**Звіт**

про виконання лабораторної роботи № 3

Студента групи ІН-305Б

**Дорошенка Мар’яна Олександровича**

**Завдання:**

Написати програму, яка реалізує алгоритм Дейкстри. Відкомпілювати та

відлагодити програму. Протестувати роботу програми використовуючи

різні вхідні дані: За початкову точку вибрати вершину 3.

**Код програми:**

#include <iostream>

#include <vector>

#include <climits>

const int V = 9;

int minDistance(const std::vector<long long>& dist, const std::vector<bool>& sptSet) {

long long minDist = LLONG\_MAX;

int minIndex = -1;

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

if (!sptSet[v] && dist[v] < minDist) {

minDist = dist[v];

minIndex = v;

}

}

return minIndex;

}

void printSolution(const std::vector<long long>& dist) {

std::cout << "Вершина \t Відстань від початкової вершини\n";

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

std::cout << i << "\t\t" << dist[i] << std::endl;

}

}

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

std::vector<long long> dist(V, LLONG\_MAX);

std::vector<bool> sptSet(V, 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] != LLONG\_MAX && dist[u] + graph[u][v] < dist[v]) {

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

}

}

}

printSolution(dist);

}

int main() {

int graph[V][V] = {

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

{9, 0, 0, 0, 8, 0, 0, 0, 0},

{0, 0, 0, 0, 0, 0, 0, 0, 0},

{14, 0, 0, 0, 0, 0, 0, 6, 0},

{0, 8, 0, 0, 0, 10, 0, 0, 0},

{0, 0, 0, 0, 10, 0, 2, 0, 0},

{0, 0, 0, 0, 0, 2, 0, 9, 0},

{0, 0, 0, 6, 0, 0, 9, 0, 8},

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

};

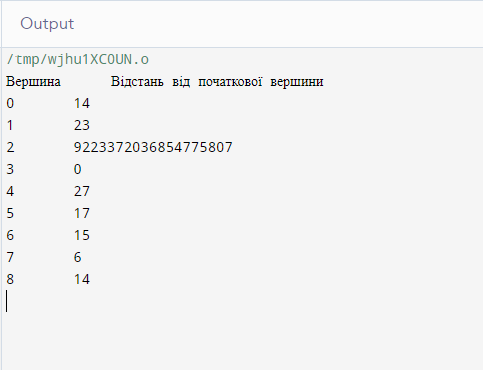
int startVertex = 3;

dijkstra(graph, startVertex);

return 0;

}

**Результат:**

****