

## Book Review

*Language H (1962) Manual*, 1964; 111 pages. (London: The National Cash Register Co. Ltd., 42s.)

The attempt to float Basic English as a world language in the late forties had no chance of success, not because it was difficult for the non-English speaker to learn, but because it was virtually impossible for the English speaker to remember which of his ordinary words and turns of phrase constituted the sub-set, Basic English, and which did not. Within the thicket of his vocabulary, the boundary posts of Basic English were too undifferentiated to discern.

There is a lesson here for inventors of commercial computer languages. They assume that the language is easier to use if their executive functions are words in normal English usage. They encourage the use of English operands, and they supply linking words which serve no other purpose than to provide a spurious semblance of English. They have two aims. They want the layman to be able to read a computer program. But why should he? It is none of his business. They claim that a system flow-diagram can be written in their language, and lo!—programming is already accomplished. One such language has developed such an extensive vocabulary that the programmer is left groping for his boundaries, the programmer who would perhaps be happier if functions were given in Latin or Swahili.

Of course we do need computer languages which are not creatures of a particular machine or machine group. We need more of the simple basic ones, which will, it is to be hoped, drive out the prolix. This is why Language H should be welcome. It does, indeed, use English words rather than Erse. It employs fourteen words which are redundant or behave as punctuation "but help to amplify the meaning of the statement". Presumably these are included to appeal to

this importunate layman. The considerations used in developing the language, however, give a clue to its character. To quote from the Preface:

"the smallest possible number of effective phrases should be provided"

"the strain on the memory of the user should be kept as low as possible"

"The wealth of possible expressions, such as compound conditional phrases, has been deliberately reduced in order to make learning and accurate use easier"

Nobody is likely to quarrel with those aims.

Language H is expected to behave as a high-level language, but there is none of the tiresome indenting which goes with compound conditions. The program example given, if it were not for the linking words, smacks of a symbolic order code in three-address mode, using literals and interspersed with macro-instructions. Since the language is not machine-orientated, the resemblance is mainly on the surface, but the simple structure should enable relatively simple and efficient compilers to be developed for machines outside the National-Elliott range.

On the question of macro-instructions, input and output are adequately explained as GET and FILE, but I could find no reference in the manual to hierarchical sorting procedures, although SORT is used as a reserved word. Indeed, less than half of the reserved words appear in the index, and this is inconvenient.

The main text of the manual runs to thirty pages, the rest being taken up by thirteen appendices, some referring solely to the machines which already have compilers. A handy little pocket book is included as a programmer's *aide-mémoire*.

F. I. MUSK