#include <stdio.h>

#include <stdlib.h>

#define MAX\_VERTICES 100

struct Node {

int data;

struct Node\* next;

};

struct Graph {

struct Node\* adjList[MAX\_VERTICES];

int visited[MAX\_VERTICES];

};

void initializeGraph(struct Graph\* graph, int numVertices) {

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

graph->adjList[i] = NULL;

graph->visited[i] = 0;

}

}

void addEdge(struct Graph\* graph, int src, int dest) {

struct Node\* newNode = (struct Node\*)malloc(sizeof(struct Node));

newNode->data = dest;

newNode->next = graph->adjList[src];

graph->adjList[src] = newNode;

}

void dfs(struct Graph\* graph, int vertex) {

graph->visited[vertex] = 1;

printf("%d ", vertex);

struct Node\* temp = graph->adjList[vertex];

while (temp != NULL) {

int adjVertex = temp->data;

if (graph->visited[adjVertex] == 0) {

dfs(graph, adjVertex);

}

temp = temp->next;

}

}

int main() {

struct Graph graph;

int numVertices, numEdges;

printf("Enter the number of vertices: ");

scanf("%d", &numVertices);

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

scanf("%d", &numEdges);

initializeGraph(&graph, numVertices);

printf("Enter edges (source destination):\n");

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

int src, dest;

scanf("%d %d", &src, &dest);

addEdge(&graph, src, dest);

addEdge(&graph, dest, src); // For undirected graph

}

int startVertex;

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

scanf("%d", &startVertex);

printf("DFS traversal starting from vertex %d: ", startVertex);

dfs(&graph, startVertex);

printf("\n");

return 0;

}

Output:

