



THE UNIVERSITY
OF LAHORE
**ISLAMABAD
CAMPUS**

Data Structures and Algorithms (CS09203)

Lab Report

Name: M.Usman Ali
Registration #: SEU-F16-135
Lab Report #: 07
Dated: 21-05-2018
Submitted To: Sir. Usman Ahmed

The University of Lahore, Islamabad Campus
Department of Computer Science & Information Technology

Experiment # 7

Double Link list-Basic Insertion

Objective

The objective of this session is to understand graph and adjacent matrix in C++. using C++..

Software Tool

1. I use Code Blocks with GCC compiler.

1 Theory

In graph theory and computer science, an adjacency matrix is a square matrix used to represent a finite graph. The elements of the matrix indicate whether pairs of vertices are adjacent or not in the graph. In the special case of a finite simple graph, the adjacency matrix is a $(0,1)$ -matrix with zeros on its diagonal. If the graph is undirected, the adjacency matrix is symmetric. The relationship between a graph and the eigenvalues and eigenvectors of its adjacency matrix is studied in spectral graph theory. The adjacency matrix should be distinguished from the incidence matrix for a graph, a different matrix representation whose elements indicate whether vertex–edge pairs are incident or not, and degree matrix which contains information about the degree of each vertex.

2 Task

2.1 Procedure: Task 7

Write a C++ code using functions for: 1.Add edges in adjacent matrix.
2.Display the added edges.

2.2

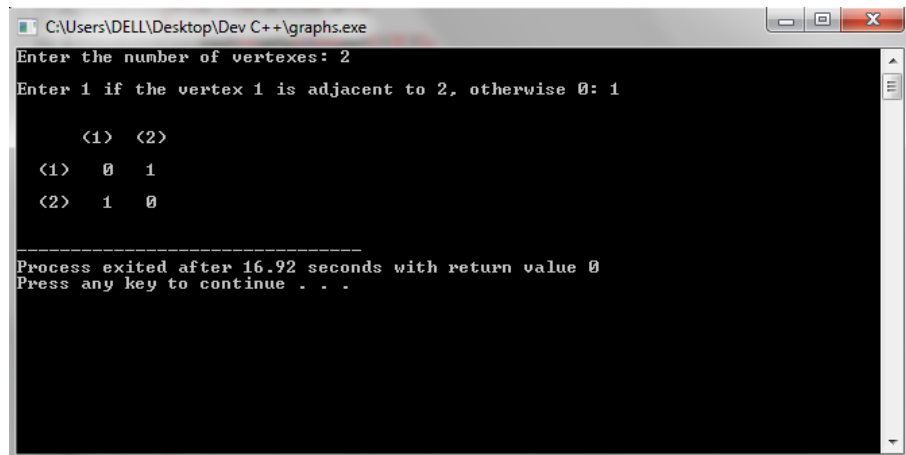


Figure 1: output

```
#include<iostream>
#include<iomanip>

using namespace std;

// A function to print the adjacency matrix.
void PrintMat(int mat[][20], int n)
{
    int i, j;

    cout<<"\n\n"<<setw(4)<<" ";
    for(i = 0; i < n; i++)
        cout<<setw(3)<<" ("<<i+1<<" )";
    cout<<"\n\n";

    // Print 1 if the corresponding vertexes are connected otherwise 0
    for(i = 0; i < n; i++)
    {
        cout<<setw(3)<<" ("<<i+1<<" )";
        for(j = 0; j < n; j++)
        {
            cout<<setw(4)<<mat[i][j];
        }
        cout<<"\n\n";
    }
}
```

```

    }
}

int main()
{
    int i, j, v;

    cout<<"Enter the number of vertexes: ";
    cin>>v;

    int mat[20][20];

    cout<<"\n";
    // Take input of the adjacency of each pair of vertexes.
    for(i = 0; i < v; i++)
    {
        for(j = i; j < v; j++)
        {
            if(i != j)
            {
                cout<<"Enter 1 if the vertex "<<i+1<<" is
                cin>>mat[i][j];

                mat[j][i] = mat[i][j];
            }
            else
                mat[i][j] = 0;
        }
    }

    PrintMat(mat, v);
}

```

3 Conclusion

In today lab we have discussed how we can create edges in adjacent matrix and display it on a screen .