In the beginning, at the birth of computing, there were no programming languages. Programs looked something like this:

```
00110001 00000000 00000000 00110001 00000001
00000001 00110011 00000001 00000010 01010001
00001011 00000010 00100010 00000010 00001000
01000011 00000001 00000000 01000001 00000001
00000001 00010000 00000010 00000000 01100010
00000000 00000000
```

That is a program to add the numbers from 1 to 10 together and print out the result: `1 + 2 + ... + 10 = 55`. It could run on a simple, hypothetical machine. To program early computers, it was necessary to set large arrays of switches in the right position or punch holes in strips of cardboard and feed them to the computer. You can probably imagine how tedious and error-prone this procedure was. Even writing simple programs required much cleverness and discipline. Complex ones were nearly inconceivable.

Of course, manually entering these arcane patterns of bits (the ones and zeros) did give the programmer a profound sense of being a mighty wizard. And that has to be worth something in terms of job satisfaction.

Each line of the previous program contains a single instruction. It could be written in English like this:

```
1. Store the number 0 in memory location 0. 2.
Store the number 1 in memory location 1. 3. Store
the value of memory location 1 in memory location
2. 4. Subtract the number 11 from the value in
memory location 2. 5. If the value in memory
location 2 is the number 0,    continue with
instruction 9. 6. Add the value of memory location
1 to memory location 0. 7. Add the number 1 to the
value of memory location 1. 8. Continue with
instruction 3. 9. Output the value of memory
location 0.
```