ensured for non-uniform distributions irrespective of the distribution of the key over the group of slots. Finally the applicability of H as the preliminary hashing function F in other external hashing methods, in particular extendible hashing, is being investigated.

Acknowledgements

The authors wish to thank Professor P. Stocker of the University of East Anglia for useful suggestions and comments on this study, which was partially funded by the Science and Engineering Research Council.

REFERENCES

1. S. M. Deen, D. Nikodem and A. Vashista, The design of a canonical database (PRECI). *Computer Journal* **24**, 200–209 (1981).
2. P. A. Larson, Dynamic hashing. *BIT* **18**, 184–201 (1978).
3. M. Scholl, New file organisations based on dynamic hashing. *ACM TODS* **6**, 194–211 (1981).
4. R. Fagin, J. Nievergelt, N. Pippenger and H. R. Strong, Extendible hashing—a fast access method for dynamic files. *ACM TODS* **4**, 315–344 (1979).
5. W. Litwin, Virtual hashing—a dynamically changing hashing. *Proceedings of the VLDB Conference*, Berlin, 517–573 (1978).
6. V. Y. Lum, P. S. T. Yuen and M. Dodd, Key to address transform techniques. *Communications of the ACM* **14**, 228–239 (1971).
7. S. M. Deen, An implementation of impure surrogates. Proceedings of 1982 Conference on VLDB, Mexico (1982).

Received May 1982

Book Reviews

Continued from p. 494

E. W. MARTIN AND W. C. PERKINS
FORTRAN for Business Students: A Programmed Instruction Approach
Wiley, Chichester, 1981. 811 pp. £12.15.

This book suffers from serious defects both in the material presented and in the manner of presentation.

The pedagogic approach is very stultifying. The book is *not* a programmed instruction text, despite the title. Instead, a sequence of small frames is presented, each containing some text and a question to be answered. However, if a wrong answer is given, the student is not directed to remedial material, but merely reconsiders why he gave that answer (and by now, of course, he has been shown the correct one). The level of the questions is of such utter triviality that the student is defended from boredom only by a mounting indignation against the insult to his intelligence.

The subject matter itself is not convincing. Leaving aside any consideration of the merits of ANSI FORTRAN IV in general, there is little case for its use as a first language for business students. The authors' discussion of files is, in consequence of their use of FORTRAN, inadequate and long deferred, central though the topic is to commercial data processing.

The technology described is out of date, both hardware and software (perhaps because such a massive tome took a decade to write!). It is assumed that the usual computing environment consists of access via punched cards to a batch-processing mainframe, which rather undermines the authors' claim that the book will serve as a text for any student with access to FORTRAN (further subverted by the occasional non-standardness of the FORTRAN they use). There is no attempt to encourage the student to think in terms of the constructs of structured programming, and that most valuable of structural tools, the subprogram, is not introduced until p. 535, where it is allocated only 35 pages.

In summary, this monumental volume is a poor realization of a misguided project.

C. D. F. MILLER
Leeds

R. E. BERRY
Programming Language Translation
Ellis Horwood, Chichester, 1982. 175 pp. £15.00, £6.50 paper.

This is a pleasant little book, which I enjoyed reading. Its title might mislead people into thinking that it is about translating from one high-level language to another, but in fact it is about translating from high-level to low-level, i.e. compiling.

The first six chapters (lexical analysis, syntax definition and syntax analysis, symbol tables—structure and access, the run time environment, semantic processing, run time support) are concerned mainly with the high-level side of the fence; the next three (assemblers, macros, loaders) more with the low-level side. The final two (Pascal S compiler, Pascal S interpreter) are the real meat of the book, describing a compiler in detail with a complete listing of it, written in Pascal (but not in the Pascal S subset which it compiles).

Considerable prior knowledge of both computing in general, and Pascal in particular, are taken for granted, and detailed knowledge of the Pascal S compiler is also assumed before you reach it, particularly in the exercises at the end of each of the first eight chapters. For this reason it is rather a difficult book to read, as the King of Hearts' algorithm ('Begin at the beginning, and go on till you come to the end: then stop') certainly will not do, but no other order is suggested. Probably the only answer is to read it several times.

The Pascal S syntax is given in diagrammatic form, but with too many errors in the diagrams. Other misprints are not too bad: ('relativizer' is a nasty enough word without misprinting it 'relavitizer' though). The printing is unfortunate in that the typefaces of both the main text and the Pascal listings have virtually identical renderings of letter l and figure 1. In some places, I was actually misled by this and it is nasty even where not misleading.

Although there are references throughout the text to a bibliography, I came to the conclusion during my reading that the bibliography itself has been accidentally omitted. I finally came across it, nestling at the end of Chapter 9, but there is no indication anywhere else of where or how to find it.

Why is it that Pacal books so often give reserved words underlining, as here, or boldface, as if it were Algol? There are those, including myself, who believe that a language ought to distinguish such words from identifiers, and those who believe it to be disadvantageous. Whoever is right about this, the fact is that Pascal does not do it, and it is misleading (or unfair to Pascal) to pretend that it has this advantage (or disadvantage) when it does not.

I. D. HILL
Harrow

MICHAEL SHAVE
Data Structures
McGraw-Hill, New York, 1981.

This book serves as a useful self-teaching text introducing a range of concepts. The range brings in on the one hand techniques which may be unfamiliar to a computer scientist and conversely some explanations are approached from the point of view of a computer scientist and may therefore present a somewhat different but interesting aspect of the work to those who are working in a professional computer applications role.

The author sets out the theory underlying a number of algorithms to cover functions. From these functions the choice would have to be made in order to set up a system to handle data, data structure or conventional files. In principle the data may be numerical or text or mixed.

The author covers storage structures including sequential, linked, binary trees, also allocation of space to linked structures, dynamic block allocation, methods of garbage collection and operations on tree structures.

A companion text which leads on to an assessment of the deployment of such algorithms with the various strands of database technology or alternative application systems would be both useful and interesting.

M. M. BARRITT
Edinburgh

Continued on p. 422