The lead-off paper in the book is by Bombieri and Davenport, "On the large sieve method". Most of the other papers are on number theory, but (about) $1/\pi$ of them are on analysis, more exactly, 7 out of 22 of them.

D. S.

43[10] —ROBERT SPIRA, *Cyclotomic Polynomial Generator and Tables*, Version A, Michigan State University, East Lansing, Mich., October 1969, 45 pp., 28 cm., deposited in UMT file.

This is an emended version of an undated report released several months earlier, which was found to contain several serious typographical errors in the arrangement of the tabular entries.

The numerical table herein consists of a parallel listing of values of the Euler totient, $\phi(n)$ and of the coefficients of the cyclotomic polynomial $Q_n(x)$ for $n = 1(1)250$. (This polynomial is defined as the irreducible monic polynomial of degree $\phi(n)$ that has as its zeros the primitive nth roots of unity.)

This main table is preceded by a detailed description (including flow charts and listings) of the four FORTRAN programs used in the calculations.

The introduction describes the mathematical procedure followed in the generation of the cyclotomic polynomials, which the author ascribes to Lehmer [1].

J. W. W.

1. D. H. LEHMER, *Guide to Tables in the Theory of Numbers*, Bulletin No. 105, National Research Council, Washington, D. C., 1941, p. 73.

44[12] —J. HILSENRATH, G. G. ZIEGLER, C. G. MESSINA, P. J. WALSH & R. J. HERBOLD, *Omnitab, A Computer Program for Statistical and Numerical Analysis*, National Bureau of Standards Handbook No. 101, 1966, reissued 1968 with corrections, 1x + 275 pp., 26 cm. Price $3.00.

Computing has come a long way from the early beliefs of von Neumann that a computer user will be a scientist who will know the range of every number entering in his calculation, and who will be so motivated that machine language programming will present no problem. In fact, even the use of floating-point arithmetic was considered to be "playing with fire". Today we find a veritable Tower of Babel of languages, collections of algorithms, subroutine libraries and operating systems, and the promise of a console in every home for doing Junior's homework and to facilitate menu preparation. It is therefore hard to realize that there exist large numbers of problem-oriented research people who want access to a large digital computer, but who do not want to learn programming, for example, they may just want "a least-squares fit". For these people large numbers of packages and "general-purpose" systems have been devised.

OMNITAB, developed by the Statistical Engineering Laboratory of the National Bureau of Standards, is a completely assembled interpretive program which provides facilities for doing a wide range of statistical and engineering type calculations. Originally written for the IBM 7090/7094, this volume is a manual for users with access to a computer with the OMNITAB system and indicates in detail the necessary

control cards, rules for carrying out computations and gives numerous examples. Instructions are given to the system in a form of English sentences simulating desk computing. Indicative of the popularity of this system, is the fact that subsequent to the issuance of this volume, the system has been rewritten in ASA Fortran and implemented in several other computers such as the Univac 1108, IBM 360-50 and 65 and CDC 6400 and 6600. While the volume under review indicates control cards for the 7094 only, various versions are in use throughout the country and a version has even been unveiled recently in a time-sharing environment.

As an old-fashioned "programming expert", this reviewer has a certain antipathy toward the concept of "instant programs"—I probably feel that one should have enough motivation to write the appropriate subroutine, if one wants to use anything beyond the arithmetic operators and elementary functions. However, the audience is there, and systems such as OMNITAB have served a real need. The user of this or similar systems will be well advised to read the section entitled diagnostic features. What is pointed out there, cannot be repeated too often:

"The concept of a general-purpose program rests in some measure on the assumption that the user, though not a programmer, is familiar with the behavior of the mathematical functions he is using or trying to compute . . . diagnostic features are incorporated . . . however diagnostic statements are no substitute for sound mathematical analysis, which is necessary to avoid the more serious pitfalls of numerical computations."

In this regard, it would have been useful to list the possible diagnostic statements generated by each command. In fact, the numerical methods used in the functions and routines should have been given. Not only would they be useful to users outside of OMNITAB, but they are a must when trying to gauge numerical accuracy. In the final analysis, it probably does not matter whether one learns to program or to use "packages" or both. Of paramount importance is the question of the accuracy of the results. The hope is that all users of computers in future generations, those in the physical as well as those in the social sciences, will learn the elements of numerical analysis.

MAX GOLDSTEIN

Courant Institute of Mathematical Sciences
New York University
New York, New York 10012

45[12].—F. R. A. HOPGOOD, *Compiling Techniques*, American Elsevier Publishing Co., Inc., New York, 1969, vi + 123 pp., 23 cm. Price $6.00.

This elegant monograph, suitable both for self-study and for text use in short courses on compiling, manages in its brief (126 pages) compass to discuss many of the most salient issues of compiler writing in an illuminating manner. An introductory section (28 pages) discusses compiler-related data structures and their computer treatment, and includes a thumbnail account of hashing. Backus notation is then introduced. Lexical analysis is discussed in a 10-page chapter which is, unfortunately, less transparent than other passages of the book. Many of the principal parsing methods are then nicely surveyed in a 20-page chapter. Code generation is next dis-