Practical no.7

/*

⭕ PROBLEM STATEMET:-

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.

***/**

```
#include<iostream>
using namespace std;

string city[10];
int d[10][10];
int size;
int val;

void cityname(){
   cout<<"Enter the number of cities : ";
   cin>>size;
   for(int i=0;i<size;i++){
      cout<<"Enter Name city no. "<<i+1<<" : ";
      cin>>city[i];
   }
}
```

```
void distance(){
   for(int i=0;i<size;i++){
      for(int j=0;j<size;j++){
         if(i!=j && d[i][j]==0){
            cout<<"Enter distance between "<<city[i]<<" -> "<<city[j]<<" : ";
            cin>>val;
            d[i][j]=val;
            d[j][i]=val;
         }
      }
   }
}

void add(){
   cout<<"Enter Name of city : ";
   cin>>city[size];
   size++;
   distance();
}

void display(){
   for(int i=0;i<size;i++){
      cout<<"         "<<city[i];
   }
   cout<<endl;
   cout<<endl;
   for(int i=0;i<size;i++){
      cout<<city[i];
      for(int j=0;j<size;j++){
```

```
            cout<<"        "<<d[i][j];
        }
        cout<<endl;
        cout<<endl;
    }
}

int main(){
    cityname();
    distance();
    int ch;
    while(true){
        cout<<"1. Add a city "<<endl;
        cout<<"2. Display weight representation : "<<endl;
        cout<<"3. Exit"<<endl;
        cout<<"enter choice : ";
        cin>>ch;
        if(ch==1){
            add();
        }
        else if(ch==2){
            display();
        }
        else if(ch==3){
            break;
        }
    }
}
```

Output

Enter the number of cities : 2

Enter Name city no. 1 : a

Enter Name city no. 2 : b

Enter distance between a -> b : 40

1. Add a city

2. Display weight representation :

3. Exit

enter choice : 2

	a	b
a	0	40
b	40	0

1. Add a city

2. Display weight representation :

3. Exit

enter choice : 3

PS C:\Users\kalya\Downloads\new(updated)\git1.py> cd "c:\Users\kalya\Downloads\new(updated)\git1.py\" ; if ($?) { g++ giit7.cpp -o giit7 } ; if ($?) { .\giit7 }

Enter the number of cities : 2

Enter Name city no. 1 : a

Enter Name city no. 2 : b

Enter distance between a -> b : 5

1. Add a city

2. Display weight representation :

3. Exit

enter choice : 1

Enter Name of city : c

Enter distance between a -> c : 3

Enter distance between b -> c : 6

1. Add a city

2. Display weight representation :

3. Exit

enter choice : 2

	a	b	c
a	0	5	3
b	5	0	6
c	3	6	0

1. Add a city

2. Display weight representation :

3. Exit