

Binary Search Tree

Time Limit: 1 Second
Memory Limit: 2048 MB

Implement a multiset with binary search tree that supports the following operations on the set:

- **Insert** x : insert a new element x into the multiset.
- **Delete** x : delete x from the multiset. If there is more than one such value in the multiset, delete one of them. It is guaranteed that the given value is in the multiset.
- **Rank** x : count the number of elements in the multiset that is strictly less than x .
- **Kth** k : find the k -th smallest element in the multiset. If k is greater than the number of elements in the set, output "Wrong!".

It is guaranteed that the test cases are generated randomly.

Input

The first line of input contains a single integer n ($1 \leq n \leq 10^5$) - the number of operations.

The next n lines contain the description of the operations. Each line starts with one of **Insert**, **Delete**, **Rank**, and **Kth**. If the operation is **Insert**, **Delete**, or **Rank**, it will be followed by an integer x ($-10^9 \leq x \leq 10^9$). If the operation is **Kth**, it will be followed by an integer k ($1 \leq k \leq n$).

Output

For each operation of type **Rank** or **Kth**, output an integer denoting the answer to the query, or "Wrong!" (case sensitive) if no such value exists.

Sample Inputs

```
6
Insert 1
Insert 2
Insert 3
Kth 2
Rank 3
Kth 5
```

Sample Outputs

```
2
2
Wrong!
```
