Name sineha darshan talreja

Cs231237

Dsa lab assignment 7

Q1)

```
import java.util.LinkedList;

import java.util.Queue;

import java.util.Stack;


public class QueueOperations {

  public static void reverseFirstKElements(Queue<Integer> queue, int k) {

    if (queue == null || k > queue.size() || k <= 0) {

      throw new IllegalArgumentException("Invalid value of k or queue is null");

    }


    Stack<Integer> stack = new Stack<>();

    for (int i = 0; i < k; i++) {

      stack.push(queue.poll());

    }

    while (!stack.isEmpty()) {

      queue.offer(stack.pop());

    }

    int size = queue.size();

    for (int i = 0; i < size; i++) {

      queue.offer(queue.poll());

    }

  }

  public static int getMinimum(Queue<Integer> queue) {
    if (queue.isEmpty()) {
        throw new IllegalStateException("Queue is empty");
    }

    int min = Integer.MAX_VALUE;
    int size = queue.size();
    for (int i = 0; i < size; i++) {
        int current = queue.poll();
        if (current < min) {
            min = current;
        }
        queue.offer(current);
    }

    return min;
}

public static void main(String[] args) {
    Queue<Integer> queue = new LinkedList<>();
    queue.offer(10);
    queue.offer(20);
    queue.offer(30);
    queue.offer(40);
    queue.offer(50);
    int k = 3;
    System.out.println("Original Queue: " + queue);
    reverseFirstKElements(queue, k);
    System.out.println("Queue after reversing first " + k + " elements: " + queue);
```

```
        System.out.println("Minimum element in the Queue: " + getMinimum(queue));
    }
}
```