**\*\*\*AI DFS BFS \*\*\***

#include <iostream>

#include <vector>

#include <queue>

using namespace std;

// DFS

void dfs(vector<vector<int>> &adj, int node, vector<int> &vis) {

if (adj[node].size() == 0) {

return;

}

for (auto it : adj[node]) {

if (vis[it] == 0) {

vis[it] = 1;

cout << it << " ";

dfs(adj, it, vis);

}

}

}

// BFS

void bfs(vector<vector<int>> &adj, vector<int> &vis) {

queue<int> q;

q.push(0);

while (!q.empty()) {

int front = q.front();

q.pop();

cout << front << " ";

for (auto it : adj[front]) {

if (vis[it] == 0) {

vis[it] = 1;

q.push(it);

}

}

}

}

int main() {

int n;

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

cin >> n;

vector<vector<int>> adj(n);

cout << "Enter adjacency list:" << endl;

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

cout << "Enter number of neighbors for node " << i << ": ";

int num\_neighbors;

cin >> num\_neighbors;

cout << "Enter neighbors for node " << i << ": ";

for (int j = 0; j < num\_neighbors; ++j) {

int neighbor;

cin >> neighbor;

adj[i].push\_back(neighbor);

}

}

vector<int> vis(n, 0);

cout << "BFS: ";

bfs(adj, vis);

cout << endl;

vector<int> vis2(n, 0);

cout << "DFS: ";

cout << "0"

<< " ";

dfs(adj, 0, vis2);

return 0;

}