GRAPH THEORY ASSIGNMENT

Name – Hirak Jyoti Das

Roll No – 200710007024

Branch – Computer Science & Engineering

```java
// Write a program to check whether euler path , euler circuit
//exists in a graph or not.

import java.io.*;
import java.util.*;
import java.util.LinkedList;

class EulerInJava {
    int V;
    LinkedList<Integer> adj[];

    EulerInJava(int v) {
        V = v;
        adj = new LinkedList[v];
        for (int i = 0; i < v; ++i)
            adj[i] = new LinkedList();
    }

    void addEdge(int v1, int v2) {
        adj[v1].add(v2);
        adj[v2].add(v1);
    }

    void visit(int v, boolean visited[]) {
        visited[v] = true;
        Iterator<Integer> i = adj[v].listIterator();
        while (i.hasNext()) {
            int n = i.next();
            if (!visited[n])
                visit(n, visited);
        }
    }

    boolean isConnected() {
        boolean visited[] = new boolean[V];
        int i;
        for (i = 0; i < V; i++)
            visited[i] = false;
        for (i = 0; i < V; i++)
            if (adj[i].size() != 0)
                break;
        if (i == V)
            return true;
        visit(i, visited);
        for (i = 0; i < V; i++)
            if (visited[i] == false && adj[i].size() > 0)
                return false;

        return true;
    }

    int isEulerian() {
        if (isConnected() == false)
            return 0;
        int odd = 0;
        for (int i = 0; i < V; i++)
            if (adj[i].size() % 2 != 0)
                odd++;
        if (odd > 2)
            return 0;
        return (odd == 2) ? 1 : 2;
    }

    void test() {
        int res = isEulerian();
        if (res == 0)
            System.out.println("The graph is does not have an Euler
path or an Euler circuit");
        else if (res == 1)
            System.out.println("The graph has a Euler path");
        else
            System.out.println("The graph has a Euler circuit");
    }

    public static void main(String args[]) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the number of vertices in the graph:
");
        int verti = sc.nextInt();
        EulerInJava obj = new EulerInJava(verti);
        char c = 'y';
        System.out.println("*The numbering of the vetices of the graph
must start from 0");
        while (c == 'y' || c == 'Y') {
            System.out.println("Enter the first vertice of the edge:");
            int v1 = sc.nextInt();
            System.out.println("Enter the second vertice of the
edge:");
            int v2 = sc.nextInt();
            obj.addEdge(v1, v2);
            System.out.println("Enter 'y' if more edges exists:");
            c = sc.next().charAt(0);
        }
        obj.test();
    }
}
```