Lab #12

Task# 1:

#include <iostream>

using namespace std;

class UndirecGraph

{

private:

int N;

int M;

int\*\* arr; // 2D array for the adjacency matrix

public:

// Constructor to initialize

UndirecGraph(int n, int m) : N(n), M(m)

{

arr = new int\*[N];

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

arr[i] = new int[N];

for (int j = 0; j < N; j++) {

arr[i][j] = 0; // Initialize all cells to 0

}

}

}

void addEdge(int x, int y)

{

if (x > 0 && x <= N && y > 0 && y <= N)

{

arr[x - 1][y - 1] = 1; //updating rows and columns

arr[y - 1][x - 1] = 1;

}

}

void printAdjacencyMatrix()

{

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

for (int j = 0; j < N; j++)

{

cout << arr[i][j] << " ";

}

cout <<endl;

}

}

};

int main() {

int N = 8; // Number of vertices

int M = 7; // Number of edges

int edges[7][2] = { {1, 2}, {2, 3}, {4, 5}, {1, 5}, {6, 1}, {7, 4}, {3, 8} };

UndirecGraph graph(N, M);

// Add edges to the graph

for (int i = 0; i < M; i++)

{

int X = edges[i][0];

int Y = edges[i][1];

graph.addEdge(X, Y);

}

cout << "Adjacency Matrix:" <<endl;

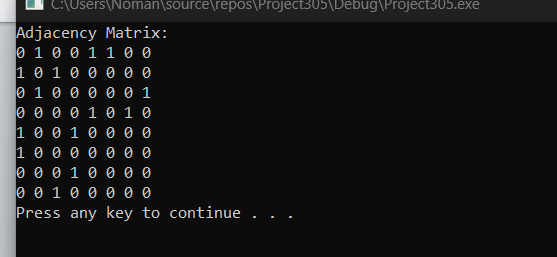
graph.printAdjacencyMatrix();

system("pause");

return 0;

}

Output:



Task#2:

#include <iostream>

using namespace std;

class UndirectedGraph

{

private:

int N;

int\*\* arr;

int\* edgeCount; // Array to store the number of edges for each node

public:

UndirectedGraph(int n) : N(n) //using intializer

{

arr = new int\*[N];

for (int i = 0; i < N; i++)

{

arr[i] = new int[N];

for (int j = 0; j < N; j++) {

arr[i][j] = 0;

}

}

edgeCount = new int[N];

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

edgeCount[i] = 0;

}

}

void addEdge(int x, int y)

{

if (x >= 0 && x < N && y >= 0 && y < N)

{

arr[x][y] = 1;

arr[y][x] = 1;

edgeCount[x]++;

edgeCount[y]++;

}

}

void printAdjacencyMatrix() {

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

for (int j = 0; j < N; j++)

{

cout << arr[i][j] << " ";

}

cout << endl;

}

}

void printEdgeCount()// displaying the edges of each node

{

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

cout << "Node " << i << " has " << edgeCount[i] << " edges." << endl;

}

}

};

int main()

{

int N = 10; // Number of vertices

UndirectedGraph graph(N);

int edges[][2] = { {0, 1}, {0, 6}, {0, 8}, {1, 4}, {1, 6}, {1, 9},

{2, 4}, {2, 6}, {3, 4}, {3, 5}, {3, 8}, {4, 5},

{4, 9}, {7, 8}, {7, 9} };

int M = sizeof(edges) / sizeof(edges[0]);

// Add edges to the graph

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

int X = edges[i][0];

int Y = edges[i][1];

graph.addEdge(X, Y);

}

cout << "Adjacency Matrix:" << endl;

graph.printAdjacencyMatrix();

cout << "\nEdge Counts:" << endl;

graph.printEdgeCount();

system("pause");

return 0;

}

Output:

