**Kathmandu University**

**Department of Computer Science and Engineering**

**Dhulikhel, Karve**

****

**COMP 202**

Mini Project

**Submitted By:**

Sampurna KC

Roll No.-24

CE, 2nd Year, 1st Sem

**Submitted to:**

Rajani Chulyadyo

Department of Computer Science and Engineering

**Implementing graph data structure and using it to create a network of friends. Using graph traversal methods to find friends of friends in this network.**

A Graph is a non-linear data structure consisting of nodes and edges. The nodes are sometimes also referred to as vertices and the edges are lines or arcs that connect any two nodes in the graph.

In the mini project, STL of the C++ has been used for simplification. Queue and list from STL have been used to implement the graph data structure. In the project, it is assumed that each person will be recognized by an integer instead of strings and will have used the following operations.

1. Graph (int V) as a constructor
2. addEdge (int x, int y) for creating connections in between any two persons in the graph.
3. areFriends (int person1, int person2) for finding out if there is any connection/friendship among any two persons in the graph.
4. printFriendsofAllMembers () for printing out the friends of each person in the graph individually.
5. viewAllMembers () for traversing through the persons and displaying them without their friends lists in the graph, here the used traversing method is BFS.
6. friendsOfFriend (int source, int friendOfSource) for finding out the friends of a source.

The first two operations are the constructor and making a connection between any two members of the network(graph).

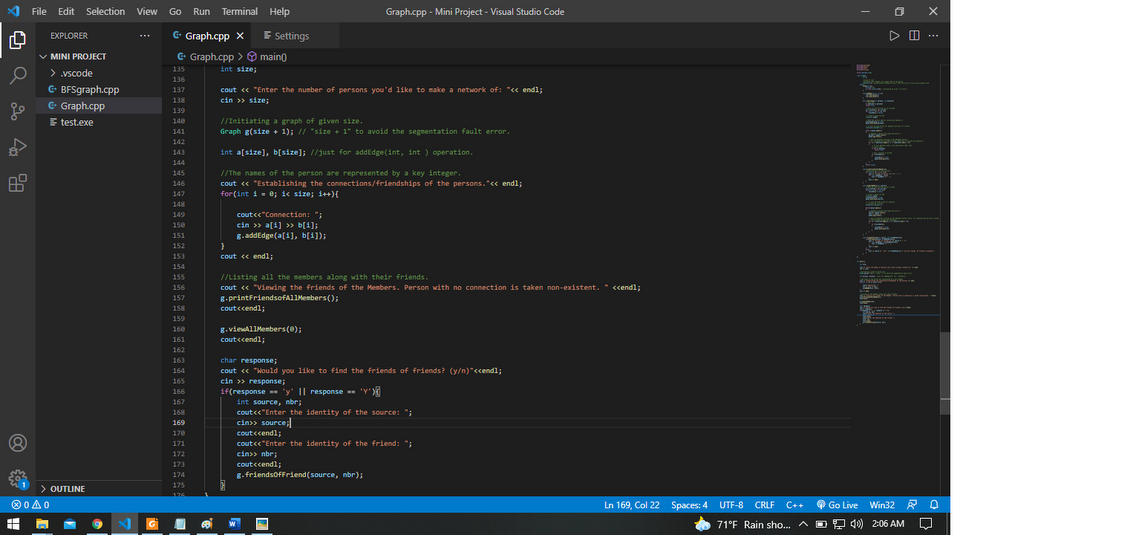
areFriends (int person1, int person2) checks if there is any connection to the given two persons by traversing through all the persons in the graph. If there exist no such connections, it returns false, else true.

printFriendsofAllMembers () is a nested for loop for printing all the persons and their friends.

viewAllMembers () is a BFS traversal to print only the members of the network. It was more logical to use BFS here instead of DFS just because the traversing technique would clear an entire level once it started. This would be more feasible to carry the operations like finding out the friends of friends in a graph which requires level by level traversing in my opinion.

friendsOfFriend (int source, int friendOfSource) is basically an operation which checks if there is a connection/ friendship between two persons stored in “source” and “friendOfSource” variables. If they are connected that is, if they are friends, then we print out the friends of “friendOfSource”. Thus, to find out the easier way to find the friends of friend, the printFriendsofAllMembers () operation helps us to find the related friends of a person in the first place.

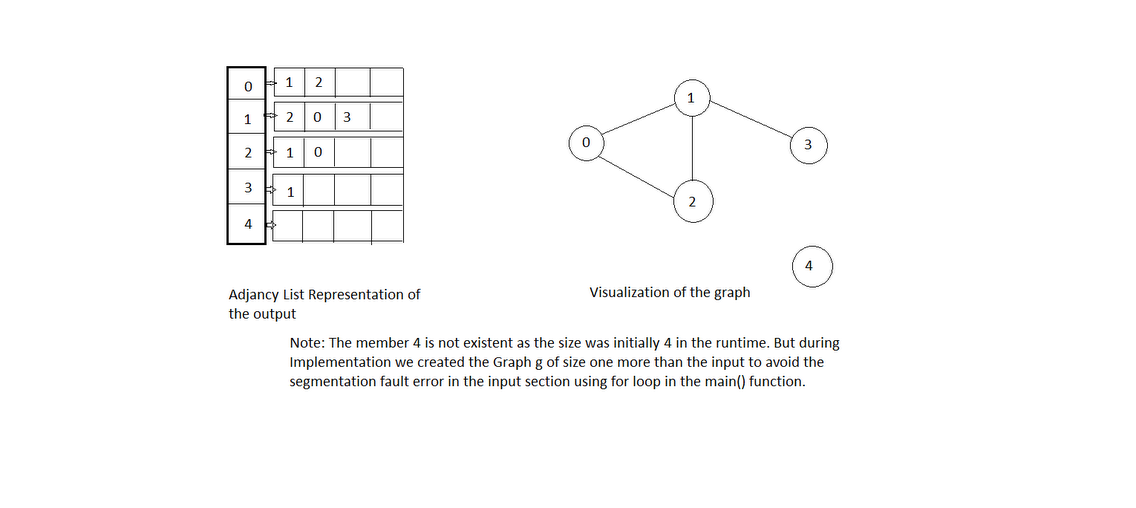
The Implementation of the project is:



The output of the project looks like:

### 

Graph representation can be shown as:



What can be done to improve further?

1. Generic Implementation of the graph can be used.
2. The graphical output would be easier to understand than the terminal view.