Théorie et Pratique*, Gauthier-Villars, Paris, 1965, 204 pp., 27 cm. Price 45 francs.

The title of the book (Algol—Theory and Practice) shows clearly that it is directed both to those interested in the theoretical aspects of algorithmic languages, and to application programmers who mainly want a reference manual.

The beginning (Chapters 1 and 2) gives the basic definitions and concepts concerning the structure of programs and the "ways of thought" responsible for the formulation of the Algol language. Although the material is very abstract, the authors have made a valuable effort to make the reading attractive by giving numerous examples.

The following chapters give a thorough description of the Algol language including a special chapter for Boolean expressions.

Finally, a whole chapter is devoted to examples and at the end a summary of the Algol syntax is given. This will probably be very helpful to those who need a reference manual.

The book should appeal to many people of different interests. Those interested in mechanical languages will appreciate the systematic presentation of the material. Advanced programmers will be interested in the detailed description of the Algol system and in the examples given. However, this is not an elementary book; far from that, beginning programmers should stay away from it lest they be con-