**Assignment number: 5**

**Subject: ADVANCED DATA STRUCTURES LAB**

Name: ***RIA MITTAL***

Class: ***SECOND YEAR ENGINEERING***

Division: ***B***

Roll no: ***222008***

Batch: ***B1***

**PROBLEM STATEMENT:**

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 representation used.

**Code:**

#include<iostream>

using namespace std;

class node

{

friend class list;

string name;

node \*next;

int time;

};

int n;

class list

{

node \*a[100];

string city[100];

int adj[100][100];

public:

int accept();

int display();

int matrix();

int get\_index(string);

public :

list()

{

for(int i=0;i<n;i++)

for(int j=0;j<n;j++)

adj[i][j]=0;

}

};

int list::accept()

{

char ch;

int j=0;

for(int i=0;i<n;i++)

{

cout<<"\n-------------new city---------------\n";

node \*temp=new node;

cout<<"\nenter the name of the city\n";

cin>>temp->name;

city[i]=temp->name;

a[i]=temp;

cout<<"\n----enter the destinations and flight timings-----\n";

do

{

j++;

node \*s=new node;

cout<<"\nenter the destination \n";

cin>>s->name;

cout<<"\nenter the timings\n";

cin>>s->time;

temp->next=s;

temp=temp->next;

cout<<"\nare there any more cities to fly to?(y/n)\n";

cin>>ch;

}while(ch=='y' || ch=='Y' && j<n);

}

}

int list::get\_index(string nme)

{

for(int i=0;i<n;i++)

{

if(city[i]==nme)

return i;

}

}

int list::matrix()

{

node \*temp;

for(int i=0;i<n;i++)

{

temp=a[i]->next;

while(temp!=NULL)

{

int index=get\_index(temp->name);

adj[i][index]=temp->time;

temp=temp->next;

}

}

}

int list::display()

{

for(int i=0;i<n;i++)

cout<<"\t"<<city[i];

cout<<endl;

for(int i=0;i<n;i++)

{

cout<<city[i]<<"\t";

for(int j=0;j<n;j++)

{

cout<<adj[i][j]<<"\t";

}

cout<<endl;

}

}

int main()

{

list obj;

cout<<"\nenter the total number of flights\n";

cin>>n;

obj.accept();

obj.matrix();

obj.display();

return 0;

}

**OUTPUT:**

enter the total number of flights

4

-------------new city---------------

enter the name of the city

pune

----enter the destinations and flight timings-----

enter the destination

delhi

enter the timings

12

are there any more cities to fly to?(y/n)

y

enter the destination

mumbai

enter the timings

6

are there any more cities to fly to?(y/n)

n

-------------new city---------------

enter the name of the city

mumbai

----enter the destinations and flight timings-----

enter the destination

delhi

enter the timings

4

are there any more cities to fly to?(y/n)

y

enter the destination

kolkata

enter the timings

3

are there any more cities to fly to?(y/n)

n

-------------new city---------------

enter the name of the city

kolkata

----enter the destinations and flight timings-----

enter the destination

pune

enter the timings

10

are there any more cities to fly to?(y/n)

y

enter the destination

delhi

enter the timings

1

are there any more cities to fly to?(y/n)

n

-------------new city---------------

enter the name of the city

delhi

----enter the destinations and flight timings-----

enter the destination

pune

enter the timings

11

are there any more cities to fly to?(y/n)

y

enter the destination

mumbai

enter the timings

7

are there any more cities to fly to?(y/n)

y

enter the destination

kolkata

enter the timings

9

are there any more cities to fly to?(y/n)

n

pune mumbai kolkata delhi

pune 0 6 0 12

mumbai 0 0 3 4

kolkata 10 0 0 1

delhi 11 7 9 0