#include <iostream>

#define HIGH 99999

using namespace std;

#include<cstring>

class graph

{

int g[10][10],n;

int v\_array[10];

string dept[10];

public:

void initialize()

{

cout<<"\n Enter total department: ";

cin>>n;

cout<<"\n Enter department names: ";

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

{

cin>>dept[i];

}

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

{

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

{

cout<<"\n Enter distance between "<<dept[i]<<" and "<<dept[j]<<" :";

cin>>g[i][j];

}

}

}

void display()

{

cout<<" ";

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

{

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

}

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

{

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

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

{

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

}

}

}

void prims()

{

int cost=0;

memset(v\_array,false,sizeof(v\_array));

v\_array[0]=true;

cout<<"\nEDGE:Weight\n";

int n\_edges=0;

while(n\_edges<n-1)

{

int min=HIGH;

int r=0;

int c=0;

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

{

if(v\_array[i])

{

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

{

if(min>g[i][j])

{

if(!v\_array[j]&&g[i][j])

{ min=g[i][j];

r=i;

c=j;

}

}

}

}

}

cout<<"\n "<<dept[r]<<"-"<<dept[c]<<":"<<g[r][c];

cost=cost+g[r][c];

v\_array[c]=true;

n\_edges++;

}

cout<<"\n Cost of minimum spanning tree using Prim's: "<<cost;

}

};

int main()

{

graph g;

g.initialize();

g.display();

g.prims();

return 0;

}

OUTPUT:

Enter total department: 3

Enter department names:

IT

ENTC

CE

Enter distance between IT and IT :0

Enter distance between IT and ENTC :2

Enter distance between IT and CE :3

Enter distance between ENTC and IT :2

Enter distance between ENTC and ENTC :0

Enter distance between ENTC and CE :1

Enter distance between CE and IT :3

Enter distance between CE and ENTC :1

Enter distance between CE and CE :0

IT ENTC CE

IT0 2 3

ENTC2 0 1

CE3 1 0

EDGE:Weight

IT-ENTC:2

ENTC-CE:1

Cost of minimum spanning tree using Prim's: 3