Data Structures and Algorithms
(CS09203)

Lab Report

Name:	Farhan Naseer
Registration #:	SEU-F16-125
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
How to create a simple Graph in C++

Objective
The objective of this session is to create the graph and tell the number of edges and vertices using C++.

Software Tool
1. I use Code Blocks with GCC compiler.

1 Theory

This section discusses how to create the graph and tell the number of edges and vertices . Graphs are used to model electrical circuits, chemical compounds, highway maps, and so on. They are also used in the analysis of electrical circuits, finding the shortest route, project planning, linguistics, genetics, social science, and so forth Graph is a non linear data structure, it contains a set of points known as nodes (or vertices) and set of lines known as edges (or Arcs) which connects the vertices. A graph is defined as follows Graph is a collection of nodes and edges which connects nodes in the graph Generally, a graph G is represented as G = (V , E), where V is set of vertices and E is set of edges.

2 Task

2.1 Procedure: Task 5

Write a C++ code using functions for the following operations. 1.Creating Graph using edges and vertices.

2.2

```
#include <iostream>
```

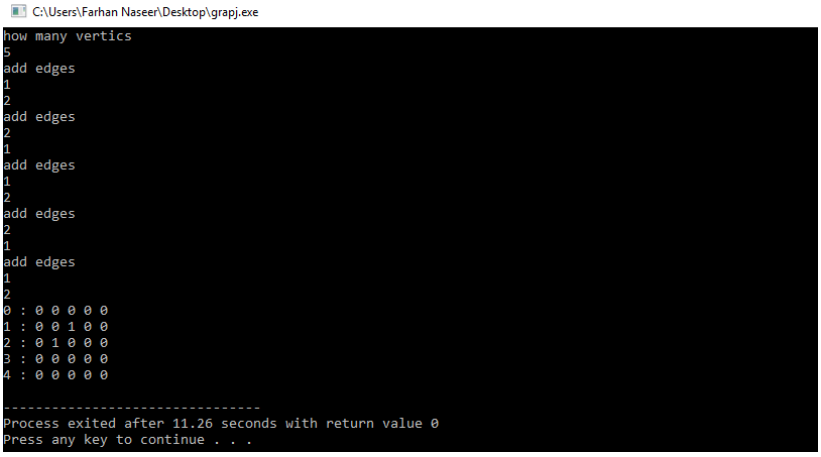

Figure 1: output

```
using namespace std;
class Graph {
private:
      bool** adjMatrix;
      int numVertices;
public:
      Graph(int numVertices) {
            this->numVertices = numVertices;
            adjMatrix = new bool*[numVertices];
            for (int i = 0; i < numVertices; i++) {
                  adjMatrix[i] = new bool[numVertices];
                  for (int j = 0; j < numVertices; j++)
                        adjMatrix[i][j] = false;
            }
    }

      void addEdge(int i, int j) {
                  adjMatrix[i][j] = true;
                  adjMatrix[j][i] = true;
    }

      void removeEdge(int i, int j) {
                  adjMatrix[i][j] = false;
                  adjMatrix[j][i] = false;
```

```
    }

      bool isEdge(int i, int j) {
                return adjMatrix[i][j];
    }

    void toString() {
      for (int i = 0; i < numVertices; i++) {
                cout << i << " : ";
                for (int j = 0; j < numVertices; j++)
                      cout << adjMatrix[i][j] << " ";
                cout << "\n";
      }

    }

    ~Graph() {
            for (int i = 0; i < numVertices; i++)
                  delete[] adjMatrix[i];
            delete[] adjMatrix;
    }
};


int main(){

 int a,b,c=0,y=0;
       cout<<"how many vertics"<<endl;
        cin>>c;
        Graph g(c);
           while(y!=c){

                     cout<<"add edges "<<endl;
        cin>>a>>b;
        g.addEdge(a,b);

        y++;

           }
        g.toString();
```

}

3 Conclusion

In today lab we have discussed how we can create a Graph using vertices and edges and how to display it on a screen by having a code.