

1480 Jewel

Jimmy wants to make a special necklace for his girlfriend. He bought many beads with various sizes, and no two beads are with the same size. Jimmy can't remember all the details about the beads, for the necklace is so long. So he turns to you for help.

Initially, there is no bead at all, that is, there is an empty chain. Jimmy always sticks the new bead to the right of the chain, to make the chain longer and longer. We number the leftmost bead as Position 1, and the bead to its right as Position 2, and so on. Jimmy usually asks questions about the beads' positions, size ranks and actual sizes. Specifically speaking, there are 4 kinds of operations you should process:

Insert x : Put a bead with size x to the right of the chain ($0 < x < 2^{31}$, and x is different from all the sizes of beads currently in the chain)

Query_1 $s t k$: Query the k -th smallest bead between position s and t , inclusive. You can assume $1 \leq s \leq t \leq L$, (L is the length of the current chain), and $1 \leq k \leq \min(100, t - s + 1)$

Query_2 x : Query the rank of the bead with size x , if we sort all the current beads by ascent order of sizes. The result should be between 1 and L (L is the length of the current chain)

Query_3 k : Query the size of the k -th smallest bead currently ($1 \leq k \leq L$, L is the length of the current chain)

Input

There are several test cases in the input. The first line for each test case is an integer N , indicating the number of operations. Then N lines follow, each line contains one operation, as described above.

You can assume the amount of 'Insert' operation is no more than 100000, and the amounts of 'Query_1', 'Query_2' and 'Query_3' are all less than 35000.

Output

Output 4 lines for each test case. The first line is 'Case T :', where T is the id of the case. The next 3 lines indicate the sum of results for 'Query_1', 'Query_2' and 'Query_3', respectively.

Explanation for the sample: The answers for the 5 queries are 6, 4, 3, 4, 1, respectively.

Sample Input

```
10
Insert 1
Insert 4
Insert 2
Insert 5
Insert 6
Query_1 1 5 5
Query_1 2 3 2
Query_2 4
Query_3 3
Query_3 1
```

Sample Output

Case 1:

10

3

5