Министерство науки и высшего образования Российской Федерации  
Федеральное государственное автономное учреждение высшего образования  
«Пермский национальный исследовательский политехнический университет»

Электротехнический факультет  
Кафедра «Информационные технологии и автоматизированные системы»

**О Т Ч Ё Т  
по лабораторной работе**

Дисциплина: «Основы алгоритмизации и программирования»

Выполнил  
студент группы ИВТ-21-1б  
Мифтахов И.Г

Проверил  
ст. п. кафедры ИТАС  
Яруллин Д.В.

Пермь 2022

Постановка задачи:

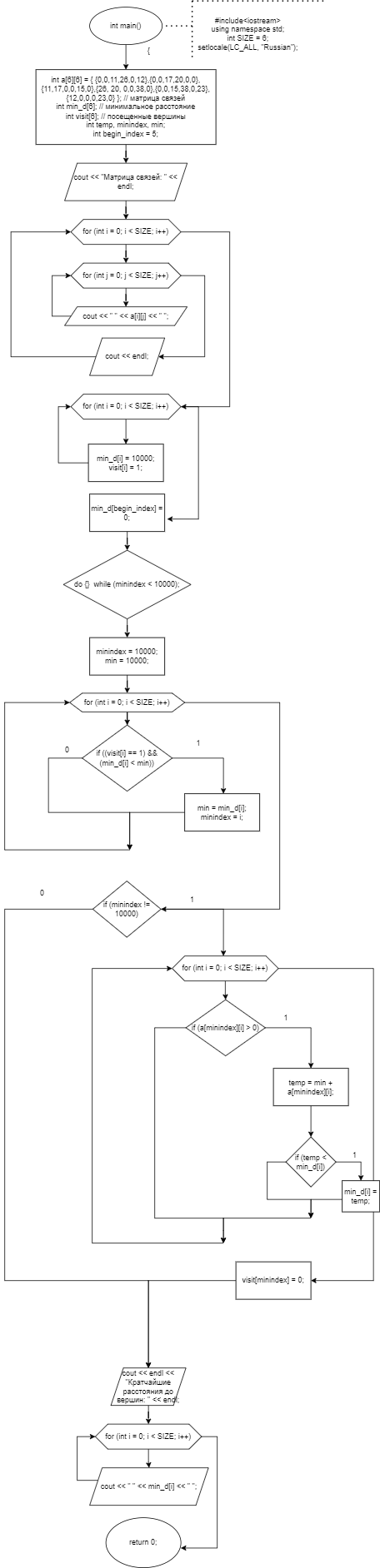
Реализовать граф, а также алгоритм Дейкстры, выполнив все необходимые действия.

Выполнение начать с вершины 6.

Анализ задачи:

«Вручную» составил таблицу(матрицу) смежности, заполнил ее. Указал начальный индекс – 5(массив, индексы с 0). Далее реализовал самый обычный алгоритм Дейкстры. Для вывода квадратной матрицы использовал вложенный цикл for. Для вывода массива кратчайших путей использовал простой цикл for.

Блок-схема:



Код программы:

#include <iostream>

int SIZE = 6;

using namespace std;

int main()

{

setlocale(LC\_ALL, "rus");

int a[6][6] = { {0,0,11,26,0,12},{0,0,17,20,0,0},{11,17,0,0,15,0},{26, 20, 0,0,38,0},{0,0,15,38,0,23},{12,0,0,0,23,0} }; // матрица связей

int min\_d[6]; // минимальное расстояние

int visit[6]; // посещенные вершины

int temp, minindex, min;

int begin\_index = 5;

cout << "Матрица связей: " << endl;

// Вывод матрицы связей

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

{

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

{

cout << " " << a[i][j] << " ";

}

cout << endl;

}

//Инициализация вершин и расстояний

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

{

min\_d[i] = 10000;

visit[i] = 1;

}

min\_d[begin\_index] = 0;

// Шаг алгоритма

do {

minindex = 10000;

min = 10000;

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

{ // Если вершину ещё не обошли и вес меньше min

if ((visit[i] == 1) && (min\_d[i] < min))

{ // Переприсваиваем значения

min = min\_d[i];

minindex = i;

}

}

// Добавляем найденный минимальный вес

// к текущему весу вершины

// и сравниваем с текущим минимальным весом вершины

if (minindex != 10000)

{

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

{

if (a[minindex][i] > 0)

{

temp = min + a[minindex][i];

if (temp < min\_d[i])

{

min\_d[i] = temp;

}

}

}

visit[minindex] = 0;

}

} while (minindex < 10000);

// Вывод кратчайших расстояний до вершин

cout << endl << "Кратчайшие расстояния до вершин: " << endl;

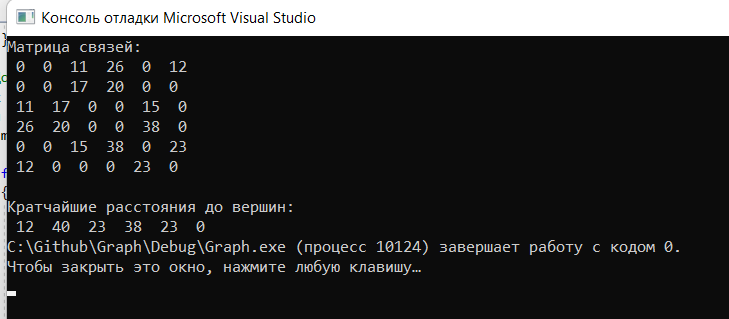
for (int i = 0; i < SIZE; i++)

cout << " " << min\_d[i] << " ";

return 0;

}

Скриншоты результатов:



Анализ результатов:

Программа работает корректно. Выводятся кратчайшие расстояния от вершины 6 до всех остальных.