8128), 495 become periodic with period 2 (amicable pair), and 54 lead into one of the two Poulet cycles.

For a review of Paxson's related tables see *Math. Comp.*, v. 26, 1972, UMT **38**, pp. 807-809.

RICHARD K. GUY

45 [10].—P. A. MORRIS, *Characteristic Polynomials of Trees on up to* 14 *Nodes*, University of the West Indies, St. Augustine, Trinidad, West Indies, December 1973. Ms of 8 pp. + 57 computer sheets deposited in the UMT file.

Herein are listed the coefficients of the characteristic polynomials of the adjacency matrices of all trees with 13 or fewer nodes.

The list was generated in two ways to provide a check on the calculations, which were performed on a 1 CL 1902A and an IBM 1620, respectively. The first method employs a theorem of Collatz and Singowitz [1], which asserts that if $\phi(T) = \sum_{k=0}^{n} (-1)^k a_k \lambda^{n-k}$ is the characteristic polynomial of a tree T on n nodes, then $a_{2k+1} = 0$ and a_{2k} equals the number of ways of finding k mutually nonadjacent edges in T. The second method uses a known decomposition theorem [2], which states that if u is a node of valency 1 connected to a node v, $T - uv$ is the tree (together with the isolated node u) formed by deleting the edge uv, and $T - u - v$ is the forest formed by deleting nodes u and v and their incident edges, then $\phi(T) = \phi(T - uv) - \phi(T - u - v)$.

A further check of the accuracy of the list was made by comparison with the corresponding data in the table of Mowshowitz [3], which includes all trees on 10 or fewer nodes.

AUTHOR'S SUMMARY

1. L. COLLATZ & U. SINGOWITZ, "Spektren endlichen Graphen," *Abh. Math. Sem. Univ. Hamburg*, v. 21, 1957, pp. 63-77.

2. F. HARARY, C. KING, A. MOWSHOWITZ & R. C. READ, "Cospectral graphs and digraphs," *Bull. London Math. Soc.*, v. 3, 1971, pp. 321-328.

3. A. MOWSHOWITZ, "The characteristic polynomial of a graph," *J. Combinatorial Theory Ser. B*, v. 12, 1972, pp. 177-193.

46 [12].—LOUIS D. GREY, *A Course in* APL/360 *with Applications*, Addison-Wesley Publishing Co., Inc., Reading, Mass., 1973, xviii + 332 pp., 24 cm. Price $7.50 (paperbound).

If this paper-back book were used merely as a reference manual for APL programmers, it would serve a useful function since it is well organized, comprehensive and well documented. But the text is far more than a work of reference. It is an excellent vehicle for teaching this most elegant and succinct language, one which is considered by some to be a serious competitor with

the other two scientific languages—Fortran and PL/I—for prominence in the world of programming languages.

Divided into three major parts, the book leaves nothing of importance unsaid. The reader is introduced to the language at a steady pace, one designed for rapid learning, and provides as pleasant a learning experience as one could hope for. One is carried from the mechanics of sitting at an APL terminal through the APL character set, to all the many varieties of functions available to the APL user.

Despite the richness of the language, one is not left in the frustrating position of not being able to see the forest for the trees. Each topic is handled carefully and each description is more than adequate.

In addition to the numerous examples given, each chapter is followed by a series of exercises making the text admirably suitable for classroom use. Chapter 19 contains an interesting discussion of what lies ahead for APL. Though it is perhaps too early to say with any certainty to what degree the APL language will be accepted by the science and engineering community, this text can only help to pave the way for its acceptance.

HENRY MULLISH

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

47 [12].—NIKLAUS WIRTH, *Systematic Programming: An Introduction,* Prentice-Hall, Inc., Englewood Cliffs, N. J., 1973, 167 pp., 24 cm. Price $10.50.

During the past year or so, the notion of "structured programming" has become quite popular in the computer field. Although the term is still rather ill-defined, it seems to include concepts such as top-down programming, goto-less programming, program verification, and the chief programmer team method of project organization. Wirth's introductory textbook "Systematic Programming" is an attempt to teach some of these ideas to the student when he is first learning about computers. Program verification is presented as being part of programming itself; the *goto* statement is barely mentioned; and an excellent chapter is devoted to the process by which an outline of a program is fleshed out in top-down style until it becomes functional.

This is definitely not a book about "how to code". It requires considerable mathematical aptitude and knowledge on the part of the student (though Wirth disclaims this requirement). Program structure is emphasized at the same time that linguistic details about PASCAL, the language used for the sample programs, are kept to a minimum. A student who intends to continue in the computer field would need to study a programming language in far greater detail, though a mathematician trying to learn about the nature of programming would be quite satisfied with Wirth's presentation.