PRN: 52

DSA LAB Exam

1. Write a Java program to

a. Perform insertion sort

IDE

Window Help

InsertionSort.java

```
1  package com.main;
2  import java.util.Scanner;
3
4  public class InsertionSort {
5
6          public static void main(String[] args)
7             {
8                int n, i, j, element;
9                Scanner scan = new Scanner(System.in);
10
11               System.out.print("Enter the Size of Array: ");
12               n = scan.nextInt();
13               int[] arr = new int[n];
14               System.out.print("Enter " +n+ " Elements: ");
15               for(i=0; i<n; i++)
16                  arr[i] = scan.nextInt();
17
18               for(i=1; i<n; i++)
19               {
20                  element = arr[i];
21
22                  for(j=(i-1); j>=0 && arr[j]>element; j--)
23                     arr[j+1] = arr[j];
24
25                  arr[j+1] = element;
26               }
27
28               System.out.println("\nThe new sorted array is: ");
29               for(i=0; i<n; i++)
30                  System.out.print(arr[i]+ " ");
31            }
32         }
33
```

Output:

```
<terminated> InsertionSort [Java Application] C:\Program Fi
Enter the Size of Array: 5
Enter 5 Elements: 30
33
42
22
10

The new sorted array is:
10 22 30 33 42
```

b. Implement queue using array

PRN: 52

ipse IDE

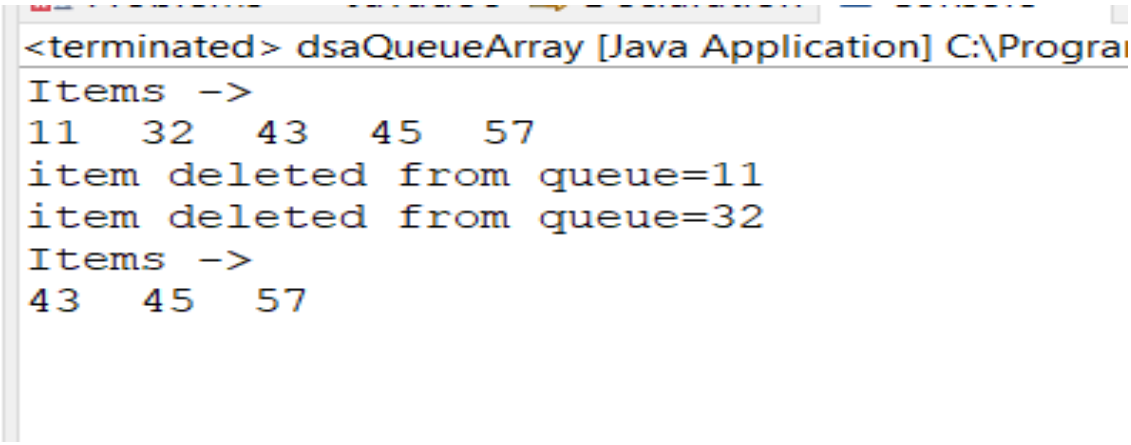
Window Help

InsertionSort.java | dsaQueueArray.java | QueueArrMain.java

```
1  package com.main;
2  import java.util.*;
3
4
5          public class dsaQueueArray {
6
7          int[] queueArray;
8          int front, rear, size;
9
10         public dsaQueueArray(int size) {
11             this.size = size;
12             queueArray = new int[size];
13             front = -1;
14             rear = -1;
15         }
16
17         public boolean isFull() {
18             return (rear == size - 1);
19         }
20
21         public boolean isEmpty() {
22             return (front == -1 || front > rear);
23         }
24
25         public void enqueue(int item) {
26             if (isFull()) {
27                 System.out.println("Queue is full!");
28                 return;
29             }
30             queueArray[++rear] = item;
31             if (front == -1) {
32                 front = 0;
33             }
34         }
35
```

```
34         }
35
36         public int dequeue() {
37             if (isEmpty()) {
38                 System.out.println("Queue is empty!");
39                 return -1;
40             }
41             return queueArray[front++];
42         }
43
44         void display()
45        {
46            int i;
47             if (isEmpty()) {
48               System.out.println("Empty Queue");
49             }
50             else {
51
52               // display element of the queue
53               System.out.println("Items -> ");
54               for (i = front; i <= rear; i++)
55                 System.out.print(queueArray[i] + "   ");
56                 System.out.println(" ");
57
58
59             }
60           }
```

PRN: 52

```
62⊖         public static void main(String[] args) {
63          dsaQueueArray q = new dsaQueueArray(5);
64          q.enqueue(11);
65          q.enqueue(32);
66          q.enqueue(43);
67          q.enqueue(45);
68          q.enqueue(57);
69          q.display();
70          System.out.println("item deleted from queue=" + q.dequeue());
71          System.out.println("item deleted from queue=" +q.dequeue());
72          q.display();
73      }
74  }
```

Output: