**Problem No: 01**

**Topic: Graph**

**Problem Title:**

Given the vertex pairs associated to the edges of an undirected graph, find the degree of each vertex.

**Objectives:**

To find the degree of each vertex of an undirected graph.

**Source Code:**

#include <iostream>

#include <vector>

using namespace std;

void createGraph(vector <int> graph[], int n, int e);

void addEdge(vector <int> graph[], int u, int v);

void printGraph(vector <int> graph[], int n);

int main()

{

int n, e;

cout << "Enter no. of vertex / node: ";

cin >> n;

cout << "Enter no. of edges: ";

cin >> e;

cout << endl;

vector <int> graph[n];

createGraph(graph, n, e);

printGraph(graph, n);

return 0;

}

void createGraph(vector <int> graph[], int n, int e)

{

int u, v;

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

cout << "Enter the start and end vertex of edge: ";

cin >> u;

cin >> v;

addEdge(graph, u, v);

}

cout << endl;

}

void addEdge(vector<int> graph[], int u, int v)

{

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

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

}

void printGraph(vector <int> graph[], int n)

{

cout << "Undirected Graph:\n";

vector <int> :: iterator it;

int deg;

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

cout << "Adjacency list of vertex " << i << ": ";

for(it = graph[i]. begin(), deg = 0; it != graph[i]. end(); it++, deg++)

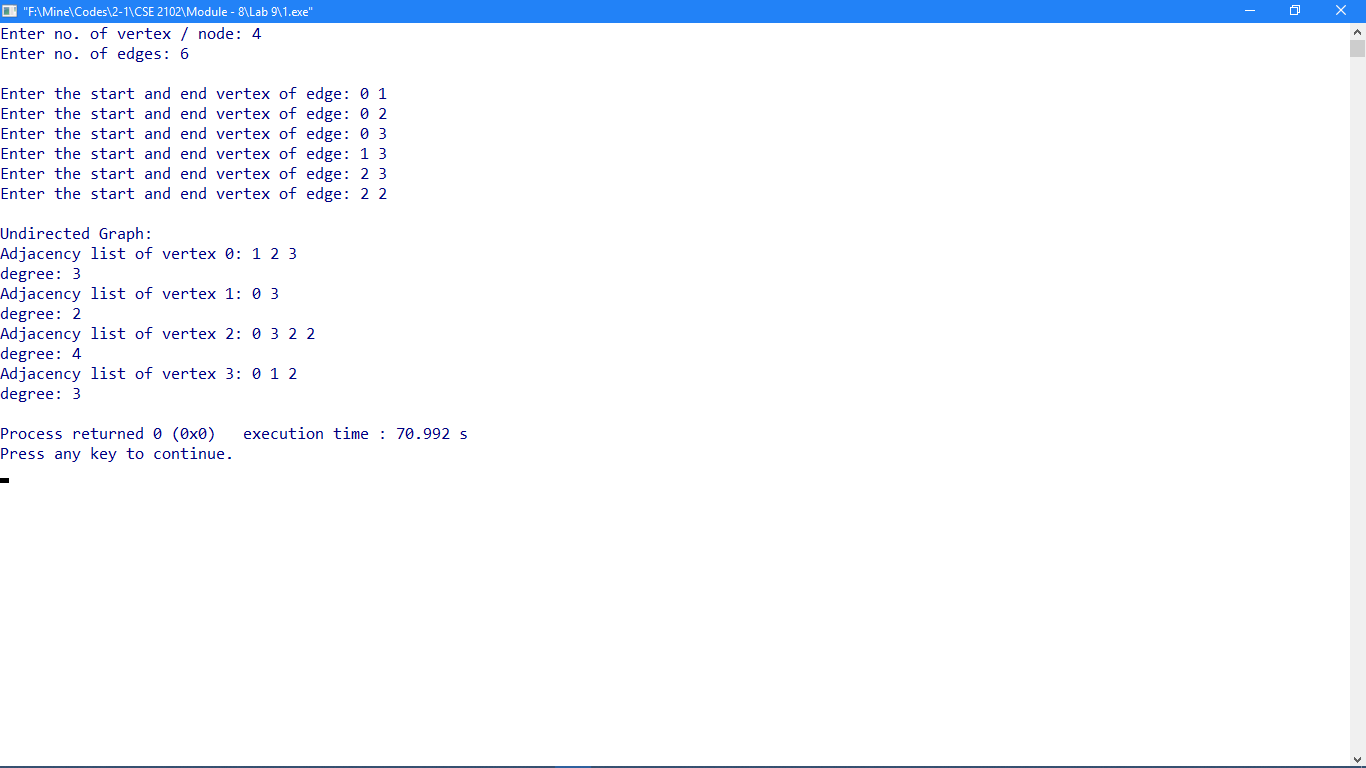
cout << \*it << " ";

cout << "\ndegree: " << deg << endl;

}

}

**Output:**



**Problem No: 02**

**Topic: Graph**

**Problem Title:**

Given the ordered pairs of vertices associated to the edges of a directed graph, determine the in-degree and out-degree of each vertex.

**Objectives:**

To find the in-degree and out-degree of each vertex of a directed graph.

**Source Code:**

#include <iostream>

#include <vector>

using namespace std;

void createGraph(vector <int> graph[], int n, int e, int \*in);

void addEdge(vector <int> graph[], int u, int v);

void printGraph(vector <int> graph[], int n, int \*in);

int main()

{

int n, e;

cout << "Enter no. of vertex / node: ";

cin >> n;

cout << "Enter no. of edges: ";

cin >> e;

cout << endl;

vector <int> graph[n];

int inDegree[n] = {0};

createGraph(graph, n, e, inDegree);

printGraph(graph, n, inDegree);

return 0;

}

void createGraph(vector <int> graph[], int n, int e, int \*in)

{

int u, v;

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

cout << "Enter the start and end vertex of edge: ";

cin >> u;

cin >> v;

addEdge(graph, u, v);

in[v]++;

}

cout << endl;

}

void addEdge(vector<int> graph[], int u, int v)

{

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

}

void printGraph(vector <int> graph[], int n, int \*in)

{

cout << "Directed Graph:\n";

vector <int> :: iterator it;

int deg;

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

cout << "Adjacency list of vertex " << i << ": ";

for(it = graph[i]. begin(), deg = 0; it != graph[i]. end(); it++, deg++){

cout << \*it << " ";

}

cout << "\nin-degree: " << in[i];

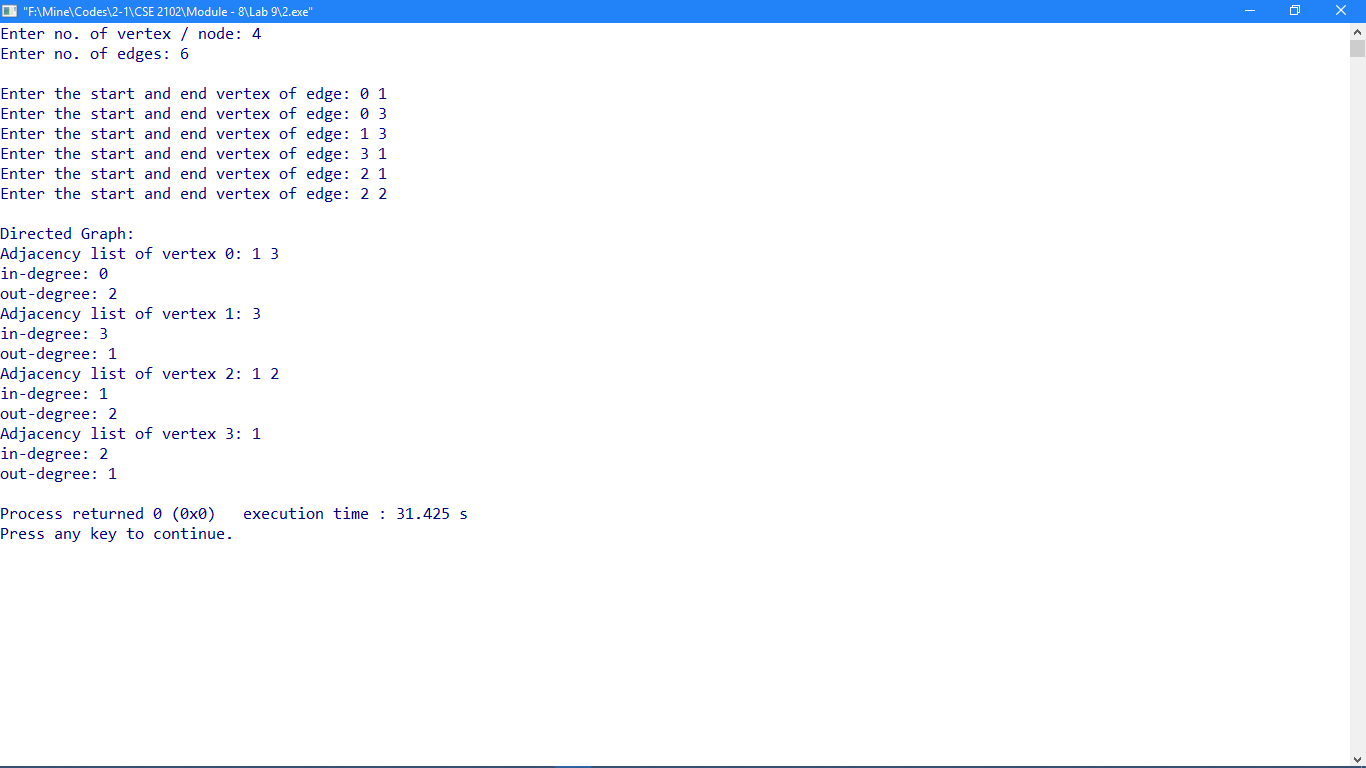
cout << "\nout-degree: " << deg;

cout << endl;

}

}

**Output:**



**Problem No: 03**

**Topic: Graph**

**Problem Title:**

Given the vertex pairs associated to the edges of an undirected graph, construct an adjacency matrix for the graph.

**Objectives:**

To construct an adjacency matrix of an undirected graph.

**Source Code:**

#include <iostream>

#include <vector>

using namespace std;

void addEdge(vector<int> graph[], int u, int v);

int main()

{

int n, e;

cout << "Enter no. of vertex / node: ";

cin >> n;

cout << "Enter no. of edges: ";

cin >> e;

cout << endl;

int m[n][n] = {};

vector <int> graph[n];

int u, v;

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

cout << "Enter the start and end vertex of edge: ";

cin >> u;

cin >> v;

addEdge(graph, u, v);

m[u][v]++;

m[v][u]++;

}

cout << endl;

cout << "Adjacency Matrix for Undirected Graph: \n";

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

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

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

cout << endl;

}

return 0;

}

void addEdge(vector<int> graph[], int u, int v)

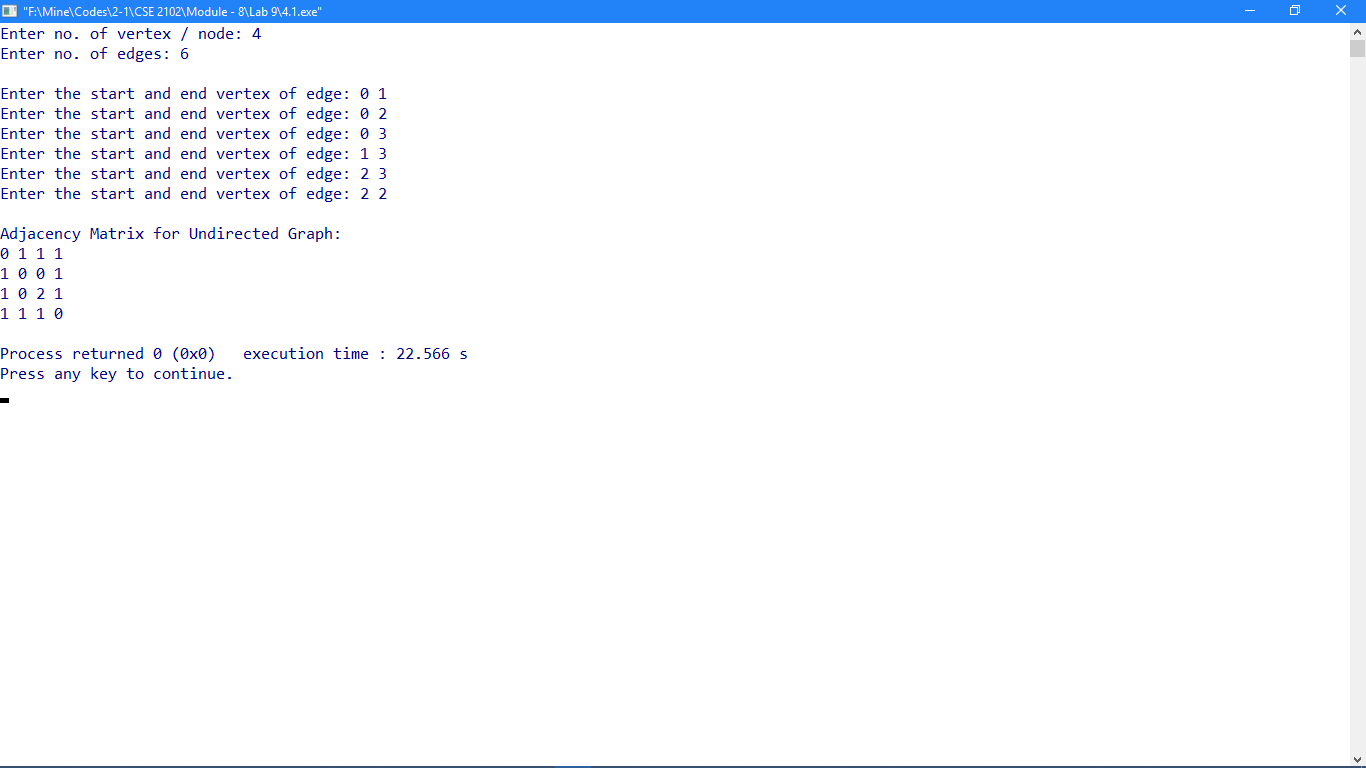
{

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

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

}

**Output:**



**Problem No: 04**

**Topic: Graph**

**Problem Title:**

Given the vertex pairs associated to the edges of a directed graph, construct an adjacency matrix for the graph.

**Objectives:**

To construct an adjacency matrix of a directed graph.

**Source Code:**

#include <iostream>

#include <vector>

using namespace std;

void addEdge(vector<int> graph[], int u, int v);

int main()

{

int n, e;

cout << "Enter no. of vertex / node: ";

cin >> n;

cout << "Enter no. of edges: ";

cin >> e;

cout << endl;

int m[n][n] = {};

vector <int> graph[n];

int u, v;

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

cout << "Enter the start and end vertex of edge: ";

cin >> u;

cin >> v;

addEdge(graph, u, v);

m[u][v]++;

}

cout << endl;

cout << "Adjacency Matrix for Directed Graph: \n";

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

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

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

cout << endl;

}

return 0;

}

void addEdge(vector<int> graph[], int u, int v)

{

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

}

**Output:**

