**Write a C program to create a graph and find the shortest path using Dijikstra’s Algorithm.**

**Algorithm:**

**CODE:**

**#include <stdio.h>**

**#include <stdlib.h>**

**#include <limits.h>**

**#include <stdbool.h>**

**#define V 5**

**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("Vertex \tDistance from Source\n");**

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

**printf("%d \t%d\n", i, 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 graph[V][V] = {**

**{0, 4, 0, 0, 0},**

**{4, 0, 8, 0, 0},**

**{0, 8, 0, 7, 0},**

**{0, 0, 7, 0, 9},**

**{0, 0, 0, 9, 0}**

**};**

**dijkstra(graph, 0);**

**return 0;**

**}**

OUTPUT:

VERTEX DISTANCE FROM SOURCE

1. 0
2. 4
3. 12
4. 19
5. 21
6. 11
7. 9
8. 8
9. 14