#include<iostream>

# define max 20

using namespace std;

char v1;

class graph

{

int g[max][max],n,c[max],ch[max],min\_dist,client\_dist,visit\_dist;

char v[max],str[max][max],min\_path[max],client\_path[max],visit\_path[max];

public:

graph(int m)

{

n=m;

visit\_dist=0;

client\_dist=0;

min\_dist=0;

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

{

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

{

g[i][j]=0;

}

}

}

void accept\_v();

void accept\_e();

void display();

void dj\_init(int);

void dj\_init();

void dj(char,char,int);

void dj\_client(char,char);

};

void graph::accept\_v()

{

int i=0;

cout<<"\n\t\t\tNames Of Landmarks";

while(i<n)

{

cout<<"\nEnter Name of Landmark ["<<i+1<<"] : ";

cin>>v[i];

i++;

}

cout<<"\n";

}

void graph::accept\_e()

{

char ch;

char v1,v2;

int i,j,cst;

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

for(j=i;j<n;j++)

{

if(i==j)

{

g[i][j]=0;

continue;

}

else

cout<<"\n\tDistance Between Landmark ["<<v[i]<<"]["<<v[j]<<"] : ";

cin>>cst;

g[i][j]=g[j][i]=cst;

}

cout<<"\n";

}

void graph::display()

{

int i,j;

i=0;

cout<<"\n";

while(i<n)

{

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

i++;

}

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

{

cout<<"\n"<<v[i];

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

{

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

}

}

cout<<"\n";

}

void graph::dj\_init()

{

int i,j;

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

{

c[i]=9999;

ch[i]=0;

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

{

str[i][j]='-';

}

}

cout<<"\n";

}

void graph::dj\_init(int i )

{

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

{

str[i][j]='-';

}

}

void graph::dj(char s,char d,int f)

{

dj\_init();

int i,j,l,k,flag,min=999,cst=0;

i=0;

while(v[i]!=s)

{

i++;

}

c[i]=0;

k=0;

str[i][k]=v[i];

do

{

ch[i]=1;

min=999;

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

{

flag=0;

cst=c[i]+g[i][j];

if(g[i][j]!=0 && i!=j && cst<c[j])

{

k=0;

c[j]=cst;

dj\_init(j);

while(flag==0)

{

str[j][k]=str[i][k];

k++;

if(str[i][k]=='-')

{

flag=1;

}

}

str[j][k]=v[j];

}

}

for(l=0;l<n;l++)

{

if(c[l]<=min&&ch[l]==0)

{

min=c[l];

i=l;

}

}

}

while(v[i]!=d);

j=0;

if(f==0)

{

min\_dist=c[i];

}

else

{

client\_dist=c[i];

}

if(f==0)

{

cout<<"Minimum Distance : "<<c[i]<<"\nShortest Path : ";

}

while(str[i][j]!='-')

{

if(f==0)

{

min\_path[j]=str[i][j];

}

if(f==0)

{

cout<<" "<<str[i][j];

}

j++;

}

min\_path[j]='\0';

cout<<"\n";

j=1;

while(str[i][j]!='-'&&f==1)

{

client\_path[j-1]=str[i][j];

j++;

}

client\_path[j-1]='\0';

cout<<"\nShortest Distance From "<<v1<<" To All Destinations:\n";

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

{

cout<<"\nFrom "<<v1<<" To "<<v[h]<<" Is: "<<c[h]<<"\n";

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

{

cout<<" "<<str[h][m];

}

}

cout<<"\n";

}

int main()

{

int n,ch;

char v2;

cout<<"\nEnter Number of Landmarks : ";

cin>>n;

graph g(n);

do

{

cout<<"\n1.Accept Names Of Landmarks.\n2.Accept Distance Between Landmarks.\n3.Display Adjacency Matrix.\n4.Display Shortest Distance.\n5.Exit.";

cout<<"\nEnter Your Choice: ";

cin>>ch;

switch(ch)

{

case 1:

g.accept\_v();

break;

case 2:

g.accept\_e();

break;

case 3:

g.display();

break;

case 4:

cout<<"\nEnter Source(Name Of Landmark From Where You Want To Start) : ";

cin>>v1;

cout<<"\nEnter Destination(Name Of Landmark Where You Want To Reach) : ";

cin>>v2;

g.dj(v1,v2,0);

break;

case 5:

cout<<"You Have Successfully Exitted....";

break;

default:

cout<<"INVALID CHOICE.";

}

}

while(ch!=5);

return 0;

}