import java.util.LinkedList;

import java.util.Queue;

import java.util.Scanner;

import java.util.Stack;

public class graph {

    LinkedList<Integer> adjacency[];

    graph(int v) {

        adjacency = new LinkedList[v];

        for (int i = 0; i < v; i++) {

            adjacency[i] = new LinkedList<Integer>();

        }

    }

    void path(int s, int d) {

        adjacency[s].add(d);

        adjacency[d].add(s);

    }

    void bfs(int nod) {

        boolean visit[] = new boolean[adjacency.length];

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

        q.add(nod);

        visit[nod] = true;

        while (!q.isEmpty()) {

            int p = q.poll();

            System.out.print(p+" ");

            for (int i : adjacency[p]) {

                if (visit[i] != true) {

                    q.add(i);

                    visit[i] = true;

                }

            }

        }

    }

    void dfs(int nod) {

        boolean visit[] = new boolean[adjacency.length];

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

        q.add(nod);

        visit[nod] = true;

        while (!q.isEmpty()) {

            int p = q.pop();

            System.out.print(p+" ");

            for (int i : adjacency[p]) {

                if (visit[i] != true) {

                    q.add(i);

                    visit[i] = true;

                }

            }

        }

    }

    public static void main(String[] args) {

        Scanner s = new Scanner(System.in);

        int e, v;

        System.out.print("Enter Edges : ");

        e = s.nextInt();

        System.out.print("Enter Vertex : ");

        v = s.nextInt();

        graph g = new graph(v);

        for (int i = 0; i < e; i++) {

            System.out.print("enter Source : ");

            int source = s.nextInt();

            System.out.print("enter Destination : ");

            int destination = s.nextInt();

            g.path(source, destination);

        }

        System.out.println("Enter Source:  ");

        int sor = s.nextInt();

        System.out.println("BFS IS : ");

        g.bfs(sor);

        System.out.println("DFS IS : ");

        g.dfs(sor);

    }

}