**Program No.:-11**

**AIM: Write a program to find the shortest distance between two nodes using Dijkstra’s algorithm.**

**Source Code:**

#include<stdio.h>

#include<conio.h>

#include<limits.h>

int vertex[10],size,weight[10][10],visited[10],key[10];

int min,i,j,source,minimum,k,index;

void build\_graph()

{

printf("enter number of vertices");

scanf("%d",&size);

printf("enter %d vertices of graph",size);

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

{

fflush(stdin);

scanf("%c",&vertex[i]);

}

printf("enter weighted matrix for the graph:\n");

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

{

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

{

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

}

}

}

void initialize\_single\_source()

{

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

{

key[i]=INT\_MAX;

visited[i]=0;

}

key[source]=0;

}

int extract\_min()

{

minimum=INT\_MAX;

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

{

if(!visited[i] && key[i]<minimum)

{

minimum=key[i];

index=i;

}

}

return index;

}

void relax(int p,int q,int w)

{

if(key[q]>key[p]+w)

{

key[q]=key[p]+w;

}

}

void dijkastra()

{

initialize\_single\_source();

for(k=0;k<size;k++)

{

min=extract\_min();

visited[min]=1;

printf("%c\t%d\n",vertex[min],key[min]);

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

{

if(weight[min][j]!=0 && !visited[j])

{

relax(min,j,weight[min][j]);

}

}

}

}

void main()

{

clrscr();

build\_graph();

printf("enter source of graph");

scanf("%d",&source);

printf("the shortest path from source is given as:\n");

dijkastra();

getch();

}

**Output:**

