NAME: RAMYA RAMESH

USN: 1BM19CS227

**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 INF 999

#define MAX 10

void dijkstra(int A[MAX][MAX], int n, int start);

void main(){

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

printf("\n\tDijkstra's algorithm\n");

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", &A[i][j]);

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

scanf("%d", &u);

dijkstra(A,n,u);

}

void dijkstra(int A[MAX][MAX], int n, int start)

{

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

int visited[MAX], count, mindist, next, i,j;

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

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

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

cost[i][j]=INF;

else

cost[i][j]=A[i][j];

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

{

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

pred[i]=start;

visited[i]=0;

}

distance[start]=0;

visited[start]=1;

count=1;

while(count < n-1){

mindist=INF;

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

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

{

mindist=distance[i];

next=i;

}

visited[next]=1;

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

if(!visited[i])

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

{

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

pred[i]=next;

}

count++;

}

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

if(i!=start)

{

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

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

j=i;

do

{

j=pred[j];

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

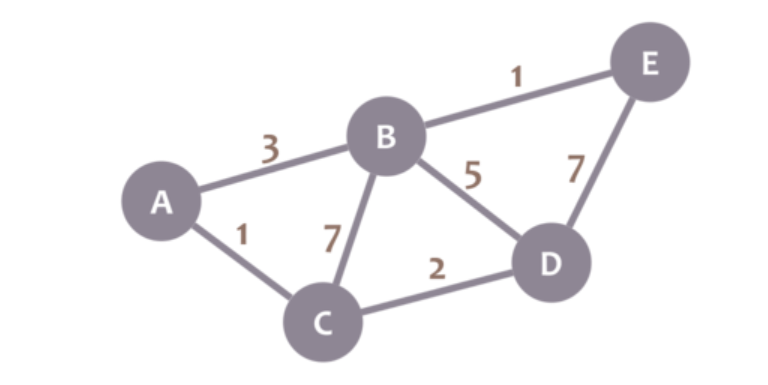
}

while(j!=start);

}

}

OUTPUT:



**Shortest paths starting from Vertex C =**

