

2929 Hang or not to hang

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 task is not as easy as it may seem, because Tom's programs may behave nondeterministically.

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

Tom's 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, $\text{MEM}[a]$ stands for the contents of the a -th register, $0 \leq a < 32$, $0 \leq x < n$, $0 \leq c \leq 1$.

The instruction set is as follows:

| Instruction | Semantics |
|-------------|--|
| AND $a b$ | $\text{MEM}[a] := \text{MEM}[a] \text{ and } \text{MEM}[b]$ |
| OR $a b$ | $\text{MEM}[a] := \text{MEM}[a] \text{ or } \text{MEM}[b]$ |
| XOR $a b$ | $\text{MEM}[a] := \text{MEM}[a] \text{ xor } \text{MEM}[b]$ |
| NOT a | $\text{MEM}[a] := \text{not } \text{MEM}[a]$ |
| MOV $a b$ | $\text{MEM}[a] := \text{MEM}[b]$ |
| SET $a c$ | $\text{MEM}[a] := c$ |
| RANDOM a | $\text{MEM}[a] := \text{random value (0 or 1)}$ |
| JMP x | jump to the instruction with the number x |
| JZ $x a$ | jump to the instruction with the number x if $\text{MEM}[a] = 0$ |
| STOP | stop the program |

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

Task

Write a program that:

- reads the program,
- computes the shortest possible running time of the program,
- writes the result.

Input

Input consists of several test case, each separated by a blank line. The first line of the file indicates the number of test cases, and it's followed by a blank line.

The first line of each dataset contains an integer n ($1 \leq n \leq 16$) 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. You may assume that the only white characters in the program are single spaces between successive tokens of each instruction.

Output

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'.

Print a blank line between datasets.

Sample Input

2

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