

## C. Okabe and Boxes

time limit per test: 3 seconds  
memory limit per test: 256 megabytes  
input: standard input  
output: standard output

Okabe and Super Hacker Daru are stacking and removing boxes. There are  $n$  boxes numbered from 1 to  $n$ . Initially there are no boxes on the stack.

Okabe, being a control freak, gives Daru  $2n$  commands:  $n$  of which are to add a box to the top of the stack, and  $n$  of which are to remove a box from the top of the stack and throw it in the trash. Okabe wants Daru to throw away the boxes in the order from 1 to  $n$ . Of course, this means that it might be impossible for Daru to perform some of Okabe's *remove* commands, because the required box is not on the top of the stack.

That's why Daru can decide to wait until Okabe looks away and then reorder the boxes in the stack in any way he wants. He can do it at any point of time between Okabe's commands, but he can't add or remove boxes while he does it.

Tell Daru the minimum number of times he needs to reorder the boxes so that he can successfully complete all of Okabe's commands. It is guaranteed that every box is added before it is required to be removed.

### Input

The first line of input contains the integer  $n$  ( $1 \leq n \leq 3 \cdot 10^5$ ) — the number of boxes.

Each of the next  $2n$  lines of input starts with a string "add" or "remove". If the line starts with the "add", an integer  $x$  ( $1 \leq x \leq n$ ) follows, indicating that Daru should add the box with number  $x$  to the top of the stack.

It is guaranteed that exactly  $n$  lines contain "add" operations, all the boxes added are distinct, and  $n$  lines contain "remove" operations. It is also guaranteed that a box is always added before it is required to be removed.

### Output

Print the minimum number of times Daru needs to reorder the boxes to successfully complete all of Okabe's commands.

### Examples

input
3 add 1 remove add 2 add 3 remove remove
output
1

input
7 add 3 add 2 add 1 remove add 4 remove remove remove add 6 add 7

add 5 remove remove remove
output
2

**Note**  
In the first sample, Daru should reorder the boxes after adding box 3 to the stack.  
In the second sample, Daru should reorder the boxes after adding box 4 and box 7 to the stack.