**Practical GroupC\_14**

**Aim:**

There are flight paths between cities. If there is a flight between city A and city B then there is an edge between the cities. The cost of the edge can be the time that flight take to reach city B from A, or the amount of fuel used for the journey. Represent this as a graph. The node can be represented by airport name or name of the city. Use adjacency list representation of the graph or use adjacency matrix representation of the graph. Check whether the graph is connected or not. Justify the storage representation used.

**Code:**

#include <iostream>

#include <cstring>

using namespace std;

void adjm1(string name[],int v)

{ int i,j;

char ch;

int g[v][v];

memset(g,0,sizeof(g)); //to set matrix to 0

for(i=0;i<v;i++)

{

for(j=i+1;j<v;j++)

{

if(g[i][j]==0)

{

cout<<"Is there a flight present from "<<name[i]<<" to "<<name[j]<<"?(Y/N)\n:";

cin>>ch;

if(ch=='Y'||ch=='y')

{

cout<<"Enter the fuel required for flight from "<<name[i]<<" to "<<name[j]<<"?(Y/N)\n:";

cin>>g[i][j];

g[j][i]=g[i][j];

}

}

}

}

cout<<"Adjacency matrix:\n ";

for(i=0;i<v;i++)

{

cout<<" "<<name[i]<<" ";

}

cout<<endl;

for(i=0;i<v;i++)

{

cout<<name[i];

for(j=0;j<v;j++)

{

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

}

cout<<endl;

}

}

int main()

{

int v,i,j,ch;

cout<<"Enter number of cities: \n";

cin>>v;

string name[v];

cout<<"Enter names of cities: \n";

for(i=0;i<v;i++)

cin>>name[i];

adjm1(name,v);

}