EC4070: Data Structures and Algorithms

LAB 04

K.J.M.U.G.S. Eranda Jayasinghe

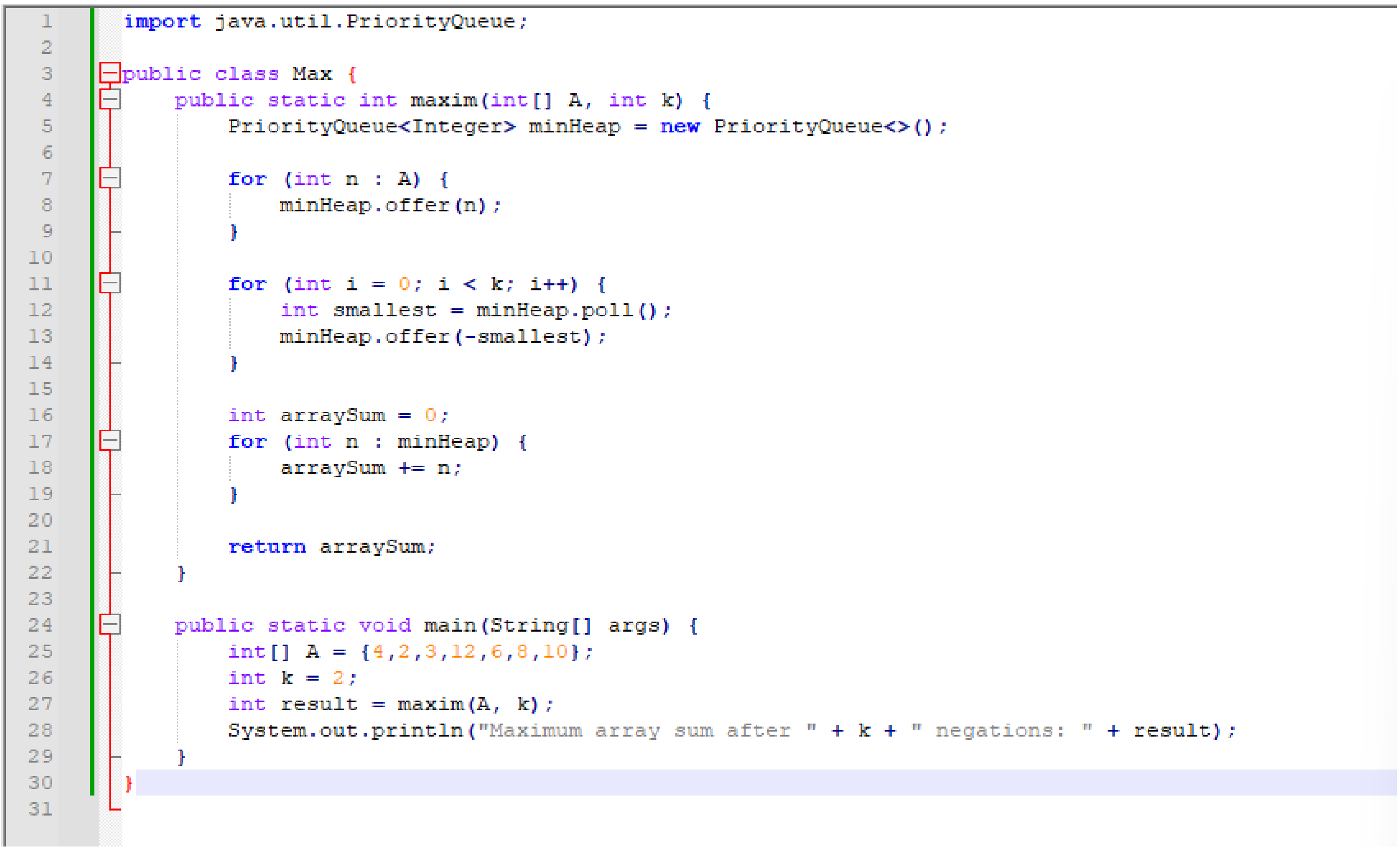
2021/E/075

SEMESTER 4

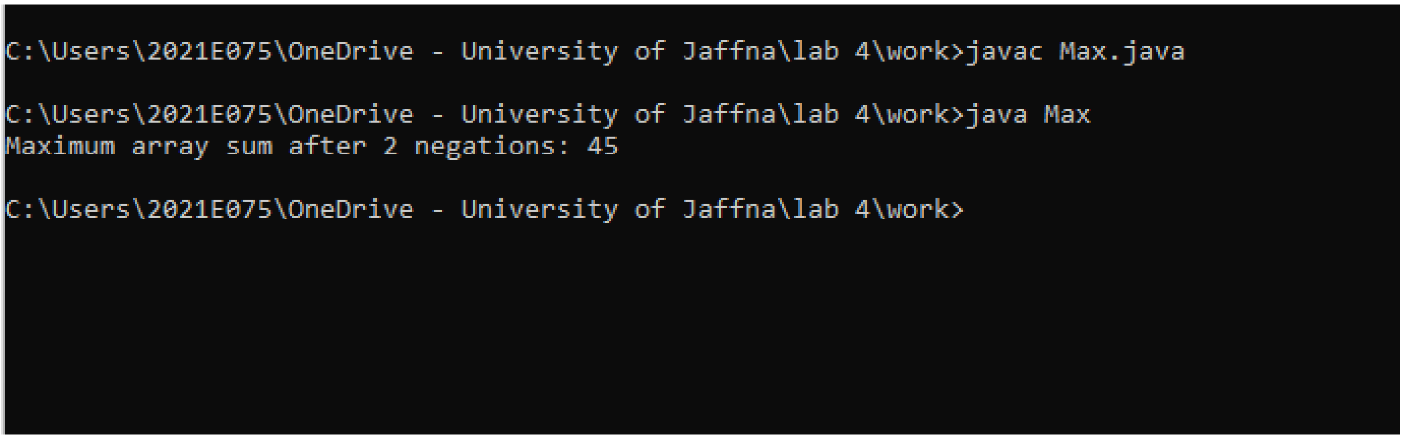
EC4070

18.10.2023

Q1.

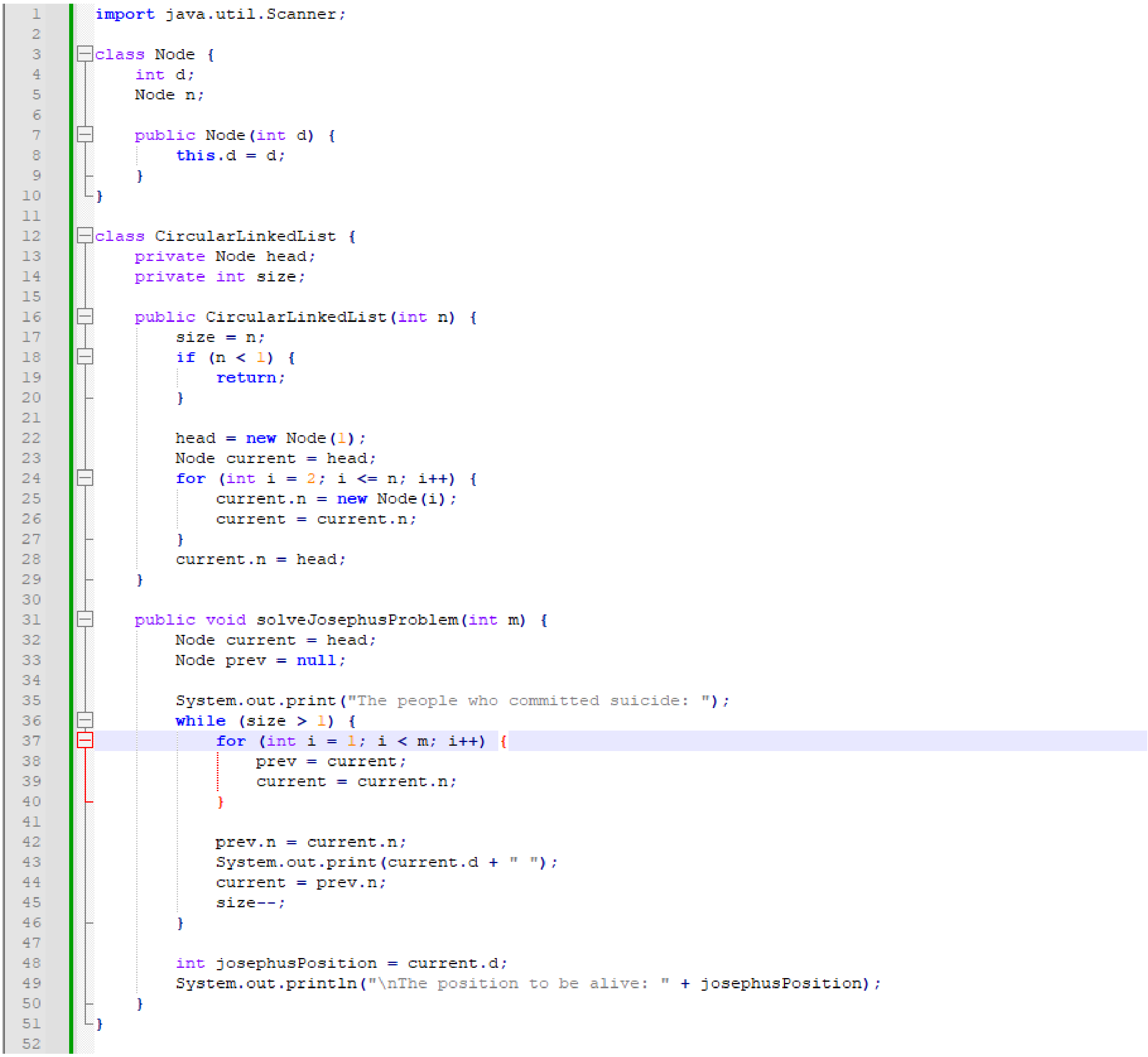


```java
import java.util.PriorityQueue;

public class Max {
    public static int maxim(int[] A, int k) {
        PriorityQueue<Integer> minHeap = new PriorityQueue<>();

        for (int n : A) {
            minHeap.offer(n);
        }

        for (int i = 0; i < k; i++) {
            int smallest = minHeap.poll();
            minHeap.offer(-smallest);
        }

        int arraySum = 0;
        for (int n : minHeap) {
            arraySum += n;
        }

        return arraySum;
    }

    public static void main(String[] args) {
        int[] A = {4,2,3,12,6,8,10};
        int k = 2;
        int result = maxim(A, k);
        System.out.println("Maximum array sum after " + k + " negations: " + result);
    }
}
```

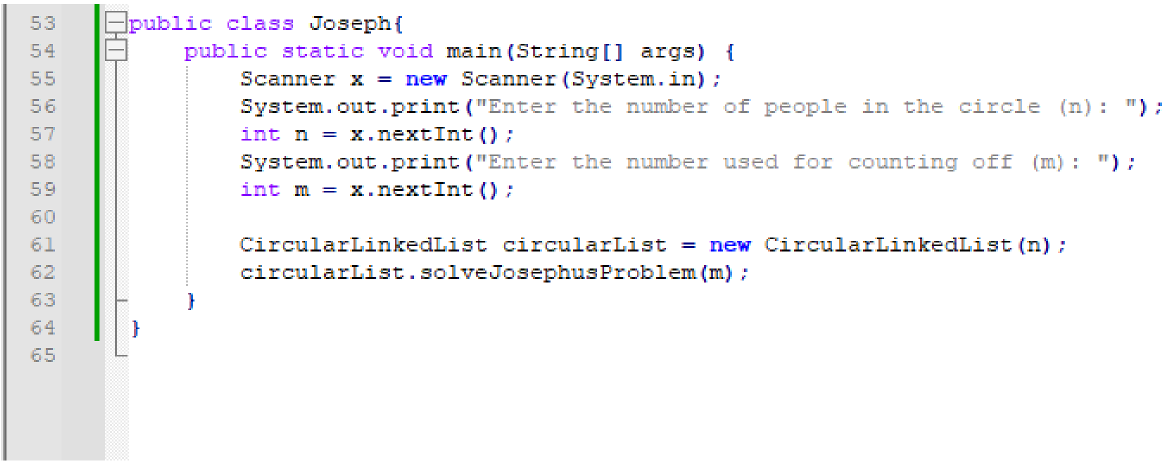


```
C:\Users\2021E075\OneDrive - University of Jaffna\lab 4\work>javac Max.java

C:\Users\2021E075\OneDrive - University of Jaffna\lab 4\work>java Max
Maximum array sum after 2 negations: 45

C:\Users\2021E075\OneDrive - University of Jaffna\lab 4\work>
```

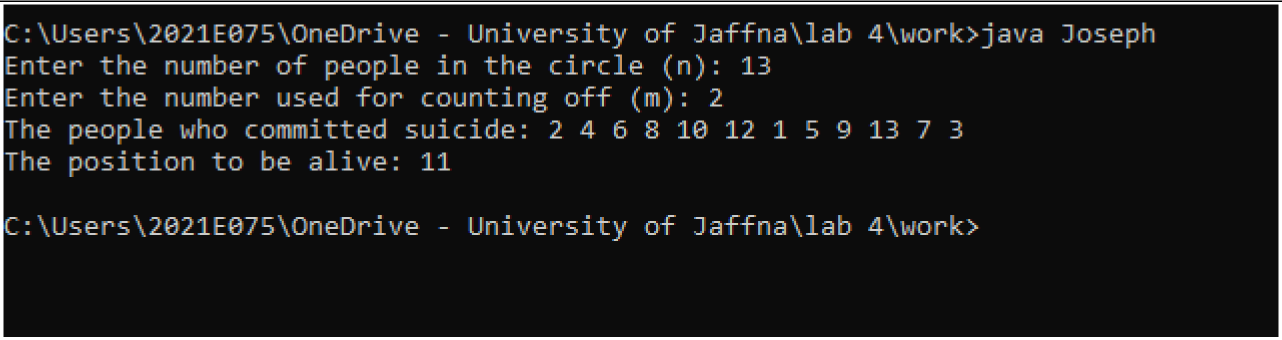
Q2.



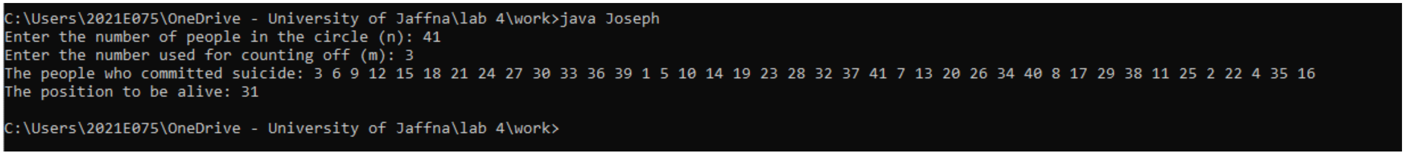
```
import java.util.Scanner;

class Node {
    int d;
    Node n;

    public Node(int d) {
        this.d = d;
    }
}

class CircularLinkedList {
    private Node head;
    private int size;

    public CircularLinkedList(int n) {
        size = n;
        if (n < 1) {
            return;
        }

        head = new Node(1);
        Node current = head;
        for (int i = 2; i <= n; i++) {
            current.n = new Node(i);
            current = current.n;
        }
        current.n = head;
    }

    public void solveJosephusProblem(int m) {
        Node current = head;
        Node prev = null;

        System.out.print("The people who committed suicide: ");
        while (size > 1) {
            for (int i = 1; i < m; i++) {
                prev = current;
                current = current.n;
            }

            prev.n = current.n;
            System.out.print(current.d + " ");
            current = prev.n;
            size--;
        }

        int josephusPosition = current.d;
        System.out.println("\nThe position to be alive: " + josephusPosition);
    }
}

```



```
public class Joseph{
    public static void main(String[] args) {
        Scanner x = new Scanner(System.in);
        System.out.print("Enter the number of people in the circle (n): ");
        int n = x.nextInt();
        System.out.print("Enter the number used for counting off (m): ");
        int m = x.nextInt();

        CircularLinkedList circularList = new CircularLinkedList(n);
        circularList.solveJosephusProblem(m);
    }
}

```



```
C:\Users\2021E075\OneDrive - University of Jaffna\lab 4\work>java Joseph
Enter the number of people in the circle (n): 13
Enter the number used for counting off (m): 2
The people who committed suicide: 2 4 6 8 10 12 1 5 9 13 7 3
The position to be alive: 11

C:\Users\2021E075\OneDrive - University of Jaffna\lab 4\work>
```



```
C:\Users\2021E075\OneDrive - University of Jaffna\lab 4\work>java Joseph
Enter the number of people in the circle (n): 41
Enter the number used for counting off (m): 3
The people who committed suicide: 3 6 9 12 15 18 21 24 27 30 33 36 39 1 5 10 14 19 23 28 32 37 41 7 13 20 26 34 40 8 17 29 38 11 25 2 22 4 35 16
The position to be alive: 31

C:\Users\2021E075\OneDrive - University of Jaffna\lab 4\work>
```