

Reverse Queue Elements [CodeStudio](#)

This program demonstrates three different approaches to reverse a queue: using a stack, using an array, and using recursion. It defines functions for each approach and showcases their usage by reversing a queue and printing the reversed queue's elements.

Approach 1: Function to reverse a queue using a stack

- **Description:** This approach reverses the queue by using a stack to temporarily store the elements.
- **Steps:**
 1. Create an empty stack.
 2. Move all elements from the queue to the stack while removing them from the queue.
 3. Move elements from the stack back to the queue to reverse their order.
- **Time Complexity:** $O(n)$, where n is the number of elements in the queue. Both pushing and popping elements from the stack involve iterating through all elements once.
- **Space Complexity:** $O(n)$, as an additional stack of the same size as the queue is used.

Approach 2: Function to reverse a queue using an array

- **Description:** This approach reverses the queue by using an array to temporarily store the elements.
- **Steps:**
 1. Create an array of the same size as the queue.
 2. Copy elements from the queue to the array in reverse order.
 3. Copy elements from the array back to the queue to reverse their order.
- **Time Complexity:** $O(n)$, where n is the number of elements in the queue. Both copying elements to and from the array involve iterating through all elements once.
- **Space Complexity:** $O(n)$, as an additional array of the same size as the queue is used.

Approach 3: Function to reverse a queue using recursion

- **Description:** This approach reverses the queue using recursion.
- **Steps:**
 1. Recursively dequeue elements from the queue until it becomes empty.
 2. When returning from the recursive calls, enqueue the elements back into the queue in reverse order.
- **Time Complexity:** $O(n)$, where n is the number of elements in the queue. Each element is dequeued and enqueued once during the recursion.
- **Space Complexity:** $O(n)$, as the recursion call stack can reach a maximum depth of n when there are n elements in the queue.