Problem E: Interpreter

A certain computer has 10 registers and 1000 words of RAM. Each register or RAM location holds a 3-digit integer between 0 and 999. Instructions are encoded as 3-digit integers and stored in RAM. The encodings are as follows:

- 100 means *halt*
- 2dn means *set register d to n (between 0 and 9)*
- 3dn means *add n to register d*
- 4dn means *multiply register d by n*
- 5ds means *set register d to the value of register s*
- 6ds means *add the value of register s to register d*
- 7ds means *multiply register d by the value of register s*
- 8da means *set register d to the value in RAM whose address is in register a*
- 9sa means *set the value in RAM whose address is in register a to the value of register s*
- 0ds means *goto the location in register d unless register s contains 0*

All registers initially contain 000. The initial content of the RAM is read from standard input. The first instruction to be executed is at RAM address 0. All results are reduced modulo 1000.

The input to your program consists of up to 1000 3-digit unsigned integers, representing the contents of consecutive RAM locations starting at 0. Unspecified RAM locations are initialized to 000.

The output from your program is a single integer: the number of instructions executed up to and including the *halt* instruction. You may assume that the program does halt.

Sample Input

```
299
492
495
399
492
495
399
283
279
689
078
100
000
000
000
```

Output for Sample Input

16