**Program 10: Write and execute a program to find shortest path to all other nodes in weighted graph using Dijkstra’s strategy**

**Output:**

Enter the Number of Nodes

6

Enter the Adjacency Matrix

0 18 2 999 999 999

18 0 2 1 90 999

2 20 0 4 50 999

999 1 3 0 3 100

999 90 50 3 0 4

999 999 999 100 4 0

Enter the Source Node

1

The shortest distance from 1 to 1 is 0

The shortest distance from 1 to 2 is 7

The shortest distance from 1 to 3 is 2

The shortest distance from 1 to 4 is 6

The shortest distance from 1 to 5 is 9

The shortest distance from 1 to 6 is 13

#include<stdio.h>

#include<conio.h>

void main()

{

int n,a[10][10],i,j,source,mstcost;

printf(" Enter the Number of Nodes \n");

scanf("%d",&n);

printf(" Enter the Adjacency Matrix \n");

for(i=1;i<=n;i++)

for(j=1;j<=n;j++)

scanf("%d",&a[i][j]);

printf(" Enter the Source Node \n");

scanf("%d",&source);

Dijkstra(int n, int a[10][10], int source)

{

int d[100],s[10],i,j,u,v,minval,sum=0;

for(i=1;i<=n;i++)

{

s[i]=0;

d[i]=a[source][i];

}

s[source]=1;

for(i=1;i<=n;i++)

{

minval = 999;

for(j=1;j<=n;j++)

if(s[j]==0) if(d[j] < minval)

{

minval = d[j];

u=j;

}

s[u]=1;

sum=sum+d[u];

for(v=1;v<=n;v++)

if(s[v]==0 && d[v] > d[u]+a[u][v])

d[v]=d[u]+a[u][v];

}

for(i=1;i<=n;i++)

printf("The shortest distance from %d to %d is %d \n",source,i,d[i]);

}

Dijkstra(n,a,source);

}

Enter the Number of Nodes

4

Enter the Adjacency Matrix

0 10 999 999

999 0 10 999

999 999 0 10

10 999 999 0

Enter the Source Node

1

The shortest distance from 1 to 1 is 0

The shortest distance from 1 to 2 is 10

The shortest distance from 1 to 3 is 20

The shortest distance from 1 to 4 is 30