Multithreading

You are a computer scientist with access to a supercomputer that can run a huge amount of processes at once.

The supercomputer is controlled using a command-line interface. The processes you can run are saved in storage and each process is uniquely identified by a positive integer ***ID***.

The process with ***ID x*** can be run using the command ***RUN x*** and can be terminated using ***END x***.

Since entering commands one-by-one is quite tedious, you can instead run a script that contains a list of commands, one per each line, and the supercomputer will execute them in order.

Even though the supercomputer has great computing power, it is not good at error handling, and will crash if an invalid command is executed. A command is invalid if:

- *The command runs a process which is already running*
- *The command ends a process which is not currently running*
- *The command ends a process while the processes started after it are still running*
  - *If **process X** is run before **process Y**, **process Y** must be ended **before** ending **process X** (or **process Y** will **crash**)*

The supercomputer will also crash if the script doesn't end all processes it started.

Since you don't want the supercomputer to crash, you need to make sure a script wouldn't cause any crashes if it is run on the supercomputer. Write a program to find out if there are any possible crashes in a given script.

Input Format

First line contains a single positive integer ***N***, number of lines in the script.
N lines follow, each with either "***RUN x***" or "***END x***", where ***x*** is a positive integer denoting the process ID.

Constraints

- $1 \leq N \leq 10^7$
- $1 \leq x \leq 10^8$

Limits

- **Time Limit**: 1s
- **Memory Limit**: 256MB

Output Format

A single integer ***k***, where

- ***k = 0*** if the script doesn't crash, or
- ***k = n + 1*** if all commands are valid but the script doesn't end all processes it started, or
- ***k = j*** $(1 \leq j \leq n)$ where ***j*** is the first invalid command in the script.

Sample Input 0

```
6
RUN 1
RUN 3
END 3
END 1
RUN 2
END 2
```

Sample Output 0

```
0
```