19)

#include <iostream>

#include <vector>

using namespace std;

bool hasCycleUtil(const vector<vector<int>>& graph, vector<bool>& visited, vector<bool>& recursionStack, int vertex) {

visited[vertex] = true;

recursionStack[vertex] = true;

for (int neighbor : graph[vertex]) {

if (!visited[neighbor] && hasCycleUtil(graph, visited, recursionStack, neighbor)) {

return true;

} else if (recursionStack[neighbor]) {

return true;

}

}

recursionStack[vertex] = false;

return false;

}

bool hasCycle(const vector<vector<int>>& graph, int vertices) {

vector<bool> visited(vertices, false);

vector<bool> recursionStack(vertices, false);

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

if (!visited[i] && hasCycleUtil(graph, visited, recursionStack, i)) {

return true;

}

}

return false;

}

int main() {

int vertices, edges;

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

cin >> vertices;

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

cin >> edges;

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

cout << "Enter the edges (from to):" << endl;

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

int from, to;

cin >> from >> to;

graph[from].push\_back(to);

}

if (hasCycle(graph, vertices)) {

cout << "The graph contains a cycle." << endl;

} else {

cout << "The graph does not contain a cycle." << endl;

}

return 0;

}