Experiment-23 : Dijkstra’s Algorithm

23.Write a C Program For Implementation Of Shortest Path Algorithms Using Dijkstra’s Algorithm.

Code :

#include <stdio.h>

#include <limits.h>

#define V 6 // number of vertices in the graph

// function to find the minimum distance vertex

int minDistance(int dist[], int visited[]) {

int min = INT\_MAX, min\_index;

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

if (visited[v] == 0 && dist[v] <= min)

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

}

return min\_index;

}

// function to print the shortest distance array

void printSolution(int dist[]) {

printf("Vertex \t\t Distance from Source\n");

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

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

}

// function to implement Dijkstra's algorithm

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

int dist[V]; // distance array

int visited[V]; // visited array

// initialize distance array and visited array

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

dist[i] = INT\_MAX, visited[i] = 0;

dist[src] = 0; // distance of source vertex from itself is 0

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

int u = minDistance(dist, visited);

visited[u] = 1;

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

if (!visited[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, 0},

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

{0, 8, 0, 7, 0, 4},

{0, 0, 7, 0, 9, 14},

{0, 0, 0, 9, 0, 10},

{0, 0, 4, 14, 10, 0}

};

int src;

printf("Enter the source vertex (0-5): ");

scanf("%d", &src);

dijkstra(graph, src);

return 0;

}

Output :

