Partial Reversals in Linked List

You are given a **circular linked list** of length **N**, an integer **K** and number of queries **Q**. For each query, you're given an integer **pos**. For each query, you have to reverse subpart of a linked list starting from node at position **pos** to node at position **pos+K-1**.

The linked list is **circular**, i.e., if pos+K-1 is greater than N, then you have to reverse the linked list from position pos to position (pos+K-1)%N circularly. **Note that in this process, you don't have to update root node** (Look at sample test case for further clarification).

You have to return the final linked list configuration after processing all the queries.

Input Format

First line: Number of testcases, T.

For each test case:

First line: The size of circular linked list, N, length of reversal, K (1 <= K < N), and number of queries, Q.

Second line: space-separated integers denoting the value of nodes in linked list.

Last line: space-separated Q integers denoting the starting position of reversal, pos (0 <= pos < N).

Constraints

1 <= T <= 10

1 < N < 100000

1 <= Q <= 100

Output Format

For each test case, return the space separted integers denoting the final configuration of linked list staring from root node.

Sample Input 0

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

Sample Output 0

```
1 4 3 2
```

Explanation 0

Initial Linked List: 1->2->3->4

For query 1, Nodes with value 2 and 3 are reversed and we get the linked list 1->3->2->4.

For query 2, Nodes with value 4 and 1 are reversed and we get the linked list 1->4->3->2.

Submissions: 42
Max Score: 100
Difficulty: Medium

Rate This Challenge:

More

Current Buffer (saved locally, editable) C

```c
#include <stdio.h>
#include <string.h>
#include <math.h>
#include <stdlib.h>

int main() {

    /* Enter your code here. Read input from STDIN. Print output to STDOUT */
    return 0;
}
```

Line: 1 Col: 1

Upload Code as File Test against custom input Run Code Submit Code