**NAME – ABHIRAJ YOGESH SRIVASTAVA**

**ROLL NO. – 1906137**

**SUBJECT NAME – DESIGN AND ANALYSIS OF ALGORITHMS LAB**

**SUBJECT CODE – CSL4403**

**DATE – 15TH FEB, 2021**

**BRANCH – CSE 2**

**ASSIGNMENT-8**

**Q8. WAP to implement Single Shortest Path using Dijkstra’s Algorithm.**

**Source Code in C++ Language:**

#include <limits.h>

#include <stdio.h>

#define V 9

int minDistance(int dist[], bool sptSet[])

{

int min = INT\_MAX, min\_index;

for (int v = 0; v < V; v++)

if (sptSet[v] == false && dist[v] <= min)

min = dist[v], min\_index = v;

return min\_index;

}

void printSolution(int dist[])

{

printf("Node\tShortest Distance from the Source(1st Node)\n");

for (int i = 0; i < V; i++)

printf("%d\t\t%d\n",i+1,dist[i]);

}

void dijkstra(int graph[V][V], int src)

{

int dist[V];

bool sptSet[V];

for (int i = 0; i < V; i++)

dist[i] = INT\_MAX, sptSet[i] = false;

dist[src] = 0;

for (int count = 0; count < V - 1; count++)

{

int u = minDistance(dist, sptSet);

sptSet[u] = true;

for (int v = 0; v < V; v++)

{

if (!sptSet[v] && graph[u][v] && dist[u] != INT\_MAX && dist[u] + graph[u][v] < dist[v])

dist[v] = dist[u] + graph[u][v];

}

}

printSolution(dist);

}

int main()

{

int v;

printf("Enter the number of nodes in the graph.\n");

scanf("%d",&v);

int graph[V][V];

printf("Enter the edges and their respective costs in the form of adjancency matrix.\n");

for(int i=0;i<V;i++)

{

for(int j=0;j<V;j++)

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

}

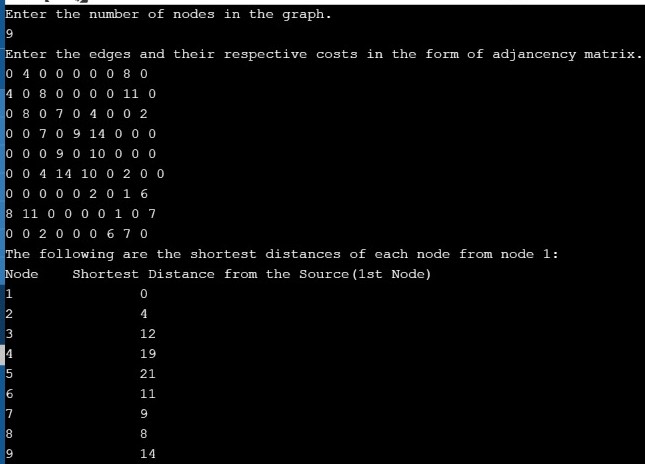
printf("The following are the shortest distances of each node from node 1:\n");

dijkstra(graph,0);

return 0;

}

**Output Screenshot:**

****