BRACT's
VISHWAKARMA INSTITUTE OF INFORMATION TECHNOLOGY, PUNE – 48
An Autonomous Institute Affiliated to Savitribai Phule Pune University, Pune

SD(LP-II) ASSIGNMENT (S.Y.B. Tech. – DIV: C)

Name: Prajwal Dabhade;

Roll no.: 223012;

GR. No.: 17u385; Batch: C1

*****************************Assignment No.3*********************************

- **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 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. Justify the storage representations used.
- **Objective:** We have to implement this using graph data structure.
- **Theory**: Graph is a data structure that consists of following two components:
  1. A finite set of vertices also called as nodes.
  2. A finite set of ordered pair of the form (u, v) called as edge.

Graphs are used to represent many real-life applications: Graphs are used to represent networks. The networks may include paths in a city or telephone network or circuit network.

Following two are the most commonly used representations of a graph.
1. Adjacency Matrix
2. Adjacency List

The choice of the graph representation is situation specific. It totally depends on the type of operations to be performed and ease of use.

Applications:

- In Computer science graphs are used to represent the flow of computation.
- Google maps uses graphs for building transportation systems, where intersection of two(or more) roads are considered to be a vertex and the road connecting two vertices is considered to be an edge, thus their navigation system is based on the algorithm to calculate the shortest path between two vertices.
- In Facebook, users are considered to be the vertices and if they are friends then there is an edge running between them.

- In Operating System, we come across the Resource Allocation Graph where each process and resources are considered to be vertices. Edges are drawn from resources to the allocated process, or from requesting process to the requested resource. If this leads to any formation of a cycle then a deadlock will occur.

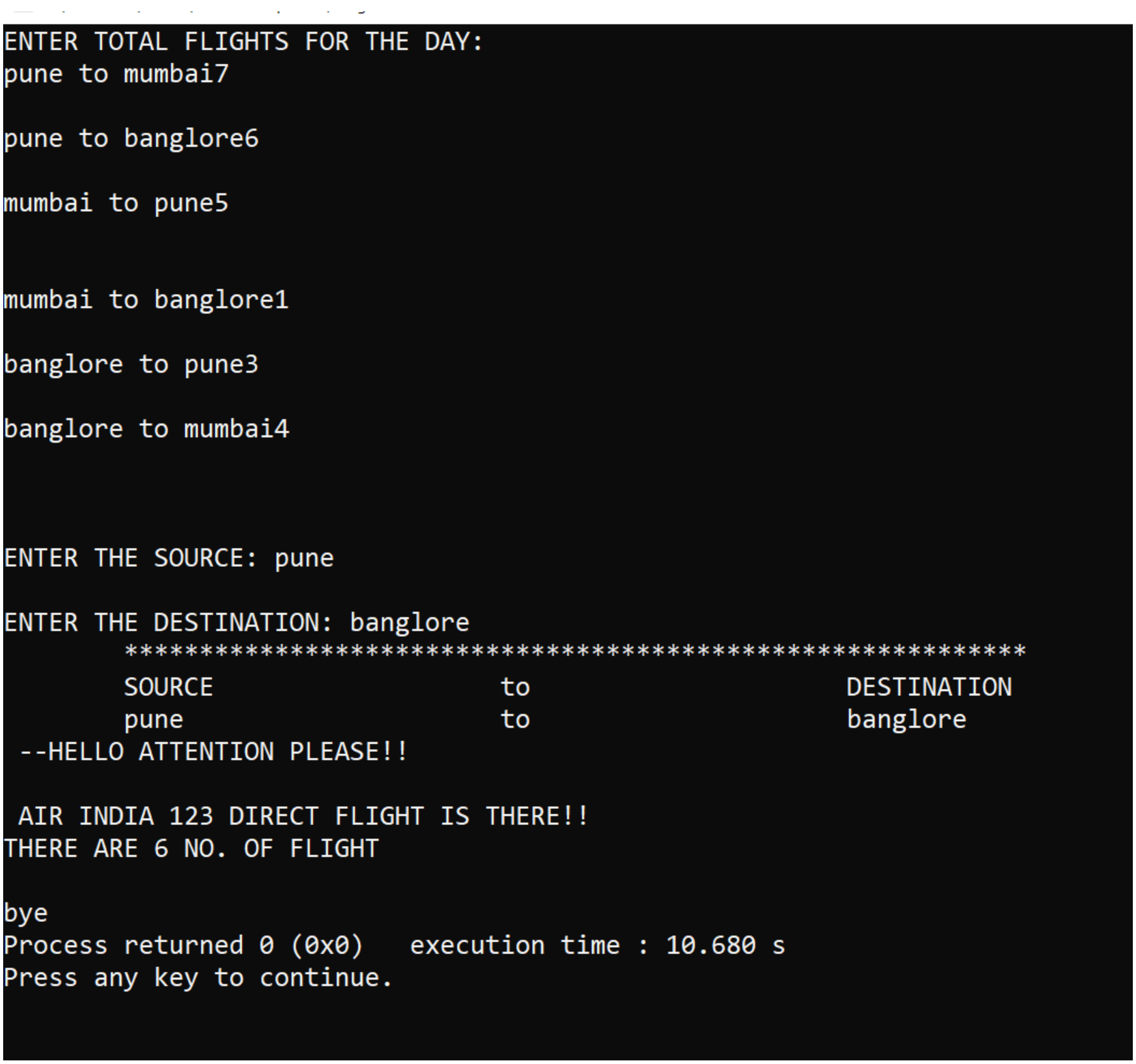
- **Program:**

```
#include<iostream>

#include<conio.h>

using namespace std;


class flight

{

public:

        int A[3][3];

        string r[4];


     void create()

     {

         int i;

        r[1]="pune";

        r[2]="mumbai";

        r[3]="banglore";

      cout<<"ENTER TOTAL FLIGHTS FOR THE DAY: ";

        for(int k=1;k<=3;k++)

        {

            for(int j=1;j<=2;j++)

            {
```

```
         A[k][j]=0;
        }
    }
    for(int k=1;k<=3;k++)
    {
        for(int j=1;j<=3;j++)
        {
            if(k!=j)
            {
            cout<<r[k]<<" to "<<r[j];
                cin>>A[k][j];
        }cout<<"\n";
        }
    }
}
void display()
{
    for(int i=1;i<=3;i++)
    {
        for(int j=1;j<=3;j++)
        {
            if(A[i][j]!=0)
               {
                cout<<A[i][j]<<" ";
               }
        }cout<<"\n";
    }
}

void input()
{
    string source,destination;
    int i,j;
        cout<<"\nENTER THE SOURCE: ";
        cin>>source;
        cout<<"\nENTER THE DESTINATION: ";
        cin>>destination;

        if(source=="pune")
        {
            i=1;
        }
        else if(source=="mumbai")
        {
            i=2;
        }
        else if(source=="banglore")
        {
            i=3;
        }
```

```
          if(destination=="pune")
         {
            j=1;
         }
         else if(destination=="mumbai")
         {
            j=2;
         }
         else if(destination=="banglore")
         {
            j=3;
         }

cout<<"\t"<<"************************************************************\n";
         cout<<" ";
         cout<<"\t"<<"SOURCE"<<"\t \t \t "<<"to"<<"\t \t \t"<<"DESTINATION"<<"\n";
         cout<<"\t"<<source<<"\t \t \t "<<"to"<<"\t \t \t"<<destination;

         search_value(i,j);

         cout<<"\nbye";
   }

  void search_value(int i,int j)
  {
  if(A[i][j]!=0)
  {
      cout<<"\n --HELLO ATTENTION PLEASE!!\n \n AIR INDIA 123 DIRECT FLIGHT IS THERE!!\n";
      cout<<"THERE ARE "<<A[i][j]<<" NO. OF FLIGHT\n";
  }else{
  cout<<"\n SORRY FOR THE INCONVINIECE!\n\n CURRENTLY WE DO NOT HAVE ANY DIRECT FLIGHT FOR THIS ROUTE\n";
  getch();
  }
  }
};
int main()
{   flight f1;
       f1.create();
       //f1.display();
       f1.input();


   return 0;
}
```

- **Output:**



```
ENTER TOTAL FLIGHTS FOR THE DAY:
pune to mumbai7

pune to banglore6

mumbai to pune5


mumbai to banglore1

banglore to pune3

banglore to mumbai4



ENTER THE SOURCE: pune

ENTER THE DESTINATION: banglore
        ************************************************************
        SOURCE                   to                     DESTINATION
        pune                     to                     banglore
 --HELLO ATTENTION PLEASE!!

 AIR INDIA 123 DIRECT FLIGHT IS THERE!!
THERE ARE 6 NO. OF FLIGHT

bye
Process returned 0 (0x0)   execution time : 10.680 s
Press any key to continue.
```

- **Conclusion:** Thus, we have studied graph data structure by implementing this example.