Shrihari K Mangle
CS-B2-74 12010517

# LAB-10

Implement Dijkstras Algorithm for Single Source Shortest Path.

**CODE :**

```c
#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
#include <string.h>
#include <math.h>
# define v 5

int vertex(int dist[], int vset[])
{
    int index, min = 999;

    for (int i = 0; i < v; i++)
    {
        if (vset[i] == 0 && dist[i] < min)
        {
            min = dist[i];
            index = i;
        }
    }
    return index;
}

void display(int dist[]){
    printf("Vertex\tDistance from source\n");
    for (int i = 0; i < v; i++)
    {
        printf("%d\t%d\n", i, dist[i]);
    }

}

void Dijkstras(int G[v][v], int src){
    int dist[10], vset[10], u;
    for (int i = 0; i < v; i++)
    {
        dist[i] = 999;
        vset[i] = 0;
    }

    dist[src] = 0; // Mark source node vertex dist as 0
```

```
    for (int i = 0; i < v-1; i++)
    {
        u = vertex(dist, vset);
        vset[u] = 1;


     for (int i = 0; i < v; i++)
     {
         if (vset[i]!=1 && G[u][i] && dist[u]!=999 &&
dist[u]+G[u][i]<dist[i])
         {
             dist[i] = dist[u]+G[u][i];
         }

     }

    }
    display(dist);

}

int main()
{
    int adjMatrix[v][v] = {
        {0, 0, 3, 0, 0},
        {0, 0, 10, 4, 0},
        {3, 10, 0, 2, 6},
        {0, 4, 2, 0, 1},
        {0, 0, 6, 1, 0},
    };

    Dijkstras(adjMatrix, 0);

    return 0;
}
```

Shrihari K Mangle
CS-B2-74 12010517

**OUTPUT :**