Assignment No: 6

/* **Title :** 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 takes
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.
*/

```cpp
#include<iostream>
#include<string.h>
using namespace std;
class flight
{
        public:
                int am[10][10];
        char city_index[10][10];
                flight();
                int create();
                void display(int city_count);

};
flight::flight()
{
        int i,j;
        for(i=0;i<10;i++)
        {
                strcpy(city_index[i],"xx");
        }
        for(i=0;i<10;i++)
        {
                for(j=0;j<10;j++)
                {
                        am[i][j]=0;
                }
        }
}
int flight::create()
{
        int city_count=0,j,si,di,wt;
        char s[10],d[10],c;
        do
        {
                cout<<"\n\tEnter Source City     : ";
                cin>>s;
                cout<<"\n\tEnter Destination City : ";
                cin>>d;
                for(j=0;j<10;j++)
                {
                                if(strcmp(city_index[j],s)==0)
                                        break;
                }
                if(j==10)
```

```cpp
                {
                        strcpy(city_index[city_count],s);
                                city_count++;
                }

                for(j=0;j<10;j++)
                {
                                if(strcmp(city_index[j],d)==0)
                                        break;
                }

                if(j==10)
                {
                        strcpy(city_index[city_count],d);
                                city_count++;
                }

                cout<<"\n\t Enter Distance From "<<s<<" And "<<d<<": ";
                cin>>wt;
                for(j=0;j<10;j++)
                {
                                if(strcmp(city_index[j],s)==0)
                                        si=j;
                                if(strcmp(city_index[j],d)==0)
                                        di=j;
                }
                am[si][di]=wt;
                cout<<"\n\t Do you want to add more cities.....(y/n) : ";
                cin>>c;
        }while(c=='y'||c=='Y');
 return(city_count);
}

void flight::display(int city_count)
{
        int i,j;
        cout<<"\n\t Displaying Adjacency Matrix :\n\t";
        for(i=0;i<city_count;i++)
                cout<<"\t"<<city_index[i];
        cout<<"\n";

        for(i=0;i<city_count;i++)
        {
                cout<<"\t"<<city_index[i];
                for(j=0;j<city_count;j++)
                {
                 cout<<"\t"<<am[i][j];
                }
                cout<<"\n";
        }
}
```

```cpp
int main()
{
        flight f;
        int n,city_count;
        char c;
        do
        {
                cout<<"\n\t***** Flight Main Menu *****";
                cout<<"\n\t1. Create \n\t2. Adjacency Matrix\n\t3. Exit";
                cout<<"\n\t.....Enter your choice : ";
                cin>>n;
                switch(n)
                {
                        case 1:
                                        city_count=f.create();
                                        break;
                        case 2:
                                        f.display(city_count);
                                        break;
                        case 3:
                                        return 0;
                }
                cout<<"\n\t Do you Want to Continue in Main Menu....(y/n) : ";
                cin>>c;
        }while(c=='y'||c=='Y');
        return 0;
}
```

/* **Output :**

sspm@sspm-OptiPlex-390:~$ g++ flight.cpp
sspm@sspm-OptiPlex-390:~$
sspm@sspm-OptiPlex-390:~$ ./a.out

        ***** Flight Main Menu *****
        1. Create
        2. Adjacency Matrix
        3. Exit
        .....Enter your choice : 1

        Enter Source City   : a

        Enter Destination City : b

         Enter Distance From a And b: 10

         Do you want to add more cities.....(y/n) : y

        Enter Source City   : b

        Enter Destination City : c

Enter Distance From b And c: 20

Do you want to add more cities.....(y/n) : y

Enter Source City   : c

Enter Destination City : a

Enter Distance From c And a: 50

Do you want to add more cities.....(y/n) : n

Do you Want to Continue in Main Menu....(y/n) : y

***** Flight Main Menu *****
1. Create
2. Adjacency Matrix
3. Exit
.....Enter your choice : 2

Displaying Adjacency Matrix :
            a   b   c
a   0   10   0
b   0   0   20
c   50   0   0

Do you Want to Continue in Main Menu....(y/n) :
*/