Reverse First K elements of Queue

Given an integer **K** and a queue of integers, we need to reverse the order of the first **K** elements of the queue, leaving the other elements in the same relative order.

Only following standard operations are allowed on queue.

- enqueue(x) : Add an item x to rear of queue
- dequeue() : Remove an item from front of queue
- size() : Returns number of elements in queue.
- front() : Finds front item.

Input: 5 3
1 2 3 4 5

Output: 3 2 1 4 5
Explanation: After reversing the given
input from the 3rd position the resultant
output will be 3 2 1 4 5.

Example 2:

Input: 4 4
4 3 2 1

Output: 1 2 3 4
Explanation: After reversing the given
input from the 4th position the resultant
output will be 1 2 3 4.

Your Task:
Complete the **provided function modifyQueue** that takes **queue and k** as **parameters** and **returns** a **modified** queue. The **printing** is done **automatically** by the **driver code**.

Expected TIme Complexity : O(n)
Expected Auxilliary Space : O(n)

Constraints:
1 <= N <= 1000
1 <= K <= N

Note: The **Input/Ouput** format and **Example** given are used for system's internal purpose, and should be used by a user for **Expected Output** only. As it is a function problem, hence a user should

not read any input from stdin/console. The task is to complete the function specified, and not to write the full code.

```
def modifyQueue(q,k):
    # code here
    if k == len(q):
        return q[::-1]
    # else:
    #     temp = q[:k][::-1]
    #     rest = q[k:]
    #     ans = temp+rest
    #     return ans
    stack1 = []
    stack2 = []
    for i in range(k):
        stack1.append(q.pop(0))
    while stack1:
        stack2.append(stack1.pop())
    # stack2 = stack2[::-1]
    while q:
        stack2.append(q.pop(0))
    return stack2
```