

# ROUND 2D - Stefan and Flash

Stefan và Flash đang chơi trò chơi giải đố. Có  $n$  ô được đánh số từ 1 đến  $n$ . Ban đầu không có hộp trên ngăn xếp.

Stefan là một kẻ quái chiêu, một kẻ đủ mạnh để Flash phải tuân theo trò chơi của hắn. Hắn đưa ra  $2n$  câu lệnh,  $n$  lệnh là để thêm một hộp vào đỉnh stack, và còn lại trong số đó là để loại bỏ một hộp từ phía trên ngăn xếp và ném nó vào thùng rác. Stefan muốn Flash ném các hộp theo thứ tự từ 1 đến  $n$ . Tất nhiên, Flash không thể làm được bởi vì hộp yêu cầu không nằm trên đỉnh stack.

Đó là lý do tại sao Flash có thể quyết định đợi cho đến khi Stefan quay đi và sau đó sắp xếp lại các hộp trong ngăn xếp theo bất kỳ cách nào anh ta muốn. Anh ta có thể làm điều đó bất cứ lúc nào giữa các lệnh của Okabe, nhưng anh ta không thể thêm hoặc bỏ các ô.

Bạn hãy giúp Flash tính số lần tối thiểu anh ta cần sắp xếp lại các ô để anh ta có thể hoàn thành thành công tất cả các lệnh của Stefan.

## Input

Dòng đầu tiên của đầu vào chứa số nguyên  $n$  ( $1 \leq n \leq 3 \cdot 10^5$ ) - số hộp.

$2n$  dòng tiếp theo bắt đầu với một chuỗi "add" hoặc "remove". Nếu dòng bắt đầu bằng "add", một số nguyên  $x$  ( $1 \leq x \leq n$ ) theo sau, chỉ ra rằng Flash nên thêm hộp với số  $x$  lên trên cùng của ngăn xếp.

Bộ test đảm bảo rằng một hộp luôn luôn được thêm vào trước khi nó được yêu cầu phải được gỡ bỏ và có chính xác  $n$  dòng chứa các thao tác "add", tất cả các hộp được thêm vào là khác nhau được sắp tăng dần, và  $n$  dòng có chứa "remove".

## Output

In số lần tối thiểu Flash cần sắp xếp lại các ô để hoàn thành tất cả các lệnh của Stefan

## Example

**Input:**

```
3
add 1
remove
add 2
add 3
remove
remove
```

**Output:**

```
1
```