

## Queue

### Exercise

```
class Queue {

    private int front;
    private int rear;
    private int maxSize;
    private int[] arr;

    Queue(int maxSize)
    {
        this.front = 0;
        this.rear = -1;
        this.maxSize = maxSize;
        this.arr = new int[this.maxSize];
    }

    public boolean isFull()
    {
        return rear == maxSize -
        1;
    }

    public boolean enqueue(int data)
    {
        if (isFull()) {
            return false;
        } else {
            arr[++rear] = data;
        }
    }
}
```

```
        return true;
    }
}
```

```
public void display()
{
    if (isEmpty()) {
        System.out.println("Queue is empty!");
    } else {
        for (int index = front; index <= rear; index++)
            { System.out.println(arr[index]);
              }
    }
}
```

```
public boolean isEmpty()
{
    return front > rear;
}
```

```
public int dequeue()
{
    if (isEmpty()) {
        return Integer.MIN_VALUE;
    } else {
        int data = arr[this.front];
        arr[front++] = Integer.MIN_VALUE;
        return data;
    }
}
```

```
public int getMaxSize()
    { return maxSize;
    }
}
```

```
class Tester {
```

```
    public static void main(String[] args) {
```

```
        Queue queue = new Queue(7);
```

```
        queue.enqueue(2);
```

```
        queue.enqueue(7);
```

```
        queue.enqueue(9);
```

```
        queue.enqueue(4);
```

```
        queue.enqueue(6);
```

```
        queue.enqueue(5);
```

```
        queue.enqueue(10);
```

```
        Queue[] queueArray = splitQueue(queue);
```

```
        System.out.println("Elements in the queue of odd numbers");
```

```
        queueArray[0].display();
```

```
        System.out.println("\nElements in the queue of even numbers");
```

```
        queueArray[1].display();
```

```
    }
```

```

public static Queue[] splitQueue(Queue queue) {
    Queue oddQueue = new Queue(queue.getMaxSize());
    Queue evenQueue = new Queue(queue.getMaxSize());

    while (!queue.isEmpty())
    {
        int data =
        queue.dequeue(); if
        (data % 2 == 0) {
            evenQueue.enqueue(data);
        } else {
            oddQueue.enqueue(data);
        }
    }

    return new Queue[]{oddQueue, evenQueue};
}
}

```

```

Elements in the queue of odd numbers
7
9
5

Elements in the queue of even numbers
2
4
6
10

```