Queue Interface

Assignment 2

```
import java.util.Deque;
import java.util.ArrayDeque;


class Tester {


   public static Deque<Integer> changeSmallest(Deque<Integer> inputStack) {
      if (inputStack.isEmpty()) {
         return inputStack;
      }


      // Step 1: Find the smallest value
      int smallest = Integer.MAX_VALUE;
      Deque<Integer> tempStack = new ArrayDeque<>();
      while (!inputStack.isEmpty()) {
         int current = inputStack.pop();
         if (current < smallest) {
            smallest = current;
         }
         tempStack.push(current);
      }


      // Step 2: Reorder elements, putting the smallest values at the bottom
      Deque<Integer> resultStack = new ArrayDeque<>();
      Deque<Integer> smallestStack = new ArrayDeque<>();
```

```
    while (!tempStack.isEmpty()) {
        int current = tempStack.pop();
        if (current == smallest) {
            smallestStack.push(current);
        } else {
            resultStack.push(current);
        }
    }

    // Step 3: Combine the stacks
    while (!smallestStack.isEmpty()) {
        resultStack.push(smallestStack.pop());
    }

    while (!resultStack.isEmpty()) {
        inputStack.push(resultStack.pop());
    }

    return inputStack;
}

public static void main(String[] args) {

    Deque<Integer> inputStack = new ArrayDeque<>();
    inputStack.push(10);
    inputStack.push(8);
```

```
        inputStack.push(5);
        inputStack.push(12);
        inputStack.push(5);

        Deque<Integer> updatedStack = changeSmallest(inputStack);

        System.out.println("Stack After Modification:");
        for (Integer value : updatedStack) {
            System.out.println(value);
        }
    }
}
```

```
C:\Users\Sarvesh\OneDrive\Desktop>java Tester4
Stack After Modification:
10
8
12
5
5
```