

## D. Points

time limit per test: 2 seconds  
memory limit per test: 256 megabytes  
input: standard input  
output: standard output

Pete and Bob invented a new interesting game. Bob takes a sheet of paper and locates a Cartesian coordinate system on it as follows: point  $(0, 0)$  is located in the bottom-left corner,  $Ox$  axis is directed right,  $Oy$  axis is directed up. Pete gives Bob requests of three types:

- `add x y` — on the sheet of paper Bob marks a point with coordinates  $(x, y)$ . For each request of this type it's guaranteed that point  $(x, y)$  is not yet marked on Bob's sheet at the time of the request.
- `remove x y` — on the sheet of paper Bob erases the previously marked point with coordinates  $(x, y)$ . For each request of this type it's guaranteed that point  $(x, y)$  is already marked on Bob's sheet at the time of the request.
- `find x y` — on the sheet of paper Bob finds all the marked points, lying strictly above and strictly to the right of point  $(x, y)$ . Among these points Bob chooses the leftmost one, if it is not unique, he chooses the bottommost one, and gives its coordinates to Pete.

Bob managed to answer the requests, when they were 10, 100 or 1000, but when their amount grew up to  $2 \cdot 10^5$ , Bob failed to cope. Now he needs a program that will answer all Pete's requests. Help Bob, please!

### Input

The first input line contains number  $n$  ( $1 \leq n \leq 2 \cdot 10^5$ ) — amount of requests. Then there follow  $n$  lines — descriptions of the requests. `add x y` describes the request to add a point, `remove x y` — the request to erase a point, `find x y` — the request to find the bottom-left point. All the coordinates in the input file are non-negative and don't exceed  $10^9$ .

### Output

For each request of type `find x y` output in a separate line the answer to it — coordinates of the bottommost among the leftmost marked points, lying strictly above and to the right of point  $(x, y)$ . If there are no points strictly above and to the right of point  $(x, y)$ , output  $-1$ .

### Examples

input
7 add 1 1 add 3 4 find 0 0 remove 1 1 find 0 0 add 1 1 find 0 0
output
1 1 3 4 1 1

input
13 add 5 5 add 5 6 add 5 7 add 6 5 add 6 6 add 6 7

```
add 7 5
add 7 6
add 7 7
find 6 6
remove 7 7
find 6 6
find 4 4
```

**output**

```
7 7
-1
5 5
```