One Instruction Computer Simulator

A computer with only one instruction! The instruction is:

SUBLEQ A B C

This means: subtract the value in M(A) from M(B) and store it in M(B); if the result is non-positive jump to the instruction in position C. M(i) represents the value stored in memory position i. The computer has a memory of 9999 integer positions, numbered from 0 to 9998. C > 9996, indicates the end of the program. Also, if A is negative, then the value of A is directly subtracted from M(B).

Since there is only one instruction, it is unnecessary to represent its opcode explicitely in memory. Therefore, an instruction is stored in main memory using three consecutive memory positions, which correspond to the three instruction parameters. The memory is organized as follows:

Position Content
0-8 input/output variables (M0 to M8)
9-9998 program memory (instructions+data)

The following pseudo-code shows the one instruction computer simulator:

```
simulate(integer M[0..9998])
  integer pc,A,B,C
  pc = 9
  while (pc<9997)
    A = M[pc]; B = M[pc+1]; C = M[pc+2]
    if(A>=0)
      M[B] = M[B] - M[A]
    else
      M[B] = M[B] - A
    if (M[B]>0)
      pc = pc + 3
    else
      pc = C
    end_if
  end_while
end_simulate
```

Each iteration of the above while instruction is called a simulation cycle. You are to translate postfix instructions into this machine language. There are at most 100 arithmetic terms and 99 operators. Numerical constants are non-negative and less than or equal to 10000.

Input

The input has several test cases, one test case per line. Each test case corresponds to an arithmetic expression in postfix notation. An expression may contain constants (integer values), input variables (M0 to M8) and arithmetic operators (+, -, *, /).

Output

For each test case, a program must be printed using the following format: First line indicates m, the number of instructions of the program; and the following m lines contain the program, one

instruction per line, where each instruction is represented by 3 integer values separated by one blank space. Your outputted program must finish within 10^7 simulation cycles for each test case.

Example

Input:
100
M1 M2 -

Output:
3
0 0 12
-100 0 0
19 19 10000

4
0 0 12
1 2 15
2 0 18
21 21 10000