Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)

Кафедра автоматизированных систем управления (АСУ)

**Графы**

**Отчет по лабораторной работе №8**

**По дисциплине**

**«Структуры и алгоритмы обработки данных в ЭВМ»**

Выполнил: студент гр. 439-1

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Зозуля Е.Д.

«\_\_\_»\_\_\_\_\_\_\_\_\_\_\_ 2021 г.

Проверил: ассистент каф. АСУ

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Яблонский Я.В.

«\_\_\_»\_\_\_\_\_\_\_\_\_\_\_ 2021 г.

Томск 2021

# Задание на лабораторную работу

Вариант 2.

# Напишите программу, которая с помощью алгоритма Флойда будет находить кратчайшие пути между всеми парами вершин графа. Граф задан списками смежности. Предусмотрите ввод данных из файла. После завершения работы с динамическими структурами данных необходимо освободить занимаемую ими память.

# Алгоритм решения задачи

Cчитываются списки смежности вместе с весами рёбер, затем запускается алгоритм Флойда

# Листинг программы

Lab8.cpp

#include <iostream>

#include <fstream>

using namespace std;

const int maxV = 1000;

int GR[maxV][maxV];

//алгоритм Флойда-Уоршелла

void Floyd(int D[][maxV], int V)

{

int k;

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

D[i][i] = 0;

for (k = 0; k < V; k++)

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

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

if (D[i][k] && D[k][j] && i != j)

if (D[i][k] + D[k][j] < D[i][j] || D[i][j] == 0)

D[i][j] = D[i][k] + D[k][j];

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

{

for (int j = 0; j < V; j++) cout << D[i][j] << "\t";

cout << endl;

}

}

//главная функция

int main()

{

setlocale(LC\_ALL, "Rus");

int ch;

cout << "Выберите способ ввода\n1.Считать с файла\n2.Ввести вручную\n";

cin >> ch;

if (ch == 1)

{

ifstream Filein("graphFloyd.txt");

Filein >> ch;

while (Filein)

{

int i, j;

Filein >> i >> j;

Filein >> GR[i][j];

cout << "GR[" << i + 1 << "][" << j + 1 << "] > " << GR[i][j]<<endl;

}

cout << "Матрица кратчайших путей:" << endl;

Floyd(GR, ch);

}

else if (ch == 2)

{

cout << "Количество вершин в графе > "; cin >> ch;

cout << "Введите матрицу весов ребер:\n";

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

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

{

cout << "GR[" << i + 1 << "][" << j + 1 << "] > ";

cin >> GR[i][j];

}

cout << "Матрица кратчайших путей:" << endl;

Floyd(GR, ch);

}

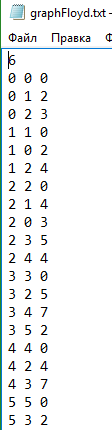
system("pause");

return 0;

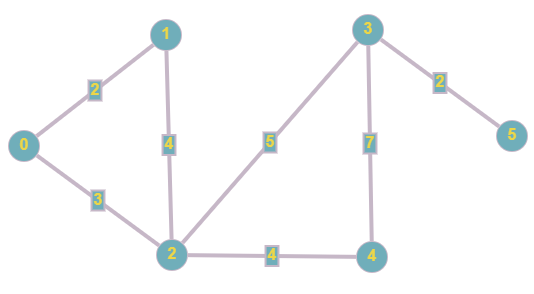
}

# Пример решения

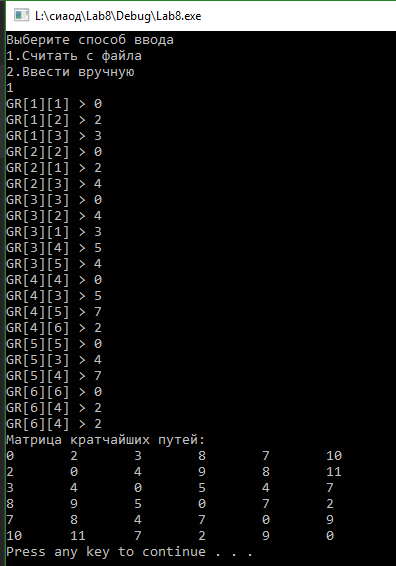
Скриншоты успешной работы программы, входные данные тестового примера, выходные данные на рисунке 4.1, 4.2 и 4.3.



*Рисунок 4.1.- Входные данные*



*Рисунок 4.3.- граф, построенный по спискам смежности*



*Рисунок 4.2.- Скриншот успешной работы программы*

# Вывод

Был изучен и использован алгоритм Флойда для построения матрицы кратчайших путей