

In the traditional RMQ (Range Minimum Query) problem, we have a static array  $A$ . Then for each query  $(L, R)$  ( $L \leq R$ ), we report the minimum value among  $A[L], A[L + 1], \dots, A[R]$ . Note that the indices start from 1, i.e. the left-most element is  $A[1]$ .

In this problem, the array  $A$  is no longer static: we need to support another operation

$$\text{shift}(i_1, i_2, i_3, \dots, i_k)(i_1 < i_2 < \dots < i_k, k > 1)$$

we do a left “circular shift” of  $A[i_1], A[i_2], \dots, A[i_k]$ .

For example, if  $A = \{6, 2, 4, 8, 5, 1, 4\}$ , then  $\text{shift}(2, 4, 5, 7)$  yields  $\{6, 8, 4, 5, 4, 1, 2\}$ . After that,  $\text{shift}(1, 2)$  yields  $8, 6, 4, 5, 4, 1, 2$ .

## Input

There will be only one test case, beginning with two integers  $n, q$  ( $1 \leq n \leq 100,000, 1 \leq q \leq 250,000$ ), the number of integers in array  $A$ , and the number of operations. The next line contains  $n$  positive integers not greater than 100,000, the initial elements in array  $A$ . Each of the next  $q$  lines contains an operation. Each operation is formatted as a string having no more than 30 characters, with no space characters inside. All operations are guaranteed to be valid.

**Warning:** The dataset is large, better to use faster I/O methods.

## Output

For each query, print the minimum value (rather than index) in the requested range.

## Sample Input

```
7 5
6 2 4 8 5 1 4
query(3,7)
shift(2,4,5,7)
query(1,4)
shift(1,2)
query(2,2)
```

## Sample Output

```
1
4
6
```