ACM-ICPC Live Archive

2944 Computer

Little Tom is learning how to program. He has just written some programs but is afraid to run them, because he does not know if they will ever stop. Please write a program to help him.

This is not as easy a task as it may seem, because the computer Tom has is not deterministic.

Given a program written by Tom, your program should tell him whether his program can possibly stop and if so, what is the shortest possible time before it stops.

The computer consists of 32 1-bit registers and the program consists of n instructions. The registers are numbered from 0 to 31 and the instructions are numbered from 0 to $n-1$.

Below, "`MEM[`a`]`" stands for the contents of the a-th register, $0 \le a, b < 32$, $0 \le x < n$, $0 \le c \le 1$.

The instruction set is the following:

Instruction	**Semantics**
`AND` a b	`MEM[`a`]` `:=` `MEM[`a`]` and `MEM[`b`]`
`OR` a b	`MEM[`a`]` `:=` `MEM[`a`]` or `MEM[`b`]`
`XOR` a b	`MEM[`a`]` `:=` `MEM[`a`]` xor `MEM[`b`]`
`NOT` a	`MEM[`a`]` `:=` not `MEM[`a`]`
`MOV` a b	`MEM[`a`]` `:=` `MEM[`b`]`
`SET` a c	`MEM[`a`]` `:=` c
`RANDOM` a	`MEM[`a`]` `:=` random value (0 or 1)
`JMP` x	jump to instruction number x
`JZ` x a	jump to instruction number x if `MEM[`a`]` `=` 0
`STOP`	stop the program

The last instruction of a program is always '`STOP`' (although there can be more than one `STOP` instruction). Every program starts with the instruction number 0. Before the start, the contents of the registers are random. Each instruction (including `STOP`) takes 1 processor cycle to execute.

Write a program that:

- reads the program from the standard input,
- computes the shortest possible running time of the program,
- writes the result to the standard output.

Input

Input consists of several test cases, each of them following the description below. A blank line separates two consecutive cases.

The first line of the input contains integer n ($1 \le n \le 16$) being the number of instructions of the program. Each of the next n lines contains one instruction of the program in the format given above.

Output

For each test case, the first and only line of the output should contain the shortest possible running time of the program, measured in processor cycles. If the program cannot stop, output should contain the word '`HANGS`'.

The outputs of two consecutive cases will be separated by a blank line.

Sample Input

```
5
SET 0 1
JZ 4 0
RANDOM 0
JMP 1
STOP

5
MOV 3 5
NOT 3
AND 3 5
JZ 0 3
STOP
```

Sample Output

```
6

HANGS
```