

Reverse First K Elements of Queue [CodeStudio](#)

This program reverses the first 'k' elements of a queue using two different approaches: one using a stack (**reverseKElementUsingStack** function) and the other using recursion (**reverseKElementUsingRecursion** function).

Approach 1: Function to reverse the first 'k' elements of a queue using a stack

- **Description:** This approach reverses the first 'k' elements of the queue using a stack.
- **Steps:**
 1. Initialize a stack **st** to temporarily store the first 'k' elements.
 2. Push the first 'k' elements from the queue into the stack to reverse their order.
 3. Pop elements from the stack and push them back into the queue to reverse the first 'k' elements.
 4. Move the remaining elements in the queue to the front to maintain their order.
- **Time Complexity:** $O(n)$, where n is the size of the queue. We perform one pass through the queue.
- **Space Complexity:** $O(k)$, as the stack stores at most 'k' elements.

Approach 2: Function to reverse the first 'k' elements of a queue using recursion

- **Description:** This approach reverses the first 'k' elements of the queue using recursion.
- **Steps:**
 1. Define a recursive function **solve** that takes the queue, 'k', and a count as parameters.
 2. In the **solve** function:
 - If 'k' is equal to the count, return (base case).
 - Pop the front element from the queue.
 - Recursively call **solve** with the updated queue, 'k', and count+1.
 - Push the front element back into the queue.
 3. Call the **solve** function to reverse the first 'k' elements.

4. Move the remaining elements in the queue to the front to maintain their order.
- **Time Complexity: $O(n)$, where n is the size of the queue. We perform one pass through the queue using recursion.**
 - **Space Complexity: $O(k)$, as the recursion stack can have at most ' k ' frames.**