Adjacency Matrix

# Source Code:

/\*

Name : Robin Singh

Rollno : 1263

Unit : 07

Program : Adjacency Matrix

\*/

#include<iostream> #include<conio.h>

#define MAX 10 using namespace std;

// Graph Template class Graph

{

int adj[MAX][MAX]; int nodes;

int edges;

public:

Graph()

{

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

{

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

{

adj[i][j] = 0;

}//end of j

}//end of i nodes = 0;

edges = 0;

}//end of graph

};

//Function

void CreateGraph(); void DisplayGraph();

void Graph::CreateGraph()

{

int i , origin , destination;

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

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

for(i=1 ; i<=edges ; i++)

{

cout << "Enter the edge: "<< i << endl; cout << "Enter the soucre node: ";

cin >> origin;

cout << "Enter the destination: "; cin >> destination;

adj[origin][destination] = 1;

adj[destination][origin] = 1;

}

}

void Graph::DisplayGraph()

{

int i,j;

for(i=1 ; i<=nodes ; i++)

{

for(j=1 ; j<=nodes ; j++)

{

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

}

cout << endl;

}

}

// Menu int main()

{

int ch; Graph g;

while(1)

{

system("cls");

cout << "\*\*\*Adjacency Matrix\*\*\*" << endl << endl; cout << "1. Create Graph\n";

cout << "2. Display Graph\n"; cout << "3. Exit\n" <<endl ;

cout << "Enter your choice: "; cin >> ch;

switch(ch)

{

case 1:

case 2:

case 3:

g.CreateGraph(); getch();

break;

g.DisplayGraph(); getch();

break;

exit(1);

default:

cout << "Enter a valid choice!"; getch();

break;

}

}

}

# Output:

