**PROGRAM-16**

**From a given vertex in a weighted connected graph, find shortest paths to other vertices using Dijkstra’s algorithm.**

**Code:**

#include<stdio.h>

#define INFINITY 999

#define MAX 10

void dijikstra(int G[MAX][MAX], int n, int startnode);

void main()

{

int G[MAX][MAX], i, j, n, u;

printf("\nEnter the no. of vertices:");

scanf("%d",&n);

printf("\nEnter the adjacency matrix:\n");

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

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

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

printf("\nEnter the starting node:");

scanf("%d",&u);

dijikstra(G,n,u);

getch();

}

void dijikstra(int G[MAX][MAX], int n, int startnode)

{

int cost[MAX][MAX], distance[MAX], pred[MAX];

int visited[MAX], count, mindistance, nextnode, i,j;

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

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

if(G[i][j]==0)

cost[i][j]=INFINITY;

else

cost[i][j]=G[i][j];

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

{

distance[i]=cost[startnode][i];

pred[i]=startnode;

visited[i]=0;

}

distance[startnode]=0;

visited[startnode]=1;

count=1;

while(count<n-1){

mindistance=INFINITY;

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

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

{

mindistance=distance[i];

nextnode=i;

}

visited[nextnode]=1;

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

if(!visited[i])

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

{

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

pred[i]=nextnode;

}

count++;

}

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

if(i!=startnode)

{

printf("\nDistance of %d = %d",i,distance[i]);

printf("\nPath = %d",i);

j=i;

do

{

j=pred[j];

printf("<-%d",j);

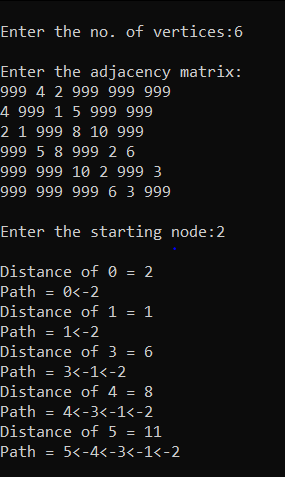
}

while(j!=startnode);

}

}

**Output:**



**Code2:**

#include<stdio.h>

void dij(int,int [20][20],int [20],int [20],int);

void main()

{

int i,j,n,visited[20],source,cost[20][20],d[20];

printf("Enter no. of vertices: ");

scanf("%d",&n);

printf("Enter the cost adjacency matrix\n");

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

{

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

{

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

}

}

printf("\nEnter the source node: ");

scanf("%d",&source);

dij(source,cost,visited,d,n);

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

if(i!=source)

printf("\nShortest path from %d to %d is %d",source,i,d[i]);

getch();

}

void dij(int source,int cost[20][20],int visited[20],int d[20],int n)

{

int i,j,min,u,w;

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

{

visited[i]=0;

d[i]=cost[source][i];

}

visited[source]=1;

d[source]=0;

for(j=2;j<=n;j++)

{

min=999;

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

{

if(!visited[i])

{

if(d[i]<min)

{

min=d[i];

u=i;

}

}

} //for i

visited[u]=1;

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

{

if(cost[u][w]!=999 && visited[w]==0)

{

if(d[w]>cost[u][w]+d[u])

d[w]=cost[u][w]+d[u];

}

} //for w

} // for j

}

