CHANDIGARH UNIVERSITY

UNIVERSITY INSTITUTE OF ENGINEERING

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Submitted By: Rajiv Paul	**Submitted To:** Urvashi Malhotra
Subject Name	Competitive Coding-I
Subject Code	20CSP-314
Branch	BE-CSE
Semester	5th

LAB INDEX

Sr. No	Program	Date	Evaluatio				Sign
			LW (12	VV (8)	FW (10	Total (30	
1.							
2.							
3.							
4.							
5.	To solve the following hacker rank problems based on Graphs.	06/10/22					
6.							
7.							
8.							
9.							
10.							

EXPERIMENT – 2.1

Student Name:Rajiv Paul	**UID:20BCS1812**
Branch: CSE	**Section/Group: 20BCS_WM_702(A)**
Semester: 5th	**Date of Performance:06/10/2022**
Subject Name: Competitive Coding	**Subject Code: 20CSP-314**

AIM OF THE EXPERIMENT:

To solve the following hacker rank problems based on Graphs.

Problem 1: https://www.hackerrank.com/challenges/journey-to-the-moon/problem?isFullScreen=true

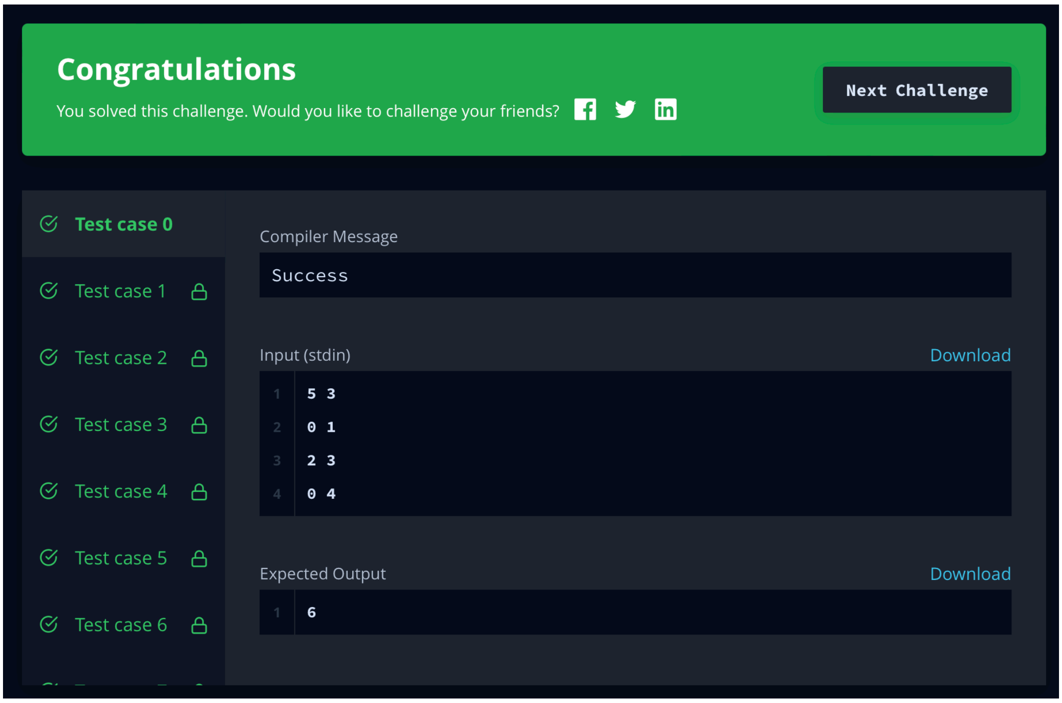
1. PROGRAM CODE:

```
import java.io.*;
import java.util.*;


public class Solution {


    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);


        int N = in.nextInt();


        Graph graph = new Graph(N);


        int P = in.nextInt();
    for(int i = 0; i < P; i++){
        int source = in.nextInt();
        int destination = in.nextInt();

        graph.addEdge(source,destination);
    }



    boolean[] visited = new boolean[N];
    List<Integer> countries = new ArrayList<Integer>();
    long combinations = 0;

    // store size of each country by traversing each cluster
    for(int i = 0; i < N; i++){
        if(!visited[i]){
            countries.add(graph.dfs(i, visited));
        }
    }



    int sum = 0;
    for(int country : countries){
        combinations += sum*country;
        sum += country;
    }

    System.out.println(combinations);
}
```

```
}

class Graph{
    List<Integer>[] vertices;

    public Graph(int count){
        vertices = new ArrayList[count];

        for(int i = 0; i < count; i++){
            vertices[i] = new ArrayList<Integer>();
        }
    }

    public void addEdge(int source, int destination){
        vertices[source].add(destination);
        vertices[destination].add(source);
    }

    // modified DFS to return number of vertices traversed
    public int dfs(int source, boolean[] visited){
        visited[source] = true;
        int count = 1;

        for(Integer vertex: vertices[source]){
            if(!visited[vertex]){
                count += dfs(vertex, visited);
            }
        }
```

```
            return count;
        }
    }
```

2. OUTPUT:



Compiler Message

```
Success
```

Input (stdin)

```
5 3
0 1
2 3
0 4
```

Expected Output

```
6
```

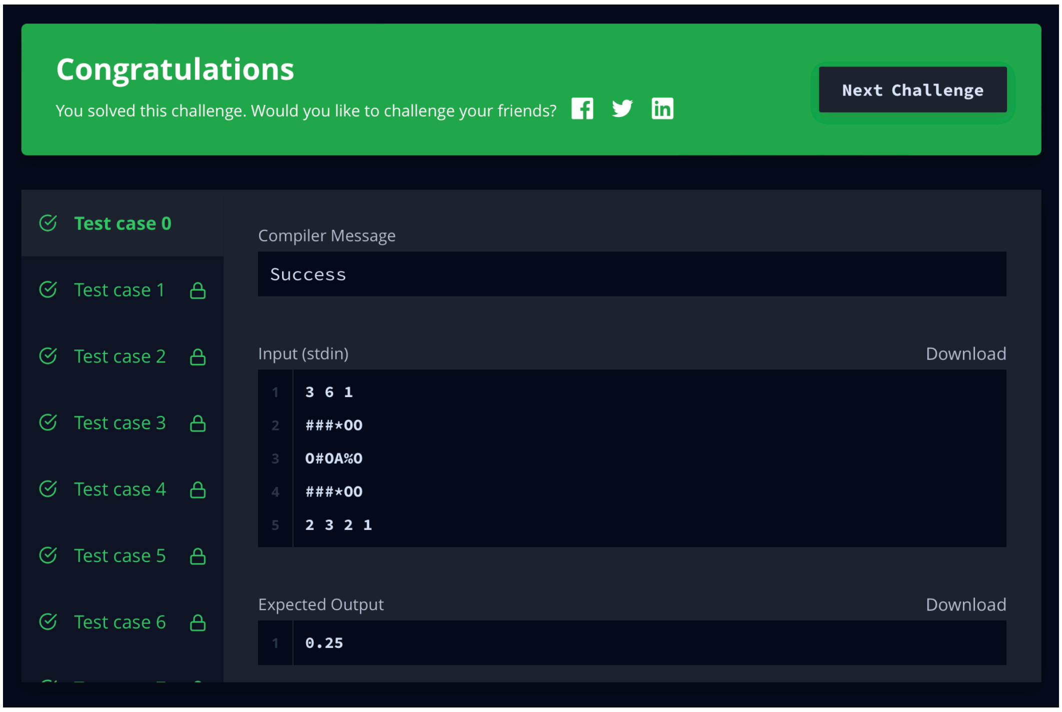
Problem 2: https://www.hackerrank.com/challenges/frog-in-maze/problem?isFullScreen=true

1. PROGRAM CODE:

```
import java.util.*;

public class Solution002 {
    static final int EXIT = Integer.MAX_VALUE;
    public static void main(String[] args) {
        java.util.Scanner sc = new java.util.Scanner(System.in);
        int n = sc.nextInt(), m = sc.nextInt(), k = sc.nextInt();
        sc.nextLine();
        int[][] nextAry2 = new int[n + 2][m + 2];
        int[][] ids = new int[n + 2][m + 2];
        int ax = -1, ay = -1, id = 0;
        for (int i = 1; i <= n; ++i) {
            char[] typeLine = sc.nextLine().toCharArray();
            for (int j = 1; j <= m; ++j) {
                switch (typeLine[j - 1]) {
                case '*':
                    nextAry2[i][j] = 1;
                    break;
                case '#':
                    nextAry2[i][j] = 0;
                    break;
                case '%':
                    nextAry2[i][j] = EXIT;
                    break;
                case 'A':
                    ax = i;
                    ay = j;
                default:
                    nextAry2[i][j] = (i << 16) | j;
                }
            }
        }
        for (int i = 0; i < k; ++i) {
            int x0 = sc.nextInt(), y0 = sc.nextInt(), x1 = sc.nextInt(), y1 = sc.nextInt();
            nextAry2[x0][y0] = (x1 << 16) | y1;
            nextAry2[x1][y1] = (x0 << 16) | y0;
        }
        for (int i = 1; i <= n; ++i)
            for (int j = 1; j <= m; ++j)
                ids[i][j] = nextAry2[i][j] > 1 ? id++ : -1;

        double[][] T = new double[id][id];
        for (int i = 1; i <= n; ++i) {
            int[] nextAry2i = nextAry2[i];
            int[] idi = ids[i];
            for (int j = 1; j <= m; ++j) {
                int cid = idi[j];
                if (idi[j] < 0) continue;

                int v = nextAry2i[j];

                if (v != EXIT) {

                    int a=v>>16,b=v&0xffff;

                    if(a!=i || b!=j) {

                        a = i;

                        b = j;

                    }

                    int w0 = nextAry2[a][b - 1], w1 = nextAry2[a - 1][b], w2 = nextAry2[a][b + 1],w3 = nextAry2[a + 1][b];

                    int c = (w0 > 0 ? 1 : 0) + (w1 > 0 ? 1 : 0) + (w2 > 0 ? 1 : 0) + (w3 > 0 ? 1 : 0);

                    if (c == 0) continue;

                    double c1 = 1.0 / c;

                    if(w0==EXIT) T[cid][ids[a][b-1]] = c1; else if(w0 > 1) T[cid][ids[w0 >> 16][w0 & 0xffff]] = c1;

                    if(w1==EXIT) T[cid][ids[a-1][b]] = c1; else if (w1 > 1) T[cid][ids[w1 >> 16][w1 & 0xffff]] = c1;

                    if(w2==EXIT) T[cid][ids[a][b+1]] = c1; else if (w2 > 1) T[cid][ids[w2 >> 16][w2 & 0xffff]] = c1;

                    if(w3==EXIT) T[cid][ids[a+1][b]] = c1; else if (w3 > 1) T[cid][ids[w3 >> 16][w3 & 0xffff]] = c1;

                    continue;

                }

                T[cid][cid] = 1.0;

            }

        }


        double[][] TP = pow(T, id, 0x10000L);

        int ida = ids[ax][ay];
        double rs = 0;
        for (int i = 1; i <= n; ++i)
            for (int j = 1; j <= m; ++j)
                if (nextAry2[i][j] == EXIT) rs += TP[ida][ids[i][j]];

        System.out.println(rs);
    }
    public static void print(double[][] x) {
        System.out.println("[");
        for(int i=0;i<x.length;++i) {
            if(i!=0) {
                System.out.print(",");
            }
            System.out.println(Arrays.toString(x[i]));
        }
        System.out.println("]");

        for (int i = 0; i < x.length; ++i) {
            if (i > 0) {
                System.out.println("\n");
            }
            for (int j = 0; j < x[i].length; ++j) {
                if (j > 0) {
                    System.out.print(' ');
                }
                System.out.print(String.format("%.20f", x[i][j]));
            }
        }
    System.out.println();
    System.out.println("----------------");
    System.out.println();
}

static void print(Object...args) {
    System.out.println(Arrays.toString(args));
}

static void mul(double[][] A, double[][] B, double[][] R, int n) {
    for (int i = 0,k=0; i < n; i++) {
        double[] Ri = R[i],Ai = A[i];
        for (int j = 0; j < n; j++)
            for (k =0, Ri[j]=0; k < n; k++) Ri[j] += Ai[k] * B[k][j];
    }
}
static double[][] pow(double[][] A, int n, long p) {
    double[][] C = new double[n][n],R = new double[n][n], t = null;
    for (int i = 0; i < n; i++) R[i][i] = 1;
    while (p != 0) {
        if (p % 2 == 1) {
            mul(A, R, C, n);
            t = C;
            C = R;
            R = t;
        }
        mul(A, A, C, n);
```

```
                    t = C;
                    C = A;
                    A = t;
                    p >>= 1;
                }
                return R;
            }
        }
```

2. OUTPUT:



Congratulations

You solved this challenge. Would you like to challenge your friends?

Compiler Message

```
Success
```

Input (stdin)

```
3 6 1
###*00
0#0A%0
###*00
2 3 2 1
```

Expected Output

```
0.25
```