DMOPC '21 Contest 2 P2 - Scrambling Swaps

Time limit: 2.0s **Memory limit:** 256M

You have a list of swaps, initially empty. Each swap is a pair of integers (x, y), representing indices in an array of length N. Process Q of the following operations:

1. Add a swap (x, y) to the beginning of the list.
2. Add a swap (x, y) to the end of the list.
3. Output a permutation of the first N positive integers such that when the list of swaps is applied in order from beginning to end, the resulting array is a given target permutation P.

A swap (x, y) is *applied* by swapping the numbers at indices x and y.

Constraints

$2 \leq N \leq 300$

$1 \leq Q \leq 3 \times 10^6$

$1 \leq x < y \leq N$

$P_1, P_2, \ldots, P_N$ is a permutation of $1, 2, \ldots, N$.

There are at most 3000 queries of the third type.

Subtask 1 [50%]

$1 \leq Q \leq 3 \times 10^3$

Subtask 2 [50%]

No additional constraints.

Input Specification

The first line contains 2 integers N and Q.

Then Q queries follow, each given on a single line. The first character on each line is either `B`, `E`, or `Q`. If it is `B` or `E`, then two integers x and y follow, representing a swap. `B` indicates that you should add the swap to the beginning of the list, whereas `E` indicates that you should add it to the end. If the first character is `Q`, then N integers follow, representing the target permutation P.

Output Specification

For each query of the third type, output any initial permutation on a single line such that applying the list of swaps in order yields the target permutation P.

Sample Input

```
4 5
B 3 4
E 2 3
Q 2 4 1 3
E 2 3
Q 3 1 2 4
```

Sample Output

```
2 1 3 4
3 1 4 2
```

Explanation

Consider the first query. If we take `2 1 3 4` and apply the swaps $[(3,4),(2,3)]$, we obtain `2 4 1 3`.

In the second query, our swap list is $[(3,4),(2,3),(2,3)]$.