**/\*TE CMPN A-27\*/**

**Expt : 9**

import java.util.\*;

class Expt\_9

{

public static void main(String arg[])

{

Scanner sc=new Scanner(System.in);

Dijkstra d=new Dijkstra();

System.out.println("Enter no. of nodes:");

int n=sc.nextInt();

System.out.println("Enter no. of edges:");

int e=sc.nextInt();

d.createGraph(e, n);

System.out.println("Enter source node:");

int x=sc.next().toUpperCase().charAt(0)-65;

d.calcPaths(x);

}

}

class Dijkstra

{

int a[][], dist[], prev[];

final int IN=10000;

boolean select[];

void createGraph(int e, int n)

{

Scanner sc=new Scanner(System.in);

a=new int[n][n];

prev=new int[n];

dist=new int[n];

select=new boolean[n];

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

{

dist[i]=IN;

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

a[i][j]=IN;

}

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

{

System.out.println("Enter edge name:");

String s=sc.next();

s=s.toUpperCase();

int x=s.charAt(0)-65;

int y=s.charAt(1)-65;

System.out.println("Enter weight:");

int wt=sc.nextInt();

a[x][y]=wt;

}

}

void calcPaths(int x)

{

int currdist;

dist[x]=0;

select[x]=true;

int src=x;

for(int k=1;k<a.length;k++)

{

for(int i=0;i<a.length;i++)

{

currdist=a[x][i]+dist[x];

if(currdist<dist[i]&&!select[i])

{

dist[i]=currdist;

prev[i]=x;

}

}

int min=IN, m=0;

for(int i=0;i<dist.length;i++)

{

if(min>dist[i]&&!select[i])

{

min=dist[i];

m=i;

}

}

x=m;

select[x]=true;

}

System.out.println();

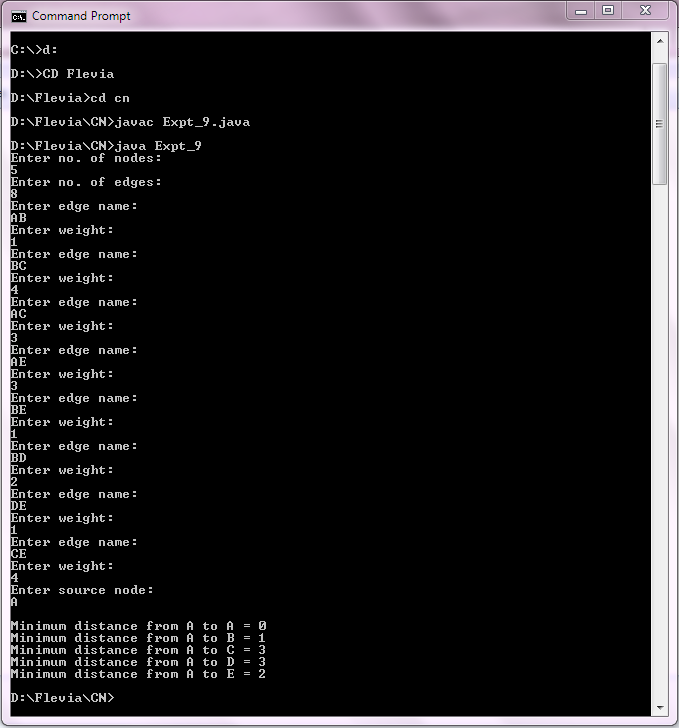
for(int i=0;i<a.length;i++)

System.out.println("Minimum distance from "+(char)(src+65)+" to "+(char)(i+65)+" = "+dist[i]);

}

}

**/\*OUTPUT:**



\*/