

DATA STRUCTURE AND ALGORITHMS(Lab Exam)

Name : *Funde Pandurang Gahininath*

Date : 22-01-2022

1. Write a Java program to

I] Implement circular queue using arrays

```
package com.labexam.entity;

import java.util.ArrayList;

class CircularQueue{

    private int size, front, rear;

    private ArrayList<Integer> queue = new ArrayList<Integer>();

    CircularQueue(int size)

    {

        this.size = size;

        this.front = this.rear = -1;

    }

    public void enqueue(int data)

    {

        if((front == 0 && rear == size - 1) ||

        (rear == (front - 1) % (size - 1)))

        {

            System.out.print("Queue is Full");
```

```
}  
  
else if(front == -1)  
{  
  
    front = 0;  
  
    rear = 0;  
  
    queue.add(rear, data);  
  
}  
  
else if(rear == size - 1 && front != 0)  
{  
  
    rear = 0;  
  
    queue.set(rear, data);  
  
}  
  
else  
{  
  
    rear = (rear + 1);  
  
    if(front <= rear)  
    {  
  
        queue.add(rear, data);  
  
    }  
  
    else  
    {  
  
        queue.set(rear, data);  
  
    }  
  
}
```

```
}  
  
}  
  
}  
  
public int deQueue()  
  
{  
  
    int temp;  
  
    if(front == -1)  
    {  
  
        System.out.print("Queue is Empty");  
  
        return -1;  
  
    }  
  
    temp = queue.get(front);  
  
    if(front == rear)  
    {  
  
        front = -1;  
  
        rear = -1;  
  
    }  
  
    else if(front == size - 1)  
    {  
  
        front = 0;  
  
    }  
  
    else
```

```
{  
  
front = front + 1;  
  
}  
  
return temp;  
  
}  
  
public void displayQueue()  
  
{  
  
if(front == -1)  
  
{  
  
System.out.print("Queue is Empty");  
  
return;  
  
}  
  
System.out.print("Elements in the " +  
  
"circular queue are: ");  
  
if(rear >= front)  
  
{  
  
for(int i = front; i <= rear; i++)  
  
{  
  
System.out.print(queue.get(i));  
  
System.out.print(" ");  
  
}  
  
System.out.println();  
  
}
```

```
}  
  
else  
  
{  
  
for(int i = front; i < size; i++)  
  
{  
  
System.out.print(queue.get(i));  
  
System.out.print(" ");  
  
}  
  
for(int i = 0; i <= rear; i++)  
  
{  
  
System.out.print(queue.get(i));  
  
System.out.print(" ");  
  
}  
  
System.out.println();  
  
}  
  
}  
  
public static void main(String[] args)  
  
{  
  
CircularQueue q = new CircularQueue(5);  
  
q.enqueue(17);  
  
q.enqueue(22);  
  
q.enqueue(13);
```

```
q.enqueue(-8);

q.displayQueue();

int x = q.dequeue();

if(x != -1)

{

System.out.print("Deleted value = ");

System.out.println(x);

}

x = q.dequeue();

if(x != -1)

{

System.out.print("Deleted value = ");

System.out.println(x);

}

q.displayQueue();

q.enqueue(13);

q.enqueue(50);

q.enqueue(57);

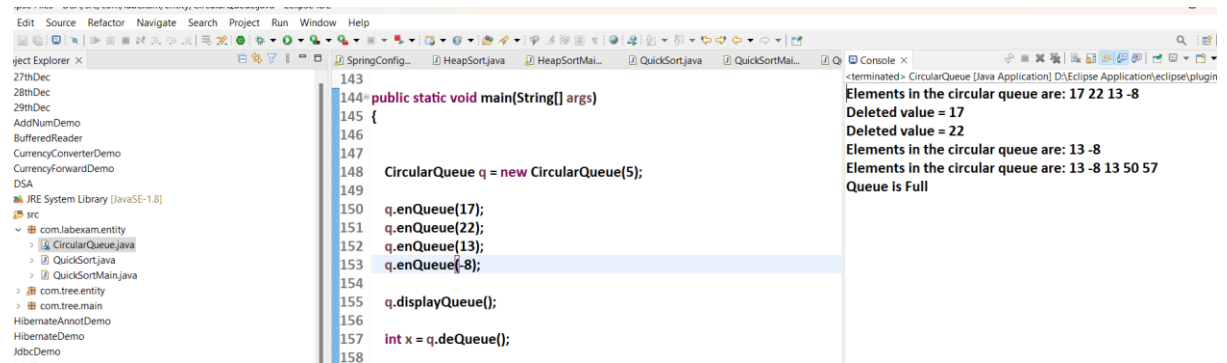
q.displayQueue();

q.enqueue(2);

}

}
```

OUTPUT:



II] Perform quick sort to arrange given set of elements

```
package com.labexam.entity;
```

```
public class QuickSort {
```

```
private int[] arr;
```

```
public QuickSort(int[] arr) {
```

```
this .arr=arr;
```

```
}
```

```
void swap(int i,int j) {
```

```
int temp=arr[i];
```

```
arr[i]=arr[j];
```

```
arr[j]=temp;
```

```
}
```

```
int partition(int start ,int end) {
```

```
int pivot=arr[end];
```

```
int i=(start-1);
```

```
for(int j =start; j <=end-1;j++) {
```

```
if(arr[j]<pivot)
```

```
{
```

```
    i++;
```

```
    swap(i,j);
```

```
}
```

```
}
```

```
swap(i+1,end);
```

```
return(i+1);
```

```
}
```

```
void quicksort(int start,int end)
```

```
{
```

```
if(start<end)
```

```
{
```

```
int i = partition(start,end);
```

```
quicksort(start, i-1);
```

```
quicksort(i+1, end);
```

```
}
```

```
}
```

```
void printArray()
```

```
{
```

```
int size = arr.length;
```



```
for(int i=0;i<size;i++)  
  
System.out.println(arr[i]+" ");  
  
System.out.println( );  
  
}  
  
}
```

Main Class:

```
package com.labexam.entity;  
  
public class QuickSortMain {  
  
    public static void main(String[] args) {  
  
        int[] arr= { 9,81,68,1,6,15,93,59,54,200};  
  
        int n =arr.length;  
  
        QuickSort qs=new QuickSort(arr);  
  
        qs.quicksort(0, n-1);  
  
        System.out.println(" < Quick Sort > Sorted array : ");  
  
        qs.printArray();  
  
    }  
  
}
```

OUTPUT:

