

## 6392 Queue Sequence

There's a queue obeying the first in first out rule. Each time you can either push a number into the queue ( $+i$ ), or pop a number out from the queue ( $-i$ ). After a series of operation, you get a sequence (e.g.  $+1 -1 +2 +4 -2 -4$ ). We call this sequence a queue sequence.

Now you are given a queue sequence and asked to perform several operations:

### 1. insert $p$

First you should find the smallest positive number (e.g.  $i$ ) that does not appear in the current queue sequence, then you are asked to insert the  $+i$  at position  $p$  (position starts from 0). For  $-i$ , insert it into the right most position that result in a valid queue sequence (i.e. when encountered with element  $-x$ , the front of the queue should be exactly  $x$ ).

For example,  $(+1 -1 +3 +4 -3 -4)$  would become  $(+1 +2 -1 +3 +4 -2 -3 -4)$  after operation 'insert 1'.

### 2. remove $i$

Remove  $+i$  and  $-i$  from the sequence.

For example,  $(+1 +2 -1 +3 +4 -2 -3 -4)$  would become  $(+1 +2 -1 +4 -2 -4)$  after operation 'remove 3'.

### 3. query $i$

Output the sum of elements between  $+i$  and  $-i$ . For example, the result of query 1, query 2, query 4 in sequence  $(+1 +2 -1 +4 -2 -4)$  is 2, 3 (obtained by  $-1 + 4$ ),  $-2$  correspond.

## Input

There are less than 25 test cases. Each case begins with a number indicating the number of operations  $n$  ( $1 \leq n \leq 100000$ ). The following  $n$  lines will be 'insert  $p$ ', 'remove  $i$ ' or 'query  $i$ ' ( $0 \leq p \leq \text{length}$  (current sequence),  $1 \leq i$ ,  $i$  is granted to be in the sequence).

In each case, the sequence is empty initially.

The input is terminated by EOF.

## Output

Before each case, print a line 'Case # $d$ ' indicating the id of the test case.

After each operation, output the sum of elements between  $+i$  and  $-i$ .

## Sample Input

```
10
insert 0
insert 1
query 1
query 2
insert 2
query 2
remove 1
```

```
remove 2
insert 2
query 3
6
insert 0
insert 0
remove 2
query 1
insert 1
query 2
```

### Sample Output

```
Case #1:
2
-1
2
0
Case #2:
0
-1
```