#include <iostream>

#include <vector>

#include <stack>

using namespace std;

void DFS(int start, vector<vector<int>>& graph, vector<bool>& visited) {

stack<int> stack;

stack.push(start);

visited[start] = true;

while (!stack.empty()) {

int current = stack.top();

stack.pop();

cout << current << " ";

for (int i = 0; i < graph[current].size(); ++i) {

int neighbor = graph[current][i];

if (!visited[neighbor]) {

stack.push(neighbor);

visited[neighbor] = true;

}

}

}

}

int main() {

int vertices, edges;

cout << "Enter the number of vertices and edges: ";

cin >> vertices >> edges;

vector<vector<int>> graph(vertices);

vector<bool> visited(vertices, false);

cout << "Enter the edges (source destination):" << endl;

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

int u, v;

cin >> u >> v;

graph[u].push\_back(v);

graph[v].push\_back(u);

}

int startNode;

cout << "Enter the starting node for DFS: ";

cin >> startNode;

cout << "DFS Traversal starting from node " << startNode << ": ";

DFS(startNode, graph, visited);

cout << endl;

return 0;

}

