NAME: Venkateswar L
BRANCH: Computer Science and Engineering SEC: F
ROLL NO.: 230701376

PROGRAM: Graph

Dijikstras Algorithm

```
#include <stdio.h>
#define INFINITY 9999
#define MAX 10

void Dijkstra(int G[MAX][MAX], int n, int start);

void Dijkstra(int G[MAX][MAX], int n, int start) {
  int cost[MAX][MAX], distance[MAX], pred[MAX];
  int visited[MAX], count, mindistance, nextnode, i, j;

  for (i = 0; i < n; i++)
  for (j = 0; j < n; j++)
     if (G[i][j] == 0)
     cost[i][j] = INFINITY;
     else
       cost[i][j] = G[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) {
    mindistance = INFINITY;

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

```
    if (distance[i] < mindistance && !visited[i]) {
      mindistance = distance[i];
      nextnode = i;
    }
```

```
    visited[nextnode] = 1;
    for (i = 0; i < n; i++)
    if (!visited[i])
        if (mindistance + cost[nextnode][i] < distance[i]) {
          distance[i] = mindistance + cost[nextnode][i];
          pred[i] = nextnode;
        }
    count++;
  }

  // Printing the distance
  for (i = 0; i < n; i++)
        printf("\nDistance from source to %d: %d", i, distance[i]);

}
int main() {
  int G[MAX][MAX], i, j, n, u;
 printf("Enter number of vertices");
 scanf("%d",&n);
 printf("Enter values for adjacency Matrix of Weighted Graph");
 for(i=0;i<n;i++)
 for(j=0;j<n;j++)
 scanf("%d",&G[i][j]);

 u = 0;
 Dijkstra(G, n, u);

  return 0;
}
```

Input:

Enter number of vertices4
Enter values for adjacency Matrix of Weighted Graph0 2 0 1
0 0 0 2
3 0 0 0
0 0 1 0

NAME: Venkateswar L
BRANCH: Computer Science and Engineering SEC: F
ROLL NO.: 230701376

Output:

Distance from source to 0: 0
Distance from source to 1: 2
Distance from source to 2: 2
Distance from source to 3: 1