34 [Z].—PHILIP M. SHERMAN, *Programming and Coding Digital Computers*, John Wiley and Sons, Inc., New York, 1963, xiv + 444 p., 24 cm. Price $11.00.

This is an elementary text on digital computer programming. The author's intent is that "no prior knowledge of digital computers and no mathematical background beyond that which is ordinarily a part of the high school curriculum" is required of the student. By and large, his presentation achieves this objective. The style is crisp, clear, and direct. The topics are treated in simple terms. As in similar texts, machine language programming is explained by introducing a hypothetical "typical" computer. An assembler language is discussed and "coding fundamentals" such as loops, use of index registers, branching, use of subroutines, and input-output operations are explained. Finally, there is a discussion of algebraic languages (Fortran and Algol are treated briefly), non-numerical problems, macro-instructions, and program debugging.

Each chapter contains a fairly complete set of exercises.

E. K. BLUM

35 [Z].—H. L. COLMAN & C. SMALLWOOD, *Computer Language*, McGraw-Hill Book Co., Inc., New York, 1963, xiv + 196 + 14 p., 23 cm. Price $5.95.

This is an elementary autoinstructional text designed to teach FORTRAN programming to students "of almost any background or professional interest," to quote from the foreword. Considerable effort has evidently gone into the autoinstructional aspect of the text. The layout is quite different from the usual text and, according to the preface, it also differs from "traditional autoinstructional texts." Each page resembles a flow chart and consists of brief statements enclosed in boxes connected by arrows. Presumably, in the theory of autoinstruction it is shown that this method of guiding the reader's eye and mind is superior to the traditional page layout. The reviewer is unable to comment on this.

The table of contents is informative. There are eight parts to this booklet as follows: 1. Introduction, 2. Program Structure, 3. Variables and Constants, 4. Input Statements, 5. Arithmetic Expressions, 6. Arithmetic Statements, 7. Control Statements, and 8. Output Statements. The ordering of these eight parts is somewhat surprising. For example, one wonders at the presentation of the notion of subroutine in part 2, almost at the outset. However, the style throughout is clear and concise and, autoinstructional or not, it is a good brief introductory text.

E. K. BLUM

36 [Z].—JAMES A. SAXON, *Programming the IBM 7090: A Self-Instructional Programmed Manual*, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1963, xiv + 210 p., 24 cm. Price $6.75.

This book calls itself a "Self-Instructional Programmed Manual". It is a workbook that attempts to instruct the beginner in some of the rudiments of coding, using the question-answer technique of "programmed" instruction. The author tells the reader that by the time he has worked through the book "... you should feel confident in being able to pull your weight as a fledgeling programmer."

The book covers only the easier parts of 7090 coding. Input-output is touched on very briefly in lesson 10, which in essence states that input-output programming is outside the scope of the book. Lesson 4 indicates that floating-point arithmetic, except in barest outline, is also outside the scope of the book. Of the 13 floating-point instructions in the 7090 only 4 are mentioned. The FDP instruction (Floating Divide and Proceed) is omitted. The FDH instruction (Floating Divide and Halt), which should *never* be used by a programmer, is the only Floating Divide instruction presented here.

It is the opinion of the reviewer that a procedure-oriented language like Fortran should be used by the beginner in his first contact with the 7090. By the time he is ready to learn the details of 7090 machine language coding he should be much more than a beginner in the field of computer programming, and this text will not satisfy his needs.

Lesson 15, the last lesson in the book, presents the student with an example of a complete 7090 program. The example is poorly programmed. It is the kind of programming one should expect from the "fledgeling" programmer, but not from his instructor.

SAUL ROSEN

Computation Center
Purdue University
Lafayette, Indiana

37 [Z].—F. J. GRUENBERGER & D. D. MCCRACKEN, *Introduction to Electronic Computers, Problem Solving with the IBM 1620*, John Wiley & Sons, Inc., New York, 1963, vii + 170 p., 28 cm. Price $2.95.

The expressed purpose of this book is to have it serve as an introduction to computer programming by discussing examples of types of problems to which computers may be applied. The specific equipment used by way of illustration is the IBM 1620 (with paper tape input and output), and the book is designed to be used as an adjunct to the manual for the 1620. It is assumed in various parts of the book that the reader, through access to the computer itself, is able to resolve specific details of coding, although some of the essentials thereof are reviewed as the book progresses.

An introduction to organization and flow charting is first presented. Then, dealing in terms of machine language, the book explains the use of loops, subroutines, scaling of numerical data, and floating-point arithmetic. Illustrative problems are drawn from number theory. In addition, some examples of the use of random numbers and the playing of simple mathematical games are introduced. However, the book is addressed to a quite elementary level of reader—a background of high school mathematics is all that is required—and so the illustrations are necessarily of an elementary character. They do serve to convey some feeling for empirical tasks that computers can perform.

There is a brief discussion of assembly programming and debugging. The treatment of automatic programming, i.e., interpreters, compilers, and generators, is very sketchy.

A. SINKOV

University of Arizona
Tempe, Arizona