Adjacency Matrix

Source Code:

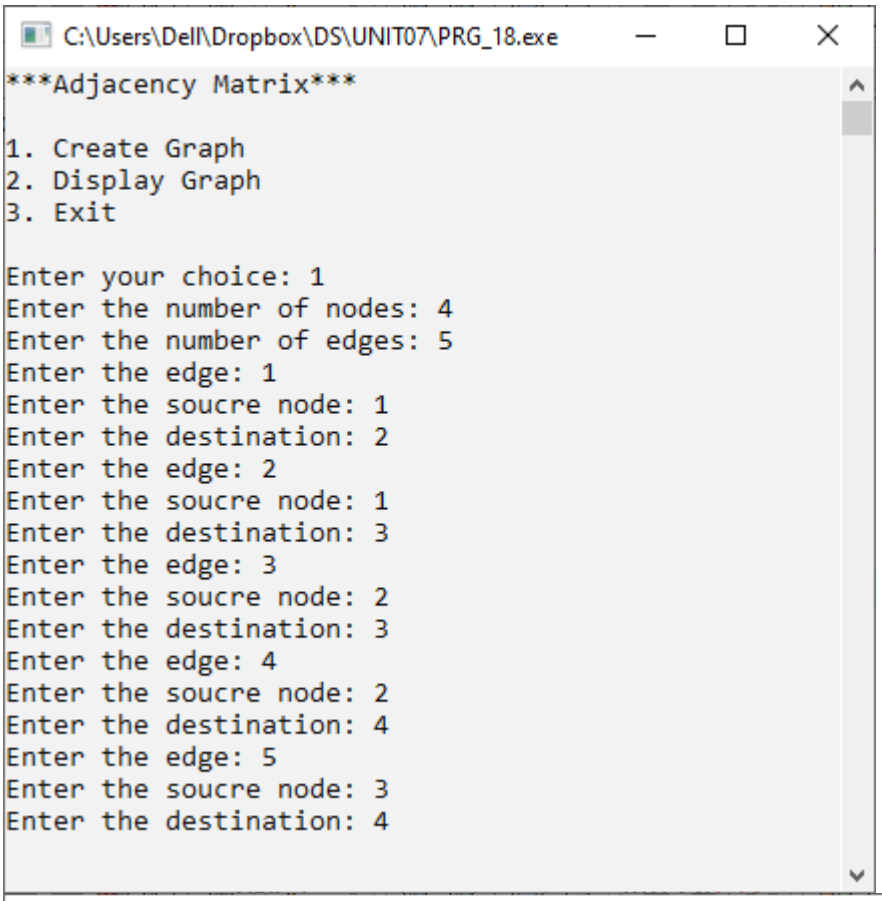
```
/*
        Name : Robin Singh
        Rollno : 1263
        Unit : 07
        Program : Adjacency Matrix
*/

#include<iostrea
m>
#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

};
```

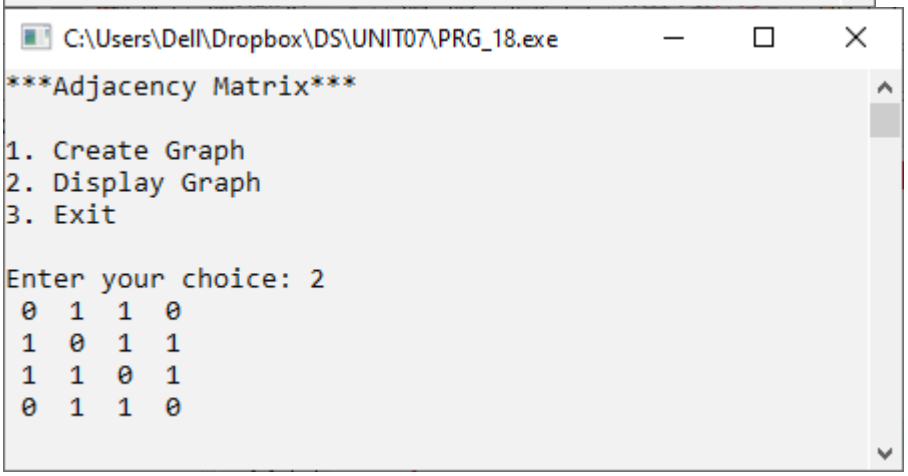
```
v           d CreateGraph();
o           void
i           DisplayGraph();
void Graph::CreateGraph()
{
        int i , origin , destination;
```

```
        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:
                        g.CreateGraph
                        (); getch();
                        break;
                case 2:
                        g.DisplayGraph
                        (); getch();
                        break;
                case 3:
                        exit(1);
                 default:
                        cout << "Enter a valid
                        choice!"; getch();
                        break;
            }

        }
}
```

Output:


C:\Users\Dell\Dropbox\DS\UNIT07\PRG_18.exe
Adjacency Matrix
1. Create Graph
2. Display Graph
3. Exit
Enter your choice: 1
Enter the number of nodes: 4
Enter the number of edges: 5
Enter the edge: 1
Enter the soucre node: 1
Enter the destination: 2
Enter the edge: 2
Enter the soucre node: 1
Enter the destination: 3
Enter the edge: 3
Enter the soucre node: 2
Enter the destination: 3
Enter the edge: 4
Enter the soucre node: 2
Enter the destination: 4
Enter the edge: 5
Enter the soucre node: 3
Enter the destination: 4


C:\Users\Dell\Dropbox\DS\UNIT07\PRG_18.exe
Adjacency Matrix
1. Create Graph
2. Display Graph
3. Exit
Enter your choice: 2
0 1 1 0
1 0 1 1
1 1 0 1
0 1 1 0