


**Addendum**

Two other languages, Transcode and PACT I, deserve to be part of the story as well, so they appear in Table 1 above although they were unfortunately missed by the authors when we first compiled this history.

Transcode was developed for the FERUT computer, a clone of the Ferranti Mark I that was installed at the University of Toronto in 1952. The authors of this language, J. N. P. Hume and B. H. Worsley, devised a way to cope with FERUT’s awkward two-level storage that was more efficient than the AUTOCODE approach being taken independently by Brooker in Manchester, because they expected Transcode programmers to be aware of FERUT’s overall characteristics. To perform the TPK algorithm they might have punched the following codes onto a paper tape:

```
000 INST 019   Inputs the following 19 Instructions.
001 READ 001.0 000.0 X00.0 Copies DRUM location 1 into X-page.
002 ADDN C01.0 C01.0 Z01.0 Sets initial value of i = 10 − j to 5 + 5.
003 BSET 000.5 000.0 000.0 Sets initial value of B5 to 0.
004 LOOP 011.0 000.0 000.0 Prepares to loop 11 times, using B6.
005 KOMP X11.5 X02.0 Z02.0 Places \( |a_i| = 0 \) in Z02.
006 \( \sqrt{j} \)RT Z02.0 000.0 Z02.0 Forms \( |a_i|^{1/2} \).
007 MULT X11.5 X11.5 Z03.0 Places \( a_i \cdot a_i \) in Z03.
008 MULT X11.5 Z03.0 Z03.0 Forms \( a_i \cdot a_i \cdot a_i \).
009 MULT C01.0 Z03.0 Z03.0 Forms \( 5a_i \cdot a_i \cdot a_i \).
010 ADDN Z02.0 Z03.0 Z02.0 Forms \( f(a_i) \).
```
The variables in Transcode, other than those in B registers, were floating-point numbers X01, X02, ..., Y01, Y02, ..., Z01, Z02, ..., consisting of three 20-bit words each, namely a 40-bit signed fraction and a signed 20-bit exponent. But Transcode programmers didn’t have to deal with binary notation; for example, the five constants C01 = 5, C02 = 0, C03 = 400, C04 = 999, C05 = 1 in the program above are specified on line 020 in decimal form as (mantissa) (sign) (exponent) (exponent sign). A constant like -0.0073 would be ‘73-3-’; and the same conventions applied to input data in a NUMB specification such as line 021. Notice that they used 1-100 for the constant zero(!). The PRINT command on line 014 would output Z01 and Z02 to a line on FERUT’s typewriter, showing both numbers in floating-point notation with ten significant digits.

Variables were stored backwards in memory, so that the address of the first word of X02 was 3 locations less than the address of X01. The main loop of this program, governed by instructions 003, 004, 016, and 017, is performed for eleven index register settings (B5, B6) = (0, 30), (3, 27), ..., (30, 0); thus X11.5 is X11 the first time, then X10, ..., then X01. Meanwhile variable i = Z01 steps through the values 10, 9, ..., 0, because of instructions 002 and 015.

Programmers could say KOPY at the end, following ENTR; then the compiled instructions would be punched on paper tape, for subsequent use as a subroutine in other programs. Since paper tape code was equivalent to teletype code, programs could also be transmitted “online” to Toronto from remote sites in Canada. [For further details, see J. N. P. Hume and Beatrice H. Worsley, “Transcode: A system of automatic coding for FERUT,” *Journal of the Association for Computing Machinery* 2 (1955), 243–252; J. N. Patterson Hume, “Development of systems software for the Ferut computer at the University of Toronto, 1952 to 1955,” *IEEE Annals of the History of Computing* 16, 2 (Summer 1994), 13–19.]