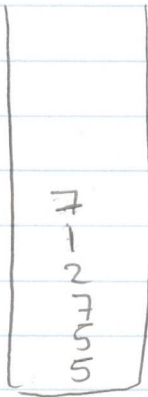Assignment 2

Q.1

[a] Since there was 44 enques and only one of the deques
returned a value, therefore that means that the
size of the queue is 44-1= [43]

[b] push(5), top(), push(5), push(7), push(2), push(3), push(9), pop
top(), pop(), push(1), push(2), top(), pop(), push(7), pop(), pop()

Values returned ⟹ (5,9,3,3,2,2,7,1)

```
7
1
2
7
5
5
```

[c] addFirst(7), size(), first(), last(), addlast(2), addFirst(13), addlas
first(), addFirst(11), last(), addFirst(1), removelast(), addfirst(9), addlas
addlast(8), first(), last(), addlast(4), size()

first   last
9 - 1 - 11 - 13 - 3 - 2 - 6 - 8 - 4

```java
1  import java.util.Stack;
2
3  /**
4   * Meetings
5   */
6  public class Meetings {
7
8      public static Stack<int[]> meeting(int max, int[][] timeSlots){
9          Stack<int[]> times = new Stack<int[]>();
10         for (int i = 0; i < timeSlots.length; i++) {
11             if(i == 0){
12                 times.push(timeSlots[i]);
13             }else{
14                 if(times.peek()[1]<timeSlots[i][1] && times.peek()[1]>=timeSlots[i][0]){
15                     int[] temp =times.pop();
16                     int[] merged = new int[]{temp[0],timeSlots[i][1]};
17                     times.push(merged);
18                 }else if(times.peek()[1]<timeSlots[i][0] && times.peek()[1]<timeSlots[i][1]){
19                     times.push(timeSlots[i]);
20                 }
21             }
22         }
23         int s = times.size();
24         Stack<int[]> Results = new Stack<int[]>();
25         int[] temp = times.peek();
26         for (int i = 0; i < s; i++) {
27             if(i==0){
28                 if(times.peek()[1]<max){
29
30                     int[] adding  = new int[]{temp[1],max};
31                     Results.push(adding);
32                     times.pop();
33                 }
34             }else if (i==s-1){
35                 int[] gap = new int[]{times.peek()[1],temp[0]};
36                 Results.push(gap);
37
38                 if(times.peek()[1]>0){
39                     int[] temp2 = times.pop();
40                     int[] adding  = new int[]{0,temp2[0]};
41                     Results.push(adding);
42                 }
43             }else{
44                 int[] gap = new int[]{times.peek()[1],temp[0]};
45                 Results.push(gap);
46                 temp = times.pop();
47             }
48         }
49         return Results;
50     }
   Run | Debug
51     public static void main(String[] args) {
52         int[][] time = new int[][]{{1,9},{2,5},{4,10},{15,20},{20,25},{30,45}};
53         Stack<int[]>res = meeting(50,time);
54         int s = res.size();
55         for (int i = 0; i <s ; i++) {
56             System.out.println("["+res.peek()[0]+","+res.pop()[1]+"]");
57         }
58     }
59 }
```