

## The Core Memory Project

### NCR Language H

Business processing language from NCR, English style.

Cormack National Cash Register 1962.

Designed to permit COBOL style operations in a simpler language, and suitable for running on a smaller machine.

Language H has been designed to avoid the complications of data and environment divisions and, as a result, a source program consists of a brief heading and a set of statements which form the procedure division. These statements may be grouped into segments, called chapters, for processing convenience.

The program statements are all executive in character, i.e. of the form "Do something" and are independent of the type of computer on which the object program will ultimately run. The first word in a source program statement is always an imperative verb and subsequent words are either operand names or "noise" words, the latter being inserted to increase the intelligibility of the statement.

For example, a valid Language H statement is:

```
ADD GROSS PAY TO RUNNING TOTAL.
```

Here, ADD is the imperative verb, GROSS PAY and RUNNING TOTAL are names of operands and TO is a redundant "noise" word. Language H employs a modest number of "reserved" words, such as ADD in the example, there being 84 reserved words in all including 15 "noise" words. The number of words used as verbs is 23 and there are 61 possible variations of program statement format. There are, in addition, 14 permissible qualifying, or conditional, phrases which may be used to alter the sequence in which program statements are performed.

For example, the statement:

```
GO TO LEAVER OUTPUT IF LEAVER FLAG IS ON
```

will be obeyed only if a particular flag called LEAVER FLAG is in the "ON" state. If the statement is obeyed, a jump is made to the appropriate program statement labelled LEAVER OUTPUT.

Operands may be classed as either numeric, alpha-numeric or special, and there are 15 different types of operand permissible in the Language.

Punctuation has no syntactic significance and may be used or not at the convenience of the programmer. In particular, the names of operands do not need hyphens and program statements do not have to terminate with full-stops.

<http://www.thcorememory.com>

## The Core Memory Project

The Compiler has been implemented for three different types of computer and work is in hand to implement the Language on at least two more types.

As an example of the compiling performance of one version of the current Language H Compilers, the following figures are of interest:

- (i) Number of statements in Source program 1,900;
- (ii) M/C code instructions (single address) produced 21,000;
- (iii) Compiling time 60 minutes;
- (iv) Expansion ratio (ii)/(i) 11;
- (v) Source Compilation rate (i)/(iii) (Statements per minute) 31.7;
- (vi) Object Compilation rate (ii)/(iii) (Instructions per minute) 350.0.

These results are obtained from a particular program and are not, therefore, universally applicable.

Language H has been used as a training language on programming courses for beginners with considerable success and a detailed programmers' manual is in the course of publication.

Several production jobs written in Language H have been operating successfully in NCR's Financial Computing Centre for more than a year and a great deal of practical operating experience in the use of the Language has been gained.

*Beaven, A. H. "PROGRESS REPORT ON LANGUAGE H" Computer Bulletin March 1964*

### Brief description

Language H is a simple, single level, procedural language allowing one level of subscripting, in which many data-processing problems may be expressed. No separate description of the data being handled is required of the user sufficient information is obtained by implication, from the way phrases are written, to provide a range of checks on validity and to enable appropriate machine-code to be produced.

The conventions required to meet the needs of filing and input and output specifications are simple and straightforward. Some of the requirements borne in mind in developing this approach have been the following:

- (1) The smallest possible number of effective phrases should be provided.
- (2) The load on the memory of the user should be kept as small as possible.
- (3) Any program should be unambiguous.
- (4) The structure of the language should be independent of any particular computer.
- (5) The unit of information handled should be of the size of a number, a name, or a short comment; only exceptionally is it, on the one hand, a single character or, on the other, a record.

## The Core Memory Project

In spite of the simplicity of the language it is not as naive as perhaps it may appear from the description. It has most of the facilities which people have been led to expect from autocoding languages, such as:

- (a) nested repeat loops;
- (b) conditional branch commands;
- (c) provision for segmentation;
- (d) a perform command to permit sections of program to be obeyed out of sequence;
- (e) one level of subscripting for list work.

Some of the proposals for the next version include provisions for sorting, random access, the increase of levels for subscripting, and the extension of the file commands.

The example in Fig. 1 is an extract from the Language H Payroll program mentioned later in the paper. It is not a complete section and is, by itself, meaningless, but it is intended to illustrate the way in which the language is used to handle data-processing problems, and to give an indication of the effect of some of the commands.

It will be noticed that punctuation is used merely to increase readability and does not have the syntactic meaning which some other languages assign to it. In other words, it is not necessary to remember that, say, a full stop terminates an imperative sentence and that a comma terminates a conditional expression or that an imperative statement is terminated by a semi-colon. The syntax of Language H is defined mainly by the way in which the commands are written.

*Cormack A.S. "Early operating experience with Language H"*