#include <stdio.h>

#include <stdlib.h>

#define MAX 100

int adjMatrix[MAX][MAX];

int visited[MAX];

int n; // Number of cities

void createGraph() {

int i, j, edges, origin, destin;

printf("Enter number of cities: ");

scanf("%d", &n);

for(i = 0; i < n; i++) {

for(j = 0; j < n; j++) {

adjMatrix[i][j] = 0;

}

}

printf("Enter number of directed edges: ");

scanf("%d", &edges);

for(i = 0; i < edges; i++) {

printf("Enter origin and destination of edge %d: ", i + 1);

scanf("%d %d", &origin, &destin);

if(origin >= n || destin >= n || origin < 0 || destin < 0) {

printf("Invalid edge!\n");

i--;

} else {

adjMatrix[origin][destin] = 1;

}

}

}

void DFS(int vertex) {

int i;

printf("%d ", vertex);

visited[vertex] = 1;

for(i = 0; i < n; i++) {

if(adjMatrix[vertex][i] && !visited[i]) {

DFS(i);

}

}

}

void BFS(int startVertex) {

int queue[MAX], front = -1, rear = -1;

int i, currentVertex;

printf("%d ", startVertex);

visited[startVertex] = 1;

queue[++rear] = startVertex;

while(front != rear) {

currentVertex = queue[++front];

for(i = 0; i < n; i++) {

if(adjMatrix[currentVertex][i] && !visited[i]) {

printf("%d ", i);

queue[++rear] = i;

visited[i] = 1;

}

}

}

}

void resetVisited() {

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

visited[i] = 0;

}

}

int main() {

int startVertex;

createGraph();

printf("Enter the starting node for DFS: ");

scanf("%d", &startVertex);

printf("Nodes reachable from %d using DFS: ", startVertex);

resetVisited();

DFS(startVertex);

printf("\n");

printf("Enter the starting node for BFS: ");

scanf("%d", &startVertex);

printf("Nodes reachable from %d using BFS: ", startVertex);

resetVisited();

BFS(startVertex);

printf("\n");

return 0;

}