PROGRAM 23:

DIJIKSTRA’S ALGORITHM:

PROGRAM:

#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;

}

int printSolution(int dist[], int n) {

printf("Vertex Distance 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, V);

}

int main() {

int graph[V][V] = { { 0, 6, 0, 0, 0, 0, 0, 8, 0 },

{ 6, 0, 8, 0, 0, 0, 0, 13, 0 },

{ 0, 8, 0, 7, 0, 6, 0, 0, 2 },

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

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

{ 0, 0, 6, 14, 10, 0, 2, 0, 0 },

{ 0, 0, 0, 0, 0, 2, 0, 1, 6 },

{ 8, 13, 0, 0, 0, 0, 1, 0, 7 },

{ 0, 0, 2, 0, 0, 0, 6, 7, 0 }

};

dijkstra(graph, 0);

return 0;

}

OUTPUT:

