Ex-12 Implementation of Dijkstra's Algorithm

Program:

```
#include<stdio.h>
#include<conio.h>

void main()
{
 int graph[15][15],s[15],pathestimate[15],mark[15];
 int num_of_vertices,source,i,j,u,predecessor[15];
 int count=0;
 int minimum(int a[],int m[],int k);
 void printpath(int,int,int[]);
 clrscr();
 printf("\nIMPLEMENTATION OF DIJKSTRA's ALGORITHM\n");
 printf("--------------------------------------");
 printf("\nEnter the no.of vertices: ");
 scanf("%d",&num_of_vertices);
 if(num_of_vertices<=0)
 {
  printf("\nThis is meaningless\n");
   exit(1);
 }
 printf("\nEnter the adjacent matrix:\n\n");
 printf("-------------------------");
    for(i=1;i<=num_of_vertices;i++)
 {
  printf("\nEnter the elements of row %d: ",i);
  for(j=1;j<=num_of_vertices;j++)
  {
   scanf("%d",&graph[i][j]);
  }
 }
 printf("Adjacency Matrix for the given graph is:\n");
 for(i=1;i<=num_of_vertices;i++)
 {
 for(j=1;j<=num_of_vertices;j++)
  {
   printf("%d  ",graph[i][j]);
  }
  printf("\n");
 }
  printf("\nEnter the source vertex:\n");
 scanf("%d",&source);
 for(j=1;j<=num_of_vertices;j++)
```

```
 {
  mark[j]=0;
  pathestimate[j]=999;
  predecessor[j]=0;
 }
 pathestimate[source]=0;

 while(count<num_of_vertices)
 {
  u=minimum(pathestimate,mark,num_of_vertices);
  s[++count]=u;
  mark[u]=1;
  for(i=1;i<=num_of_vertices;i++)
  {
   if(graph[u][i]>0)
   {
    if(mark[i]!=1)
    {
    if(pathestimate[i]>pathestimate[u]+graph[u][i])
    {
     pathestimate[i]=pathestimate[u]+graph[u][i];
     predecessor[i]=u;
    }
    }
   }
  }
 }
 for(i=1;i<=num_of_vertices;i++)
 {
 printpath(source,i,predecessor);
 if(pathestimate[i]!=999)
 printf("-> (%d)\n",pathestimate[i]);
 }
getch();
}

int minimum(int a[],int m[],int k)
{
 int mi=999;
 int i,t;
 for(i=1;i<=k;i++)
 {
  if(m[i]!=1)
  {
  if(mi>=a[i])
  {
```

```
   mi=a[i];
   t=i;
  }
 }}
 return t;
}

void printpath(int x,int i,int p[])
{
 printf("\n");
if(i==x)
{
printf("%d",x);
}
else if(p[i]==0)
printf("No path from %d to %d",x,i);
else
{
printpath(x,p[i],p);
printf("...%d",i);
}
}
```

Output:

```
IMPLEMENTATION OF DIJKSTRA's ALGORITHM
----------------------------------------------------------------
Enter the no.of vertices: 7

Enter the adjacent matrix:
------------------------------
Enter the elements of row 1: 0 2 0 1 0 0 0

Enter the elements of row 2: 0 0 0 3 10 0 0

Enter the elements of row 3: 4 0 0 0 0 5 0

Enter the elements of row 4: 0 0 2 0 2 8 4

Enter the elements of row 5: 0 0 0 0 0 0 6

Enter the elements of row 6: 0 0 0 0 0 0 0

Enter the elements of row 7: 0 0 0 0 0 1 0
```

Adjacency Matrix for the given graph is:

0 2 0 1 0 0 0
0 0 0 3 10 0 0
4 0 0 0 0 5 0
0 0 2 0 2 8 4
0 0 0 0 0 0 6
0 0 0 0 0 0 0
0 0 0 0 0 1 0

Enter the source vertex:
1

1-> (0)

1...2-> (2)

1...4...3-> (3)

1...4-> (1)

1...4...5-> (3)

1...4...7...6-> (6)

1...4...7-> (5)