Lab #2
CP2530
Queue programming

Main.java

Run

Output

```java
class CircularQueue {
    int front, rear, size;
    int[] array;

public CircularQueue(int capacity) {
    front = rear = 0;
    size = capacity;
    array = new int[capacity];
}

public void enqueue(int data) {
    //expands array to size 10 if the array is less than 10
    if ((rear + 1) % size == front) {
        if (size < 10) {
            int[] tempArray = new int[10];
            int i = 0;
            while (front != rear) {
                front = (front + 1) % size;
                tempArray[i] = array[front];
                i++;
            }
            front = 0;
            rear = i;
            size = 10;
            array = tempArray;
        } else {
            //prints when array is full
            System.out.println("Queue is full");
            return;
        }
    }
    rear = (rear + 1) % size;
    array[rear] = data;
}

```

```
java -cp /tmp/UtoERke3Nf Main
1 1 2
1 2 3
1 2 3 4
Queue size is 5
2 3 4 0 5
2 3 4 0 5 6
Dequeued: 2
3 4 0 5 6
Dequeued: 3
4 0 5 6
4 0 5 6 6
4 0 5 6 6 7
Queue size is 10
4 0 5 6 6 7 8


Queue is full

Dequeued: 4
```

```
35
36  public int dequeue() {
37      //removes integers from queue
38      if (front == rear) {
39          System.out.println("Queue is empty");
40          return -1;
41      }
42      front = (front + 1) % size;
43      return array[front];
44  }
45
46  //displays queue to see what is happening as the code executes
47  public void display() {
48      int i = front + 1;
49      while (i <= rear) {
50          System.out.print(array[i] + " ");
51          i++;
52      }
53      System.out.println();
54   }
55  }
56
57  class Main {
58      public static void main(String[] args) {
59          CircularQueue queue = new CircularQueue(5);
60          queue.enqueue(1);
61          queue.display();
62          queue.enqueue(2);
63          queue.display();
64          queue.enqueue(3);
65          queue.display();
66          queue.enqueue(4);
67          queue.display();
68          if (queue.size == 5) {
69              System.out.println("Queue size is 5");
```

```
69              System.out.println("Queue size is 5");
70          } else if (queue.size == 10) {
71              System.out.println("Queue size is 10");
72          }
73          queue.enqueue(5);
74          queue.display();
75          queue.enqueue(6);
76          queue.display();
77          System.out.println("Dequeued: " + queue.dequeue());
78          queue.display();
79          System.out.println("Dequeued: " + queue.dequeue());
80          queue.display();
81          queue.enqueue(6);
82          queue.display();
83          queue.enqueue(7);
84          queue.display();
85          if (queue.size == 5) {
86              System.out.println("Queue size is 5");
87          } else if (queue.size == 10) {
88              System.out.println("Queue size is 10");
89          }
90          queue.enqueue(8);
91          queue.display();
92          queue.enqueue(9);
93          queue.display();
94          queue.enqueue(10);
95          queue.display();
96          queue.enqueue(11);
97          queue.display();
98          System.out.println("Dequeued: " + queue.dequeue());
99          queue.display();
100     }
101 }
```