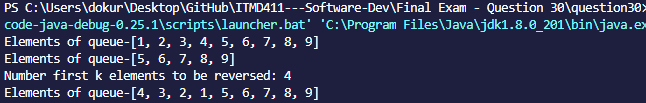
**Final Exam – Question 30**

By: Laura Pereda

Screenshot of the output for given example:



Source code for program:

package *app*;

import *java.util.LinkedList*;

import *java.util.Queue*;

import *java.util.Stack*;

*public* *class* App {

*public* *static* void main(String[] args) *throws* Exception {

*// Create an empty queue*

        Queue<Integer> q = new LinkedList<>();

*// Fill up queue*

        for (int i = 1; i < 10; i++){

            q.add(i);

        }

*// Check the elements of the queue*

        System.out.println("Elements of queue-" + q);

*// Call the reverseK function (question 30)*

        App temp = new App();

        temp.reverseK(q, 4);

    }

*// Reverses the order of the first k elements of the queue*

*public* void reverseK(Queue<Integer> q, int k){

        Queue<Integer> q2 = new LinkedList<>();

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

*// Placing into stack will reverse it*

        for(int i = 0; i < k; i++){

            stack.push(q.remove());

        }

*// Verify that the k elements were removed*

        System.out.println("Elements of queue-" + q);

*// Add these to the second queue*

        for(int i = 0; i < k; i++){

            q2.add(stack.pop());

        }

*// Store size of q before removing final elements*

        int remainder = q.size();

*// Add the remainder non-reversed elements to second queue*

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

            q2.add(q.remove());

        }

        System.out.println("Number first k elements to be reversed: " + k);

        System.out.println("Elements of queue-" + q2);

    }

}