// 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.

//============================================================================

#include <iostream>

#include<iomanip>

using namespace std;

const int MAX=30;

class node

{

node \*next;

string city;

int timeCost;

public:

friend class graph;

node()

{

next=NULL;

city="";

timeCost=-1;

}

node(string city,int weight)

{

next=NULL;

this->city=city;

timeCost=weight;

}

};

class graph

{

node \*head[MAX];

int n;

public:

graph(int num)

{

n=num;

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

head[i]=NULL;

}

void insert(string city1,string city2,int time);

void insertUndirected(string city1,string city2,int time);

void readdata(int gType);

int getindex(string s1);

void outFlights();

void inFlights();

};

void graph::inFlights()

{

int count[n];

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

count[i]=0;

cout<<"====== In degree =========\n";

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

{

cout<<"\n"<<setw(8)<<"Source"<<setw(8)<<"Destin."<<setw(8)<<"Time";

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

{

node \*p=head[j]->next;

while(p!=NULL)

{

if(p->city==head[i]->city)

{

count[i]=count[i]+1;

cout<<"\n"<<setw(8)<<head[j]->city<<setw(8)<<head[i]->city<<setw(8)<<p->timeCost;

}

p=p->next;

}

}

cout<<"\nFlights to "<<head[i]->city<<" = "<<count[i]<<endl;

cout<<"-------------------------------------\n";

}

}

void graph::outFlights()

{

int count;

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

{

node \*p=head[i]->next;

count=0;

cout<<"\n"<<setw(8)<<"Source"<<setw(8)<<"Destin."<<setw(8)<<"Time";

if(p==NULL)

{

cout<<"\nNo Flights from "<<head[i]->city;

}

else

{

while(p!=NULL)

{

cout<<"\n"<<setw(8)<<head[i]->city<<setw(8)<<p->city<<setw(8)<<p->timeCost;

count++;

p=p->next;

}

}

cout<<"\nNo. of flights: "<<count<<endl;;

cout<<"-------------------------------------\n";

}

}

int graph::getindex(string s1)

{

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

{

if(head[i]->city==s1)

return i;

}

return -1;

}

void graph::insert(string city1,string city2,int time)

{

node \*source;

node \*dest=new node(city2,time);

int ind=getindex(city1); //for getting head nodes index in array

if(head[ind]==NULL)

head[ind]=dest;

else

{

source=head[ind];

while(source->next!=NULL)

source=source->next;

source->next=dest;

}

}

void graph::insertUndirected(string city1,string city2,int time)

{

node \*source;

node \*dest=new node(city2,time);

node \*dest2=new node(city1,time); //for second flight insertion

int ind=getindex(city1); //for getting head nodes index in array

int ind2=getindex(city2);

if(head[ind]==NULL && head[ind2]==NULL) //when no flights in graph

{

head[ind]=dest;

head[ind2]=dest2;

}

else if(head[ind]==NULL && head[ind2]!=NULL) //no flight in first list but flight in second list

{

head[ind]=dest; //inserted first flight

source=head[ind2];

while(source->next!=NULL)

source=source->next;

source->next=dest2;

}

else if(head[ind]!=NULL && head[ind2]==NULL)

{

head[ind2]=dest2; //inserted first flight

source=head[ind];

while(source->next!=NULL)

source=source->next;

source->next=dest;

}

else

{

source=head[ind];

while(source->next!=NULL)

source=source->next;

source->next=dest;

source=head[ind2];

while(source->next!=NULL)

source=source->next;

source->next=dest2;

}

}

void graph::readdata(int gType)

{

string city1,city2,tmpcity;

int fcost;

int flight;

cout<<"\nENter City Details:\n ";

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

{

head[i]=new node;

cout<<"Enter City "<<i+1<<" ";

cin>>tmpcity;

head[i]->city=tmpcity;

}

cout<<"\nEnter Number of Flights to insert: ";

cin>>flight;

if(gType==1)

{

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

{

cout<<"\nEnter Source:";

cin>>city1;

cout<<"Enter Destination:";

cin>>city2;

cout<<"Enter Time:";

cin>>fcost;

insert(city1,city2,fcost);

}

}

else

{

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

{

cout<<"\nEnter Source:";

cin>>city1;

cout<<"Enter Destination:";

cin>>city2;

cout<<"Enter Time:";

cin>>fcost;

insertUndirected(city1,city2,fcost);

//cout<<"\ninserted"<<i+1;

}

}

}

int main() {

int number,choice;

int graphype;

cout<<"0. Undirected\n1.Directed\nEnter Flight data Insertion Type:";

cin>>graphype;

cout<<"\nENter Number of Airport Stations:";

cin>>number;

graph g1(number);

g1.readdata(graphype);

do

{

cout<<"------Menu------"

<<"\n1.Incoming Flights(In degree)"

<<"\n2.Outgoing Flights(Out degree)"

<<"\n3.Exit"

<<"\nEnter your choice: ";

cin>>choice;

switch(choice)

{

case 1:

cout <<"" << endl;

g1.inFlights();

break;

case 2:

g1.outFlights();

break;

default:

cout<<"\nWrong Choice";

}

}while(choice!=3);

// prints

return 0;

}

/\*OUTPUT

0. Undirected

1.Directed

Enter Flight data Insertion Type:0

ENter Number of Airport Stations:3

ENter City Details:

Enter City 1 Pune

Enter City 2 Mumbai

Enter City 3 Baramati

Enter Number of Flights to insert: 3

Enter Source:Pune

Enter Destination:Mumbai

Enter Time:100

Enter Source:Pune

Enter Destination:Baramati

Enter Time:50

Enter Source:Mumbai

Enter Destination:Baramati

Enter Time:130

------Menu------

1.Incoming Flights(In degree)

2.Outgoing Flights(Out degree)

3.Exit

Enter your choice: 1

====== In degree =========

Source Destin. Time

Mumbai Pune 100

Baramati Pune 50

Flights to Pune = 2

-------------------------------------

Source Destin. Time

Pune Mumbai 100

Baramati Mumbai 130

Flights to Mumbai = 2

-------------------------------------

Source Destin. Time

PuneBaramati 50

MumbaiBaramati 130

Flights to Baramati = 2

-------------------------------------

------Menu------

1.Incoming Flights(In degree)

2.Outgoing Flights(Out degree)

3.Exit

Enter your choice: 2

Source Destin. Time

Pune Mumbai 100

PuneBaramati 50

No. of flights: 2

-------------------------------------

Source Destin. Time

Mumbai Pune 100

MumbaiBaramati 130

No. of flights: 2

-------------------------------------

Source Destin. Time

Baramati Pune 50

Baramati Mumbai 130

No. of flights: 2

-------------------------------------

------Menu------

1.Incoming Flights(In degree)

2.Outgoing Flights(Out degree)

3.Exit

Enter your choice: 3

Wrong Choice

Process returned 0 (0x0) execution time : 99.867 s

Press any key to continue.\*/