**Routing.java :-**

import java.util.Scanner;

public class Routing

{

public static void main(String[] args)

{

Scanner sc = new Scanner(System.in);

final int inf= 9999;

int i,j,flag,vertices;

int adjmat[][] = new int[10][10];

int cost[][] = new int[10][10];

System.out.println("Pls. enter no. of routers in the network...");

vertices=sc.nextInt();

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

{

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

{

adjmat[i][j]=0;

cost[i][j]=inf;

}

}

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

{

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

{

if(i!=j)

{

System.out.println("Is there a link b/w routers:"+i+" & "+j+" [1.Yes 2.NO]");

flag=sc.nextInt();

if(flag==1)

{

adjmat[i][j]=1;

}

}

}

}

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

{

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

{

if(adjmat[i][j]==1)

{

System.out.println("Enter distance b/w the routers"+i+" & "+j);

flag=sc.nextInt();

cost[i][j]=flag;

}

else

{

cost[i][j]=inf;

}

}

}

System.out.println("Adjacency Matrix Reprsentation");

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

{

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

{

System.out.print(adjmat[i][j]+"\t");

}

System.out.println();

}

System.out.println("-----------------------------------------------------");

System.out.println("Cost Matrix Reprsentation");

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

{

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

{

System.out.print(cost[i][j]+"\t");

}

System.out.println();

}

int start;

System.out.println("Enter starting router");

start=sc.nextInt();

int distance[] = new int[10];

int pred[] = new int[10];

int visited[] = new int[10];

int count,mindistance,next=0;

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

{

distance[i]=cost[start][i];

pred[i]=start;

visited[i]=0;

}

count=0;

visited[start]=1;

distance[start]=0;

while(count<vertices)

{

mindistance=inf;

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

{

if(distance[i]<mindistance && visited[i]!=1)

{

mindistance=distance[i];

next=i;

}

}

visited[next]=1;

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

{

if(visited[i]!=1)

{

if(mindistance+cost[next][i]<distance[i])

{

distance[i]=mindistance+cost[next][i];

pred[i]=next;

}

}

}

count++;

}

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

{

if(i!=start)

{

System.out.println();

System.out.println("The Shortest Distance b/w "+start+" & "+i+" = "+distance[i]);

System.out.println("The Path followed is:");

System.out.print(i);

j=i;

do

{

j=pred[j];

System.out.print("<--"+j);

}while(j!=start);

System.out.println();

}

}

sc.close();

}

}

**Output :-**

D:\CN Lab\CN Final\Link State Routing>java Routing

Pls. enter no. of routers in the network...

4

Is there a link b/w routers:0 & 1 [1.Yes 2.NO]

1

Is there a link b/w routers:0 & 2 [1.Yes 2.NO]

0

Is there a link b/w routers:0 & 3 [1.Yes 2.NO]

1

Is there a link b/w routers:1 & 0 [1.Yes 2.NO]

1

Is there a link b/w routers:1 & 2 [1.Yes 2.NO]

1

Is there a link b/w routers:1 & 3 [1.Yes 2.NO]

1

Is there a link b/w routers:2 & 0 [1.Yes 2.NO]

0

Is there a link b/w routers:2 & 1 [1.Yes 2.NO]

1

Is there a link b/w routers:2 & 3 [1.Yes 2.NO]

1

Is there a link b/w routers:3 & 0 [1.Yes 2.NO]

1

Is there a link b/w routers:3 & 1 [1.Yes 2.NO]

1

Is there a link b/w routers:3 & 2 [1.Yes 2.NO]

1

Enter distance b/w the routers0 & 1

2

Enter distance b/w the routers0 & 3

2

Enter distance b/w the routers1 & 0

2

Enter distance b/w the routers1 & 2

3

Enter distance b/w the routers1 & 3

5

Enter distance b/w the routers2 & 1

3

Enter distance b/w the routers2 & 3

3

Enter distance b/w the routers3 & 0

2

Enter distance b/w the routers3 & 1

5

Enter distance b/w the routers3 & 2

3

Adjacency Matrix Reprsentation

0 1 0 1

1 0 1 1

0 1 0 1

1 1 1 0

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

Cost Matrix Reprsentation

9999 2 9999 2

2 9999 3 5

9999 3 9999 3

2 5 3 9999

Enter starting router

0

The Shortest Distance b/w 0 & 1 = 2

The Path followed is:

1<--0

The Shortest Distance b/w 0 & 2 = 5

The Path followed is:

2<--1<--0

The Shortest Distance b/w 0 & 3 = 2

The Path followed is:

3<--0