**Министерство науки и высшего образования Российской Федерации**

**федеральное государственное бюджетное образовательное учреждение высшего образования**

**Московский авиационный институт**

**(Национальный исследовательский университет)**

Факультет №3

«Системы управления, информатика и электроэнергетика»

Кафедра 304

**Отчет по лабораторной работе №5**

по учебной дисциплине «Информатика»

на тему «Двумерные массивы»

Вариант № 15

Руководитель работы

\_\_\_\_\_\_\_\_\_\_\_\_ Давыдкина Е. А.

«\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_2022 г.

\_\_\_\_\_\_\_\_\_\_\_\_ Секретарев В. Е.

«\_\_\_\_»\_\_\_\_\_\_\_\_\_\_ 2022 г.

Исполнитель

обучающийся группы М30-110Б-21

\_\_\_\_\_\_\_\_\_\_\_\_ Осминнов Н. М.

«\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_ 2022 г.

\_\_\_\_\_\_\_\_\_\_\_\_Пысларь А. И

«\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_ 2022 г.

Москва 2022

**Содержание.**

1. Задание……………………………………………….......................................................................3
2. Блок-схема………………………………………………………………………………………………………………4
3. Псевдокод……………………………………………………………………………………………………………….12
4. Код программы……………………………………………………………………………………………………….16
5. Тесты программы…………………………………………………………………………………………………...20
6. Вывод……………………………………………………………………………………………………………………….22

**Задание.**

Кафедра 304 Курс: ИНФОРМАТИКА II семестр

Задание 4: **Двумерные массивы**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**ВАРИАНТ № 15**

Дана целочисленная прямоугольная матрица. Определить:

1. Сумму элементов матрицы по строкам.

2. Максимальное значение суммы

Используя универсальные для различных наборов исходных данных подпрограммы, реализовать данный алгоритм для заданных матриц: X1(N1, M1), X2(N2, M2).

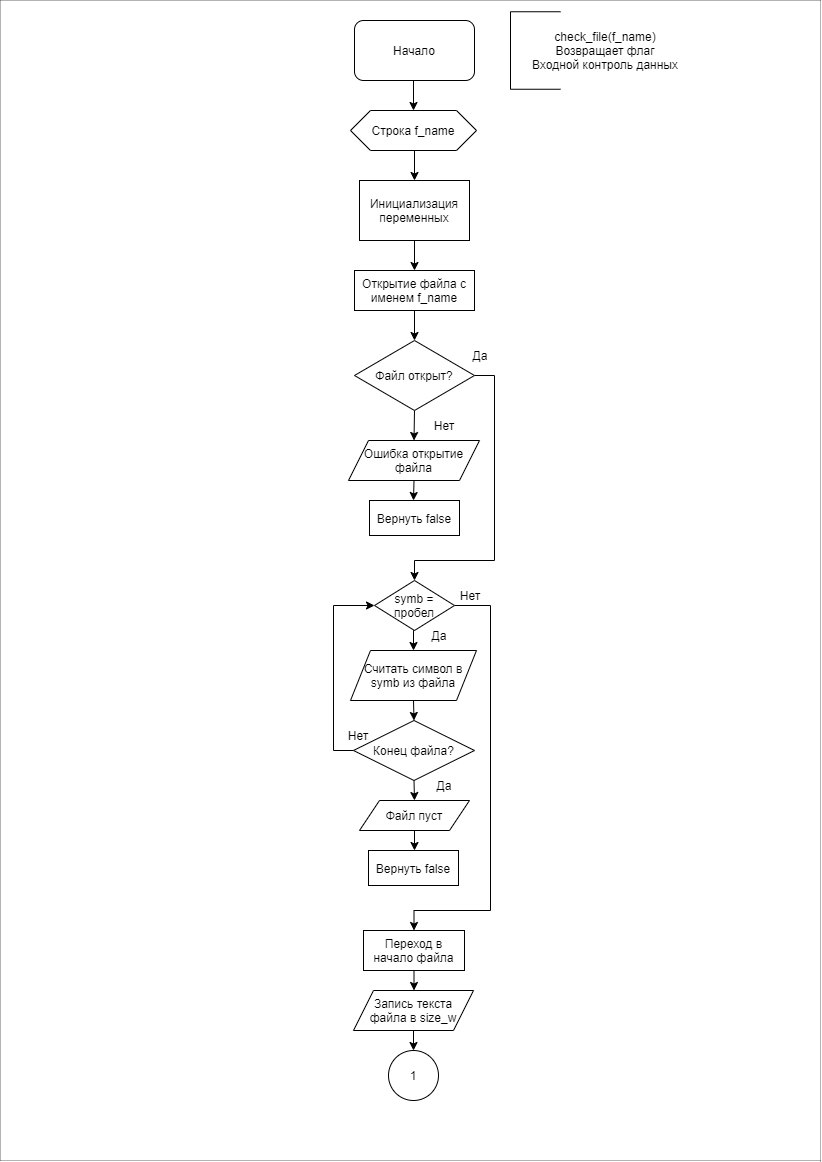
В качестве одного из вариантов исходных данных принять: N1= 5, MI = 8; N2 = 6, M2 = 9.

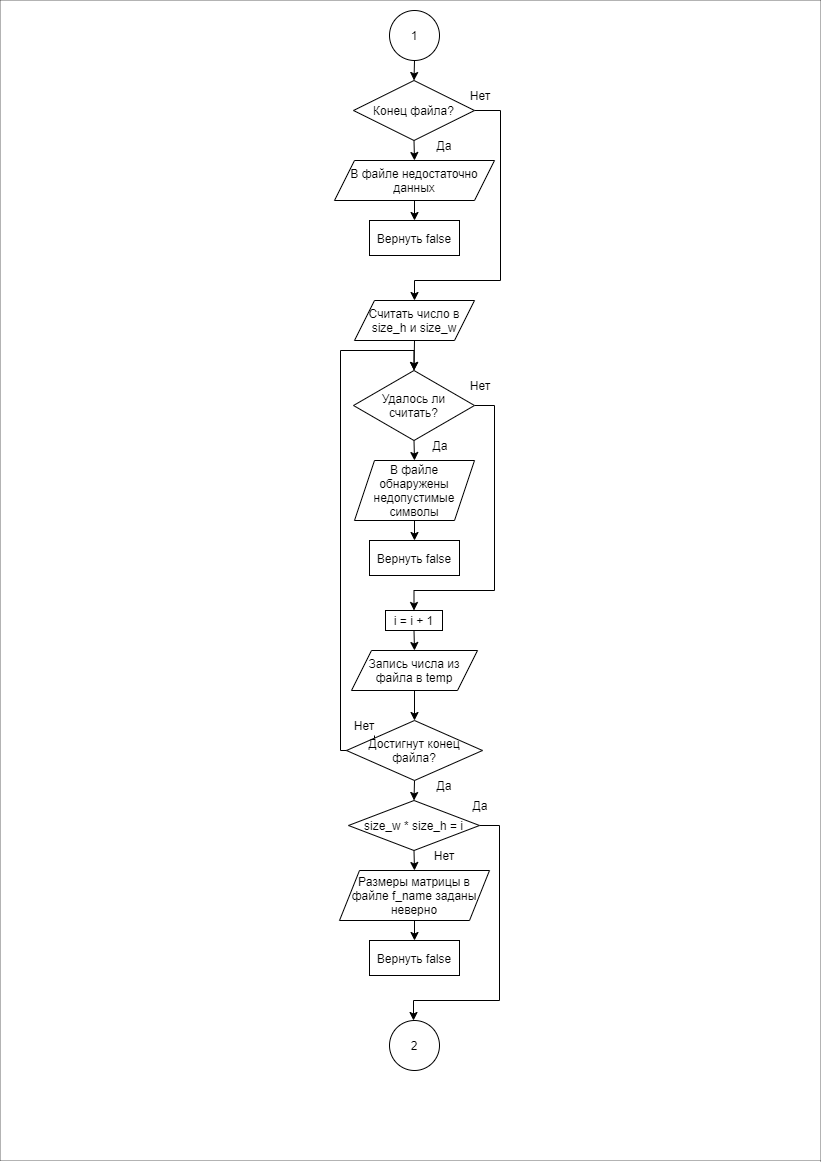
Чтение данных их файла производить с использованием функций ввода/вывода языка С++.

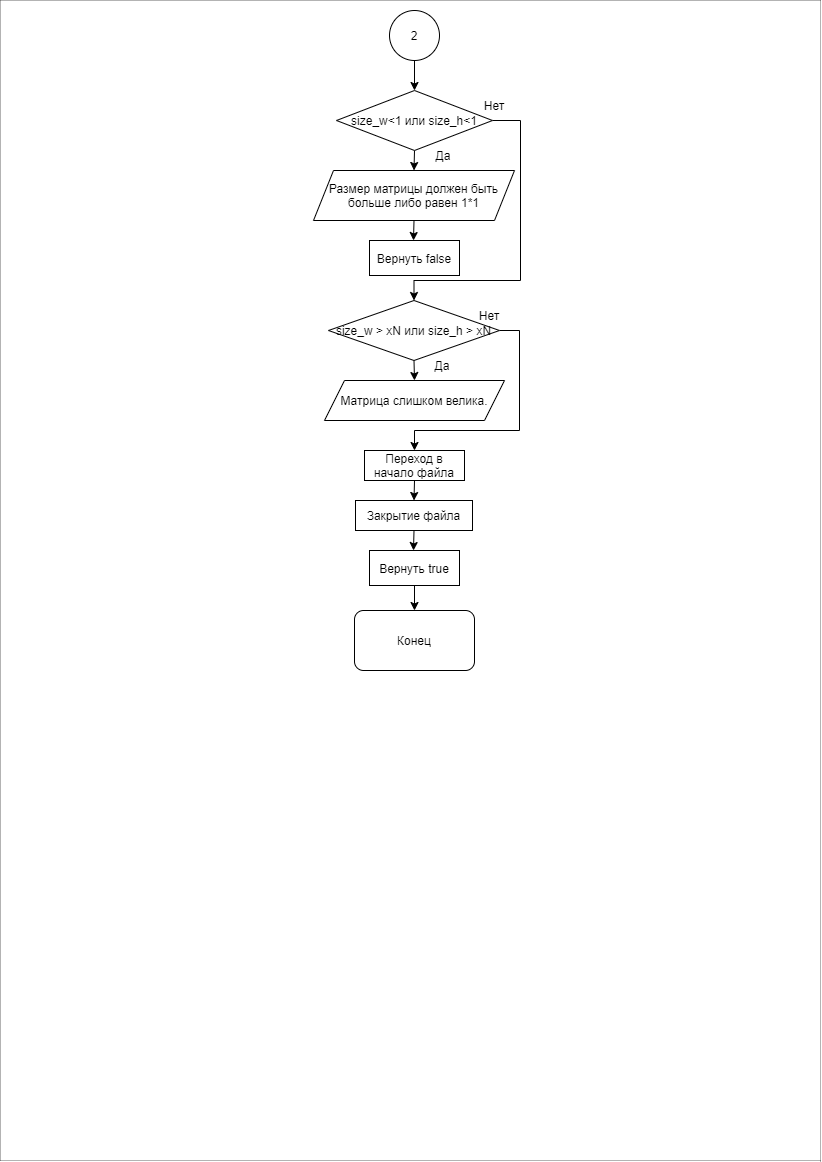
Алгоритм должен быть параметризован; обмен данными с подпрограммой должен осуществляться только через параметры; каждый из наборов исходных данных хранится в отдельном файле.

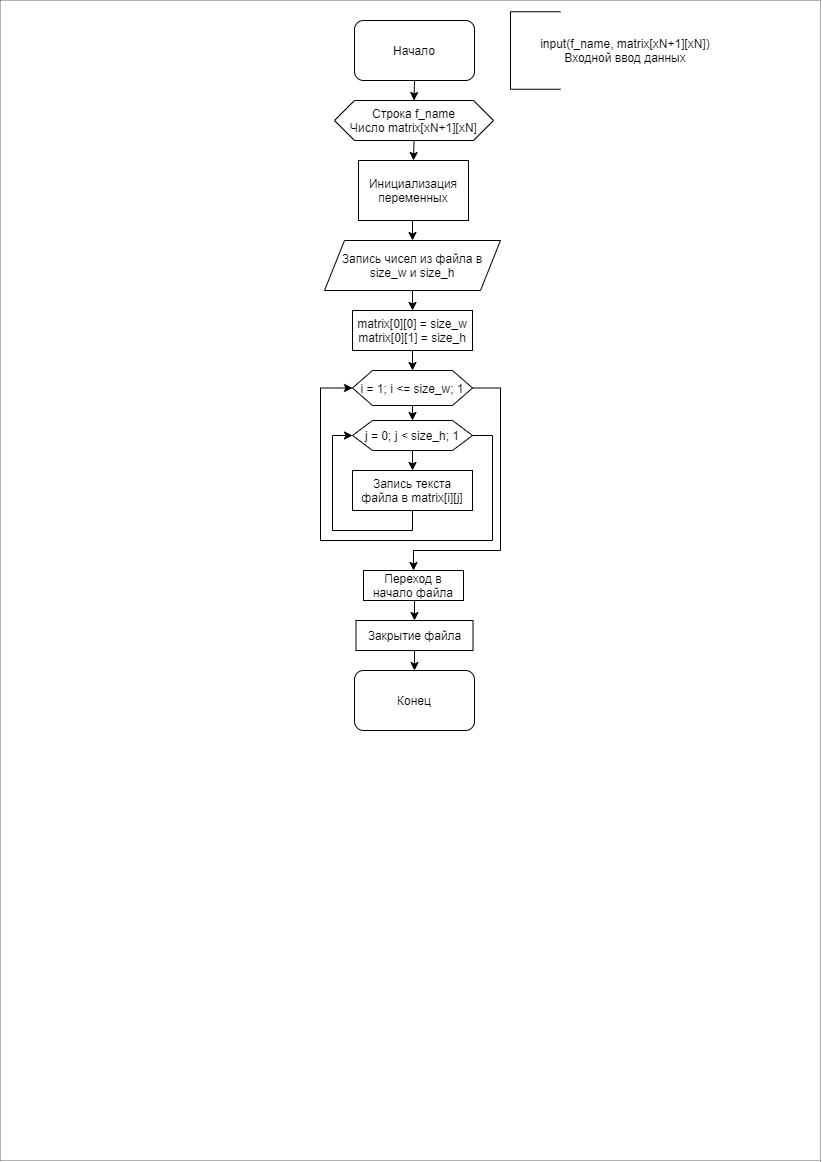
Реализовать программу в двух вариантах: в первом - при обращении к элементам массива использовать индексы, во втором - работать с динамическим массивом через указатели.

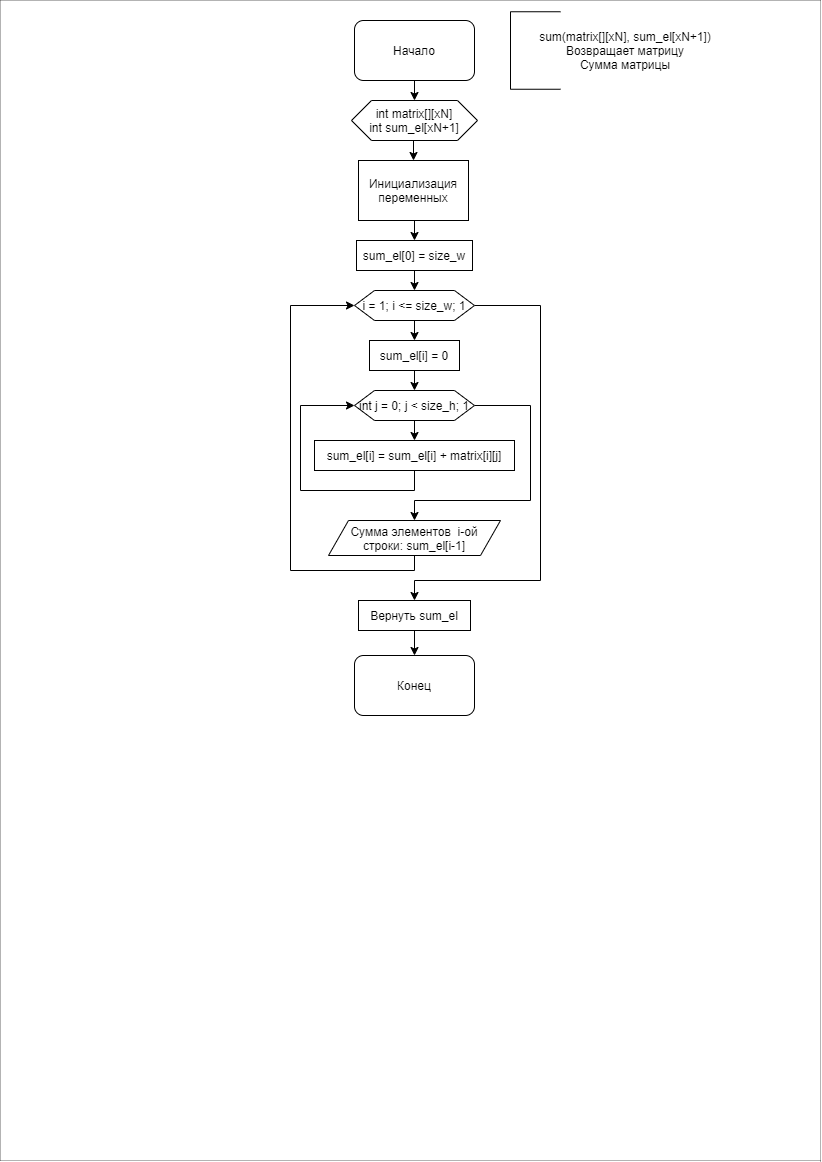
**Блок-схема.**

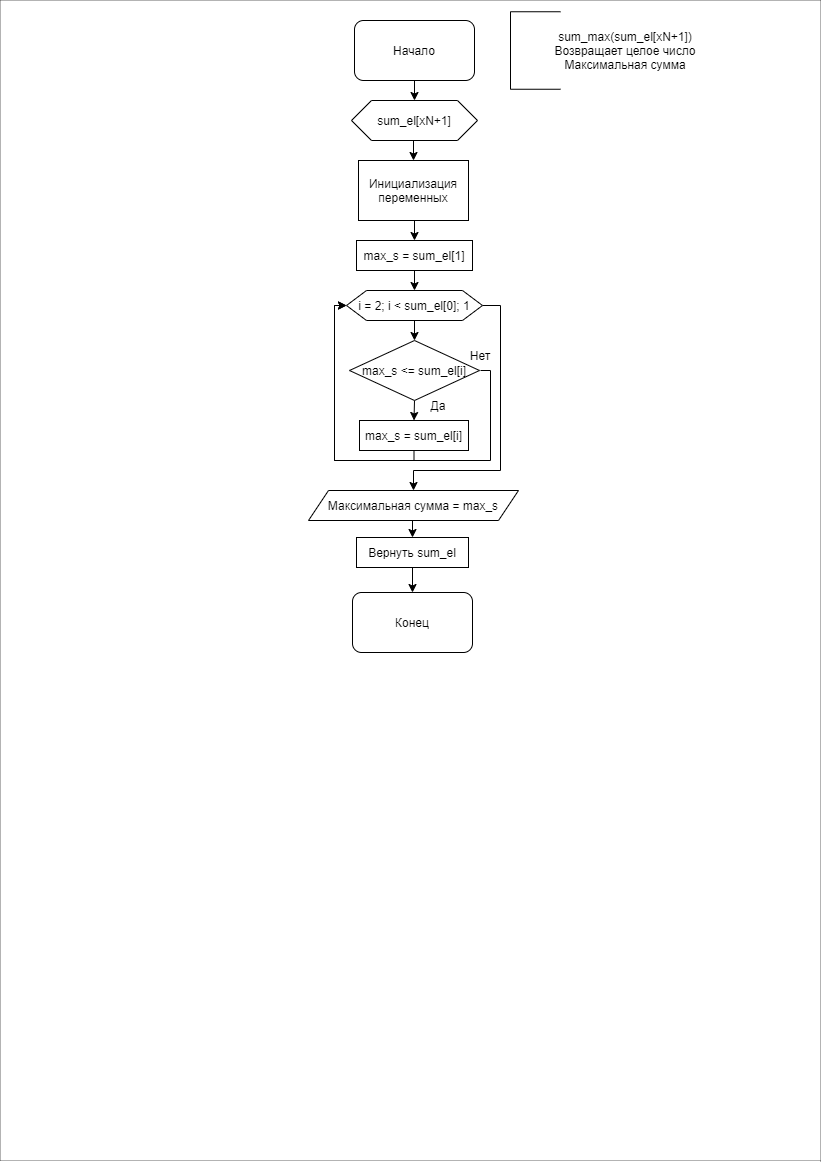


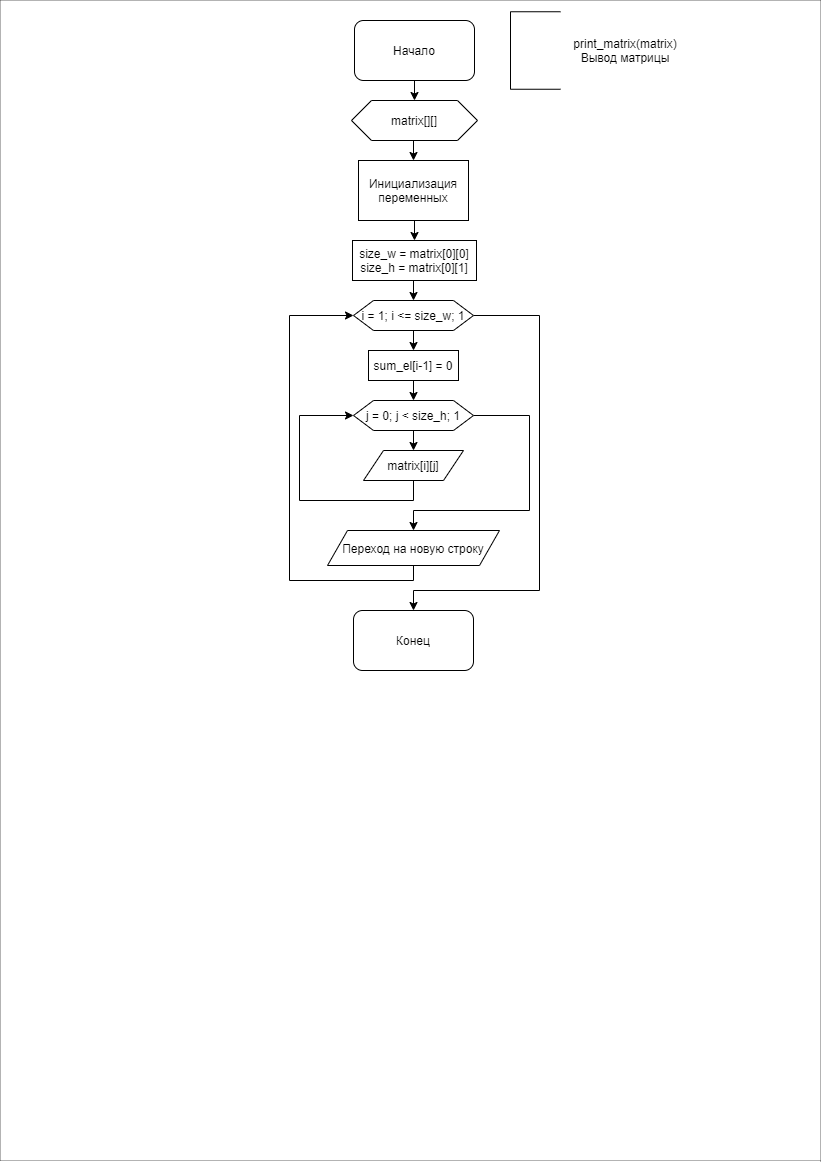


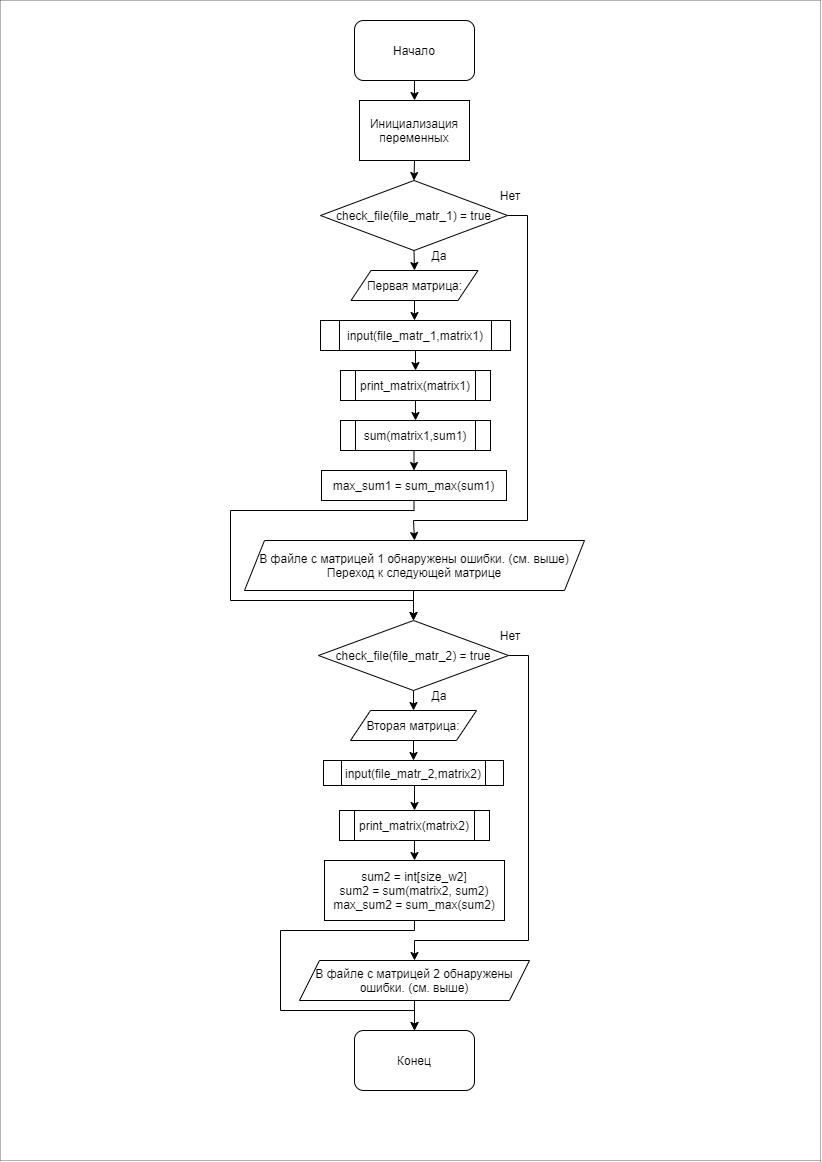












**Псевдокод.**

ФУНКЦИЯ check\_file( строка f\_name )

|

| ЦЕЛОЧИСЛЕННЫЕ:

| size\_w

| size\_h

| i = 0

| temp

|

| СИМВОЛЫ:

| symb = пробел

|

| ОТКРЫТЬ ФАЙЛ f\_name

| ЕСЛИ файл f\_name НЕ ОТКРЫТ

| ТО

| | ВЕРНУТЬ ЗНАЧЕНИЕ ЛОЖЬ

| КОНЕЦ

|

| ПОКА symb = пробелу

| | СЧИТАТЬ СИМВОЛ ИЗ ФАЙЛА f\_name В symb

| | ЕСЛИ ФАЙЛ f\_name ЗАКОНЧИЛСЯ

| | ТО

| | | ВЕРНУТЬ ЗНАЧЕНИЕ ЛОЖЬ

| | КОНЕЦ

| КОНЕЦ

|

| ПЕРЕМЕСТИТЬ курсор в начало файла

|

| СЧИТАТЬ С ФАЙЛА f\_name переменную size\_w

| ЕСЛИ ФАЙЛ f\_name ЗАКОНЧИЛСЯ

| ТО

| | ВЕРНУТЬ ЗНАЧЕНИЕ ЛОЖЬ

| КОНЕЦ

|

| СЧИТАТЬ С ФАЙЛА f\_name переменную size\_h

| ДЕЛАТЬ

| | ЕСЛИ СЧИТАТЬ НЕ УДАЛОСЬ

| | ТО

| | | ВЕРНУТЬ ЗНАЧЕНИЕ ЛОЖЬ

| | КОНЕЦ

| | i = i + 1

| | ВВЕСТИ С ФАЙЛА f\_name число в temp

| ПОКА НЕ ДОСТИГНУТ КОНЕЦ ФАЙЛА f\_name

|

| ЕСЛИ i НЕ РАВЕН size\_w \* size\_h

| ТО

| | ВЕРНУТЬ ЗНАЧЕНИЕ ЛОЖЬ

| КОНЕЦ

|

| ЕСЛИ size\_w < 1 ИЛИ size\_h < 1

| ТО

| | ВЕРНУТЬ ЗНАЧЕНИЕ ЛОЖЬ

| КОНЕЦ

|

| ЕСЛИ size\_w > xN ИЛИ size\_h > xN

| ТО

| | ВЕРНУТЬ ЗНАЧЕНИЕ ЛОЖЬ

| КОНЕЦ

|

| ЗАКРЫТЬ ФАЙЛ f\_name

| ВЕРНУТЬ ЗНАЧЕНИЕ ЛОЖЬ

КОНЕЦ ПОДПРОГРАММЫ

ПОДПРОГРАММА input( строка f\_name, ЦЕЛОЧИСЛЕННЫЙ ДВУМЕРНЫЙ МАССИВ matrix[xN+1][xN] )

| ЦЕЛОЧИСЛЕННЫЕ:

| size\_w

| size\_h

|

|

| ОТКРЫТЬ ФАЙЛ f\_name

| СЧИТАТЬ ЧИСЛО ИЗ ФАЙЛА f\_name В size\_w

| СЧИТАТЬ ЧИСЛО ИЗ ФАЙЛА f\_name В size\_h

| ЗАДАТЬ КОЛИЧЕСТВО ОДНОМЕРНЫХ МАССИВОВ size\_w + 1 В ДВУМЕРНОМ МАССИВЕ matrix

| ЗАДАТЬ КОЛИЧЕСТВО ЦЕЛОЧИСЛЕННЫХ ЭЛЕМЕНТОВ 2 в МАССИВЕ matrix[0]

|

| ДЛЯ i = 1 ПОКА i<= size\_w ДЕЛАТЬ i++

| | ЗАДАТЬ КОЛИЧЕСТВО ЦЕЛОЧИСЛЕННЫХ ЭЛЕМЕНТОВ size\_h в МАССИВЕ matrix[i]

| КОНЕЦ

|

| matrix[0][0] = size\_w

| matrix[0][1] = size\_h

|

| ДЛЯ i = 1 ПОКА i<=size\_w ДЕЛАТЬ i++

| | ДЛЯ j = 0 ПОКА j < size\_h ДЕЛАТЬ j++

| | | СЧИТАТЬ ЧИСЛО ИЗ ФАЙЛА f\_name В matrix[i][j]

| | КОНЕЦ

| КОНЕЦ

| ЗАКРЫТЬ ФАЙЛ f\_name

КОНЕЦ ПОДПРОГРАММЫ

ФУНКЦИЯ sum( ДВУМЕРНЫЙ МАССИВ matrix, ЦЕЛОЧИСЛЕННЫЙ МАССИВ sum\_el[xN+1] )

|

| ЦЕЛОЧИСЛЕННЫЕ:

| size\_w = matrix[0][0]

| size\_h = matrix[0][1]

|

| sum\_el[0] = size\_w

|

| ЗАДАТЬ РАЗМЕР size\_w ДЛЯ МАССИВА sum\_el

| ДЛЯ i = 1 ПОКА i<=size\_w ДЕЛАТЬ i++

| | sum\_el[i] = 0

| | ДЛЯ j = 0 ПОКА j < size\_h ДЕЛАТЬ j++

| | | sum\_el[i] = sum\_el[i] + matrix[i][j]

| | КОНЕЦ

| | ПЕЧАТЬ В КОНСОЛЬ sum\_el[i]

| КОНЕЦ

| ВЕРНУТЬ МАССИВ sum\_el

КОНЕЦ ПОДПРОГРАММЫ

ФУНКЦИЯ sum\_max (одномерный массив sum\_el)

| ЦЕЛОЧИСЛЕННЫЕ

| max\_s = sum\_el[1]

|

| ДЛЯ i = 0 ПОКА i < sum\_el[0] ДЕЛАТЬ i = i + 1

| | ЕСЛИ max\_s <= sum\_el[i]

| | | max\_s = sum\_el[i]

| | КОНЕЦ

| КОНЕЦ

|

| ПЕЧАТЬ В КОНСОЛЬ max\_s

| ВЕРНУТЬ ЗНАЧЕНИЕ max\_s

КОНЕЦ ПОДПРОГРАММЫ

ПОДПРОГРАММА print\_matrix( ЦЕЛОЧИСЛЕННЫ ДВУМЕРНЫЙ МАССИВ matrix[][xN])

| ЦЕЛОЧИСЛЕННЫЕ

| size\_w = matrix[0][0]

| size\_h = matrix[0][1]

| ДЛЯ i = 1 ПОКА i<=size\_w ДЕЛАТЬ i++

| | ДЛЯ j = 0 ПОКА j < size\_h ДЕЛАТЬ j++

| | | ПЕЧАТЬ В КОНСОЛЬ matrix[i][j]

| | КОНЕЦ

| | ПЕЧАТЬ В КОНСОЛЬ ПЕРЕХОДА НА СЛЕДУЮЩУЮ СТРОКУ

| КОНЕЦ

КОНЕЦ ПОДПРОГРАММЫ

ОСНОВНАЯ ФУНКЦИЯ main() // Сама программа

| ДВУМЕРНЫЕ МАТРИЦЫ:

| matrix\_1[xN+1][xN]

| matrix\_2[xN+1][xN]

|

| ЦЕЛОЧИСЛЕННЫЕ:

| size\_w1

| size\_h1

|

| size\_w2

| size\_h2

|

| СТРОКИ

| file\_matr\_1

| file\_matr\_2

|

| ЕСЛИ check\_file(ИМЯ ФАЙЛА file\_matr\_1) - ИСТИНА

| ТО

| | matrix1 = input( ИМЯ ФАЙЛА file\_matr\_1, ЦЕЛОЧИСЛЕННЫЙ ДВУМЕРНЫЙ МАССИВ matrix 1)

| |

| | print\_matrix(ЦЕЛОЧИСЛЕННЫЙ ДВУМЕРНЫЙ МАССИВ matrix1)

| |

| | ЦЕЛОЧИСЛЕННЫЙ МАССИВ sum1[xN]

| | ЦЕЛ max\_sum1

| |

| | sum1 = sum(ДВУМЕРНЫЙ ЦЕЛОЧИСЛЕННЫЙ МАССИВ matrix1, ЦЕЛОЧИСЛЕННЫЙ МАССИВ sum1)

| | max\_sum1 = sum\_max(ЦЕЛОЧИСЛЕННЫЙ МАССИВ sum1)

| |

| ИНАЧЕ

| | ПЕЧАТЬ В КОНСОЛЬ СООБЩЕНИЯ ОБ ОШИБКЕ

| КОНЕЦ

|

| ЕСЛИ check\_file(ИМЯ ФАЙЛА file\_matr\_2) - ИСТИНА

| ТО

| | matrix1 = input( ИМЯ ФАЙЛА file\_matr\_2, ЦЕЛОЧИСЛЕННЫЙ ДВУМЕРНЫЙ МАССИВ matrix 2)

| |

| | print\_matrix(ЦЕЛОЧИСЛЕННЫЙ ДВУМЕРНЫЙ МАССИВ matrix2)

| |

| | ЦЕЛОЧИСЛЕННЫЙ МАССИВ sum2[xN]

| | ЦЕЛ max\_sum2

| |

| | sum2 = sum(ДВУМЕРНЫЙ ЦЕЛОЧИСЛЕННЫЙ МАССИВ matrix2, ЦЕЛОЧИСЛЕННЫЙ МАССИВ sum2)

| | max\_sum2 = sum\_max(ЦЕЛОЧИСЛЕННЫЙ МАССИВ sum2)

| |

| ИНАЧЕ

| | ПЕЧАТЬ В КОНСОЛЬ СООБЩЕНИЯ ОБ ОШИБКЕ

| КОНЕЦ

|

КОНЕЦ ПРОГРАММЫ

**Код Программы.**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Курс ИНФОРМАТИКА \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*Project type :Win64 Console Application \*

\*Project name :Lab2 (2 sem) \*

\*File name :Лабораторная работа.exe \*

\*Language :CPP, MSVS 2021 \*

\*Programmers :Осминнов Никита Михайлович, Пысларь Александр Игоревич, М3О-110Б-21 \*

\*Modified By : \*

\*Created :18.03.2022 \*

\*Last revision:01.04.2022 \*

\*Comment :Двумерные массивы \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <iostream>

#include <fstream>

const int xN = 50;

using namespace std;

bool check\_file(string f\_name) {

    int size\_w, size\_h;

    int i = 0;

    int temp;

    char symb = ' ';

    ifstream ifs(f\_name);

    if (!ifs.is\_open()) {

        cout << "\nОшибка открытия файла \'" << f\_name << "\'.\n";

        return false;

    }

    while (symb == ' ') {

        symb = ifs.get();

        if (ifs.eof()) {

            cout << "\nФайл пуст.\n";

            return false;

        }

    }

    ifs.clear();

    ifs.seekg(0);

    ifs >> size\_w;

    if (ifs.eof()) {

        cout << "\nВ файле недостаточно данных.\n";

        return false;

    }

    ifs >> size\_h;

    do {

        if (ifs.fail()) {

            cout << "\nВ файле обнаружены недопустимые символы.\n";

            return false;

        }

        i++;

        ifs >> temp;

    } while (!ifs.eof());

    if (i != size\_w \* size\_h) {

        cout << "\nРазмеры матрицы в файле \'" << f\_name << "\' заданы неверно\n";

        return false;

    }

    if ((size\_w < 1) || (size\_h < 1)) {

        cout << "Размер матрицы должен быть больше либо равен 1\*1\n";

        return false;

    }

    if ((size\_w > xN) || (size\_h > xN)) {

        cout << "Матрица слишком велика.\n";

        return false;

    }

    ifs.clear();

    ifs.close();

    return true;

}

void input(string f\_name, int matrix[xN+1][xN]) {

    int size\_w, size\_h;

    ifstream ifs(f\_name);

    ifs >> size\_w;

    ifs >> size\_h;

    matrix[0][0] = size\_w;

    matrix[0][1] = size\_h;

    for (int i = 1; i <= size\_w; i++) {

        for (int j = 0; j < size\_h; j++) {

            ifs >> matrix[i][j];

            //cout << matrix[i][j] << ' ';

        }

        //cout << endl;

    }

    ifs.clear();

    ifs.close();

}

int\* sum(int matrix[][xN],int sum\_el[xN+1]) {

    int size\_w = matrix[0][0];

    int size\_h = matrix[0][1];

    sum\_el[0] = size\_w;

    for (int i = 1; i <= size\_w; i++) {

        sum\_el[i] = 0;

        for (int j = 0; j < size\_h; j++) {

            sum\_el[i] += matrix[i][j];

        }

        cout << "Сумма элементов " << i << "-ой строки:" << sum\_el[i - 1] << endl;

    }

    cout << endl;

    return sum\_el;

}

int sum\_max(int sum\_el[xN+1]) {

    int max\_s = sum\_el[1];

    for (int i = 0; i < sum\_el[0]; i++) {

        if (max\_s <= sum\_el[i]) {

            max\_s = sum\_el[i];

        }

    }

    cout << "Максимальная сумма = " << max\_s << endl;

    return max\_s;

}

void print\_matrix(int matrix[][xN]) {

    int size\_w = matrix[0][0];

    int size\_h = matrix[0][1];

    for (int i = 1; i <= size\_w; i++) {

        for (int j = 0; j < size\_h; j++) {

            cout << matrix[i][j];

        }

        cout << endl;

    }

    cout << endl;

}

int main()

{

    setlocale(LC\_ALL, "Russian");

    int matrix1[xN+1][xN];

    int matrix2[xN+1][xN];

    int err\_code = 0;

    int size\_w1 = 0;

    int size\_h1 = 0;

    int size\_w2 = 0;

    int size\_h2 = 0;

    const char file\_matr\_1[] = "test\_01(1).txt";

    const char file\_matr\_2[] = "test\_01(2).txt";

    if (check\_file(file\_matr\_1)) {

        cout << "Первая матрица:" << endl;    //Начало работы с перво матрицей

        input(file\_matr\_1,matrix1);         //Считываем матрицу 1

        print\_matrix(matrix1);                //Печать матрицы

        int sum1[xN];

        int max\_sum1;

        sum(matrix1,sum1);                  //Суммирование элементов строк матрицы

        max\_sum1 = sum\_max(sum1);             //Поиск максимального значения

    }

    else {

        cout << "\nВ файле с матрицей 1 обнаружены ошибки. (см. выше) \nПереход к следующей матрице\n\n\n";

    }

    if (check\_file(file\_matr\_2)) {

        cout << "Вторая матрица:" << endl;

        input(file\_matr\_2,matrix2);   //Считываем матрицу 2

        print\_matrix(matrix2);

        int max\_sum2;

        int\* sum2;

        sum2 = new int[size\_w2];

        sum2 = sum(matrix2, sum2);

        max\_sum2 = sum\_max(sum2);

    }

    else {

        cout << "В файле с матрицей 2 обнаружены ошибки. (см. выше)\n\n";

    }

    return 0;

}

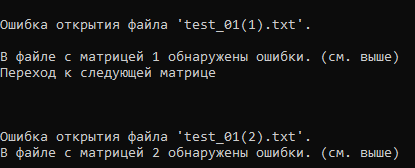
**Тесты программы.**

* + - 1. **Некорректные тесты**

**1.1**  Цель: проверить работоспособность программы при отсутствии файлов.

Ожидаемый результат: Ошибка открытия файла

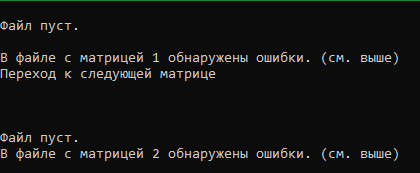
Полученный результат:



1.2 Цель: проверить работоспособность программы при пустых файлах.

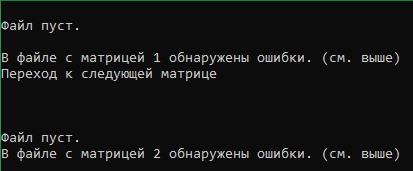
Ожидаемый результат: Файл пуст.

Полученный результат:



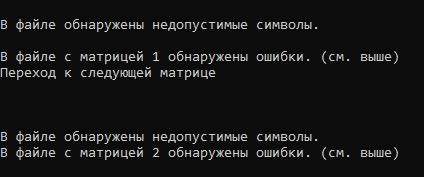
1.3 Цель: проверить работоспособность программы при работе с файлами, в которых нет матриц, но есть пробелы и переносы строк. Ожидаемый результат: В файле нет данных

Полученный результат:



1.4 Цель: проверить работоспособность программы при наличии текста в файле. Ожидаемый результат: В файле обнаружены некорректные символы.

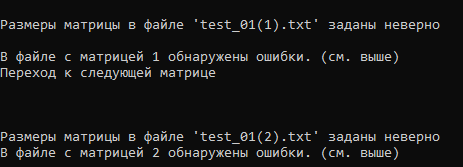
Полученный результат:



1.5 Цель: проверить работоспособность программы при некорректных размерах матриц.

Ожидаемый результат: Размер матрицы заданы неверно

Полученный результат:



* + - 1. **Корректные тесты**

2.1 Цель: проверить работоспособность программы при корректных исходных данных больше нуля.

Исходные данные:

M1 = 3 М2 = 4

N1 = 2 N2 = 2

Исходный массив: { 5 3 6 { 5 3 2 7

3 5 7} 2 5 7 8 }

Ожидаемый результат:

Первая матрица:

53

63

57

Сумма элементов 1-ой строки:8

Сумма элементов 2-ой строки:9

Сумма элементов 3-ой строки:12

Максимальная сумма = 12

Вторая матрица:

53

27

25

78

Сумма элементов 1-ой строки:8

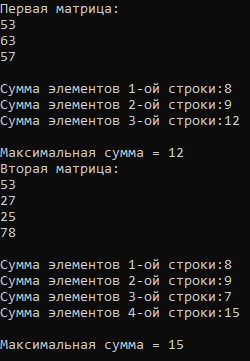
Сумма элементов 2-ой строки:9

Сумма элементов 3-ой строки:7

Сумма элементов 4-ой строки:15

Максимальная сумма = 15

Полученные данные:



2.2 Цель: проверить работоспособность программы при корректных исходных данных больше нуля.

Исходные данные:

M1 = 5 М2 = 6

N1 = 8 N2 = 9

Исходный массив: { 5 4 6 1 3 { 2 8 5 3 1 4

3 6 5 7 6 8 3 4 5 6 2

2 6 8 9 2 5 8 2 9 0 5

2 5 7 8 9 2 4 7 8 9 2

2 4 6 8 9 9 4 7 3 1 4

8 3 8 9 1 7 8 9 3 4 5

3 5 7 4 8 2 4 5 8 3 3

6 7 2 5 7 } 5 6 9 3 2 4

4 2 3 5 6 2 }

Ожидаемый результат:

Первая матрица:

54613365

76268922

57892468

98389135

74867257

Сумма элементов 1-ой строки:33

Сумма элементов 2-ой строки:42

Сумма элементов 3-ой строки:49

Сумма элементов 4-ой строки:46

Сумма элементов 5-ой строки:46

Максимальная сумма = 49

Вторая матрица:

285314834

562582905

247892947

314789345

245833569

324423562

Сумма элементов 1-ой строки:38

Сумма элементов 2-ой строки:42

Сумма элементов 3-ой строки:52

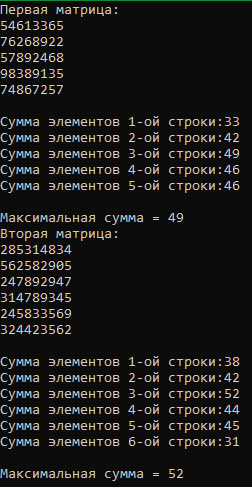
Сумма элементов 4-ой строки:44

Сумма элементов 5-ой строки:45

Сумма элементов 6-ой строки:31

Максимальная сумма = 52

Полученные данные:



**Вывод по работе.**

Разработка программы завершена на том основании, что:

1. Полученные результаты совпали с ожидаемыми.

2. считаем набор тестов полным.