**Name**: Tavishi Mahajan

**Class**: TY -CS   **Div**: D

**Roll no**:56

**Assignment 2:**  Implementation of Uninformed strategies-BFS,DFS

**Code**:

package ai\_lab;

import java.util.LinkedList;

import java.util.Queue;

import java.util.Scanner;

import java.util.Stack;

class graph {

    private int vertCount;

    private int edgeCount;

    private int[][] adjmatrix;

    public graph(int vertexCount) {

        edgeCount = 0;

        vertCount = vertexCount;

        adjmatrix = new int[vertCount][vertCount];

        for (int i = 0; i < vertCount; i++) {

            for (int j = 0; j < vertCount; j++)

                adjmatrix[i][j] = 0;

            }

        }

    public void accept(Scanner sc) {

        System.out.println("Enter the number of edges");

        edgeCount = sc.nextInt();

        for (int i = 0; i < edgeCount; i++) {

            System.out.println("enter edge (src dest) : ");

            int src = sc.nextInt();

            int dest = sc.nextInt();

            adjmatrix[src][dest] = 1;

            adjmatrix[dest][src] = 1;

        }

    }

    public void display() {

        System.out.println("\nAdjacency matrix :  \n");

        for (int i = 0; i < vertCount; i++) {

            for (int j = 0; j < vertCount; j++)

                System.out.print(adjmatrix[i][j] + "\t");

            System.out.println();

        }

    }

        public void bfsTraversal(int start){

            System.out.print("bfs traversal : ");

        boolean[] marked = new boolean[vertCount];

        Queue<Integer> q = new LinkedList<Integer>();

      q.offer(start);

        marked[start] = true;

        while(!q.isEmpty()){

            int trav = q.poll();

            System.out.print(trav +",");

            for (int dest = 0; dest <vertCount ; dest++) {

                if(adjmatrix[trav][dest]==1 && !marked[dest]){

                    q.offer(dest);

                    marked[dest]= true;

                }

            }

        }

            System.out.println();

        }

    public void dfsTraversal(int start){

        System.out.print("dfs traverssal : ");

        boolean[] marked = new boolean[vertCount];

        Stack<Integer> s = new Stack<Integer>();

        s.push(start);

        marked[start] = true;

        while(!s.isEmpty()){

            int trav = s.pop();

            System.out.print(trav +",");

            for (int dest = 0; dest <vertCount ; dest++) {

                if(adjmatrix[trav][dest]==1 && !marked[dest]){

                    s.push(dest);

                    marked[dest]= true;

                }

            }

        }

        System.out.println();

    }

    }

  public class Bfs\_dfs {

        public static void main(String[] args) {

            Scanner sc = new Scanner(System.in);

            System.out.println("enter the number of vertices :");

            int vertCount = sc.nextInt();

            graph g = new graph(vertCount);

            g.accept(sc);

            g.display();

            g.bfsTraversal(0);

            g.dfsTraversal(0);

        }

}

**Output**:

