NAME: VALLISHA.M

USN: 1BM19CS177

Course: Analysie and Design of Algorithms (ADA) (Lab)

Date : 05-07-2021

Test : Lab CIE-2

7. From a given vertex in a weighted connected graph, find shortest paths to other vertices using Dijsktra's algorithm.
(Modification : Print number of nodes along the shortest paths).

```
#include <stdio.h>

int minDistance (int dist[],int sptSet[],int V)
{
        int min = 999, min_index;
        int v;
        for (v=0; v<V; v++)
                if(sptSet[v]==0 && dist[v] <= min)
                        min = dist[v], min_index = v;
        return min_index;
}

int printSolution (int dist[], int V, int number[])
{
        int i ;
        printf(" Vertex \t\t Distance from Source \t\t Number \n");

        for (i=0; i<V; i++)
                printf ("%d \t\t %d \t\t %d", i, dist[i],
                        number[i]);
}
```

```
void dijkstra(int graph[10][10], int ~~ser~~ src, int V)
{
        int dist[V];
        int i, count, u,v;
        int sptSet[V];
        for(i =0; i< V; i =i+1)
               dist[i] = 999 , sptSet[i]=0;

        dist [src] =0;
        int number[V] =0;

        for (count =0; count < V-1; count ++)
        {
               u = minDistance(dist, sptSet, V);
               number[u] = count;
               sptSet [u] = 1;
               for (v =0; v<V; v++)
                    if (!sptSet[v] && graph[u][v] && dist[u]!=999
                     && dist [u] + graph[u][v] <dist [v]){
                              dist [v] = dist[u] + graph [u] [v];
                    }
        }

        for (i=0; i<V; i ++){
               if (sptSet[i] == 0) {
                    int max =0;
                    for(int j =0; j<V ; j++){
                           if (number[j] > max && sptSet[j]!=0)
                                 max = number[j];
                    }
                    number [i] = max;
               }
        }
        printSolution(dist, V, number);
}
```

Vallicha. M
1BM19CS177

```
int  main(){
       int  i,j, V;
       int  graph[10][10];
       printf(" Enter number of vertices ");
        Scanf ("%d", &V);
        printf(" Enter the adjasencymatrix :\n");
        for (i = 0; i< V; i++)
                  for(j=0; j<V; j ++)
                           Scanf("%d",&graph[i][j]);

           printf("Enter source vertex ; ");
           int src;
           Scanf ("%d", &src);
            dijkstra(graph, src, V);
            return 0;
}
```