Mirko is playing with stacks. In the beginning of the game, he has an empty stack denoted with number 0. In the $\mathbf{i^{th}}$ step of the game he will choose an existing stack denoted with $\mathbf{v}$, copy it and do one of the following actions:

a. place number $\mathbf{i}$ on top of the new stack

b. remove the number from the top of the new stack

c. choose another stack denoted with $\mathbf{w}$ and count how many different numbers exist that are in the new stack and in the stack denoted with $\mathbf{w}$

The newly created stack is denoted with $\mathbf{i}$.

Mirko doesn't like to work with stacks so he wants you to write a programme that will do it for him. For each operation of type *b* output the number removed from stack and for each operation of type *c* count the required numbers and output how many of them there are.

INPUT

The first line of input contains the integer $\mathbf{N}$ ($1 \leqslant \mathbf{N} \leqslant 300\,000$), the number of steps in Mirko's game.

The steps of the game are chronologically denoted with the first $\mathbf{N}$ integers.

The $\mathbf{i^{th}}$ of the following $\mathbf{N}$ lines contains the description of the $\mathbf{i^{th}}$ step of the game in one of the following three forms:

- "a v" for operation of type *a*.
- "b v" for operation of type *b*.
- "c v w" for operation of type *c*.

The first character in the line denotes the type of operation and the following one or two denote the accompanying stack labels that will always be integers from the interval $[0, \mathbf{i} - 1]$.

For each operation of type *b*, the stack we're removing the element from will not be empty.

OUTPUT

For each operation type *b* or *c* output the required number, each in their own line, in the order the operations were given in the input.

SAMPLE TESTS

input

```
5
a 0
a 1
b 2
c 2 3
b 4
```

output

```
2
1
2
```

input

```
11
a 0
a 1
a 2
a 3
a 2
c 4 5
a 5
a 6
c 8 7
b 8
b 8
```

output

```
2
2
8
8
```

Clarification of the first example: In the beginning, we have the stack $S_0 = \{\}$. In the first step, we copy S_0 and place number 1 on top, so $S_1 = \{1\}$. In the second step, we copy S_1 and place 2 on top of it, $S_2 = \{1, 2\}$. In the third step we copy S_2 and remove number 2 from it, $S_3 = \{1\}$. In the fourth step we copy S_2 and denote the copy with S_4, then count the numbers appearing in the newly created stack S_4 and stack S_3, the only such number is number 1 so the solution is 1. In the fifth step we copy S_4 and remove number 2 from it, $S_5 = \{1\}$.