**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 №3**

по учебной дисциплине «Информатика»

на тему «Одномерные массивы»

Вариант № 5

Руководитель работы

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

«\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_2021 г.

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

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

Исполнитель

обучающийся группы М30-110Б-21

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

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

\_\_\_\_\_\_\_\_\_\_\_\_Пысларь А. И

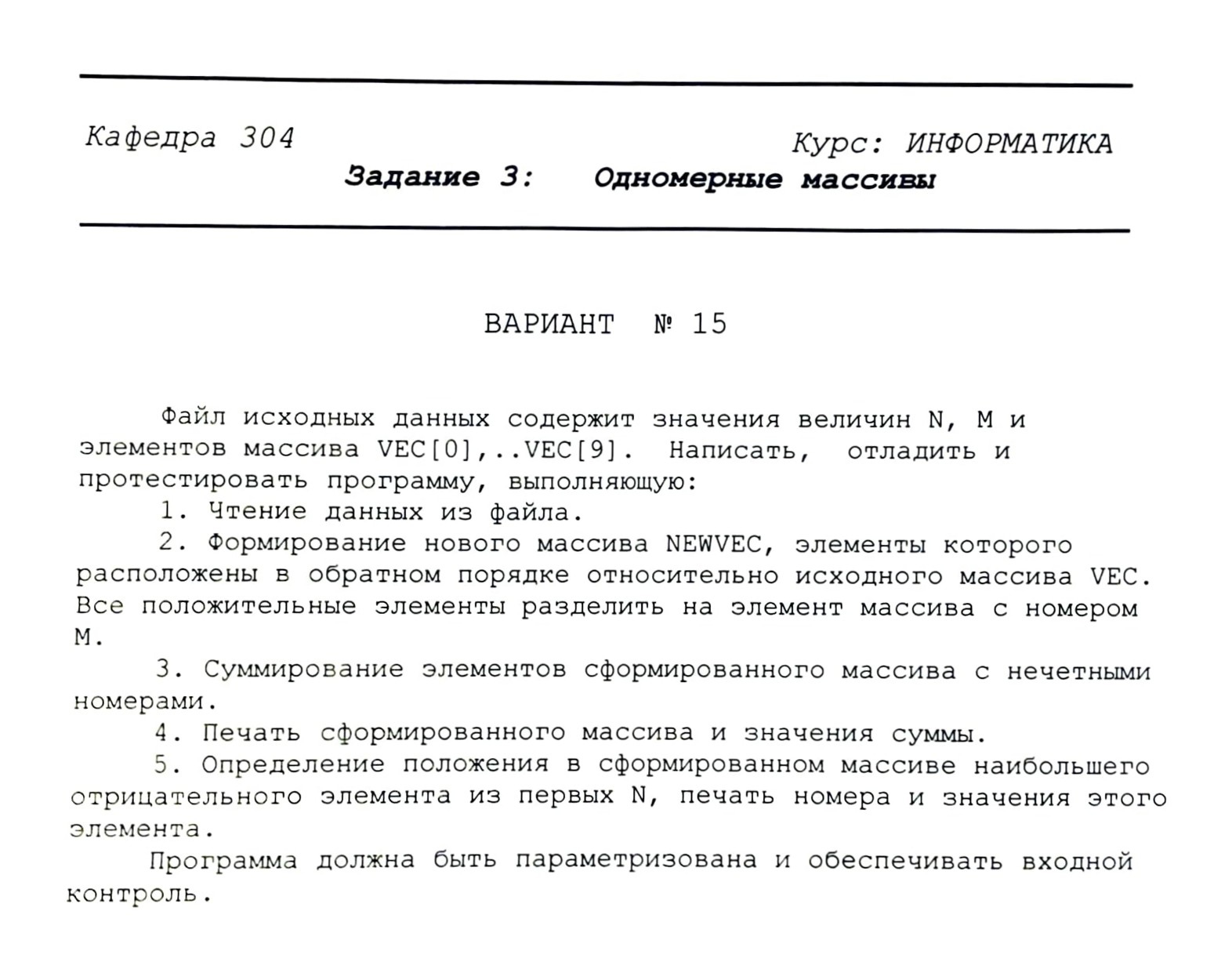
«\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_ 2021 г.

Москва 2021

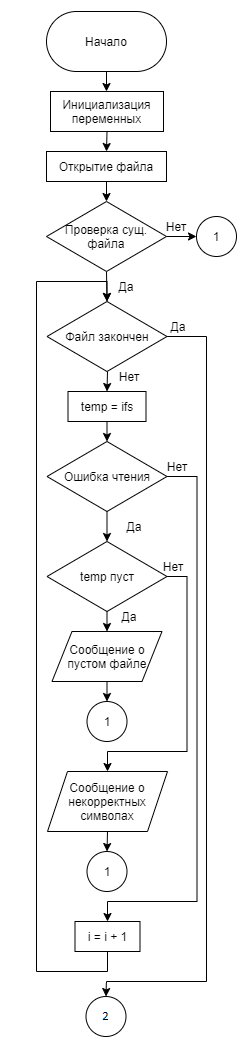
**Содержание.**

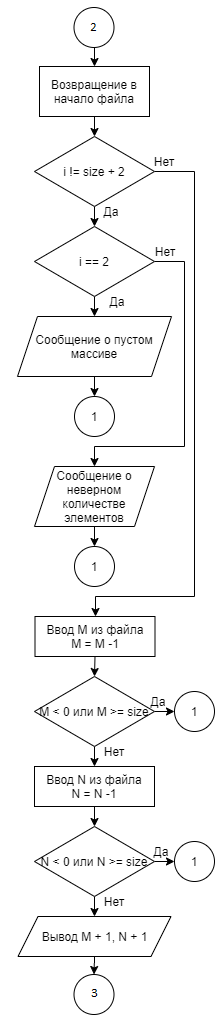
1. Задание……………………………………………….......................................................................3
2. Блок-схема………………………………………………………………………………………………………………4
3. Псевдокод……………………………………………………………………………………………………………….8
4. Код программы……………………………………………………………………………………………………….13
5. Тесты программы…………………………………………………………………………………………………...17
6. Вывод……………………………………………………………………………………………………………………….22

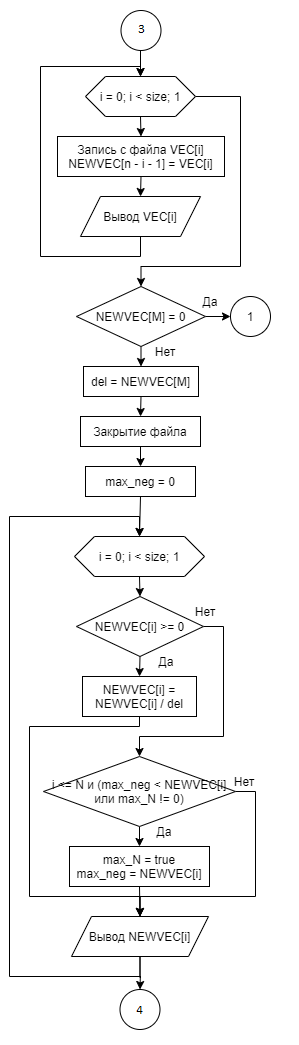
**Задание.**

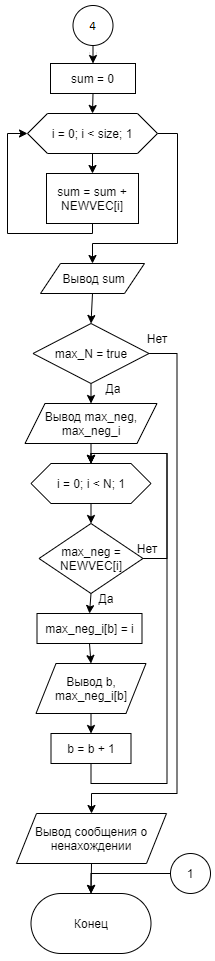
****

**Блок-схема.**





****

****

**Псевдокод.**

**ПСЕВДОКОД**

**НАЧАЛО**

size = 10 // Размер массивов

M //Номер элемента - делителя

N //Количество первых элементов нового массива, среди которых ищем минимум

I //счетчик всех циклов

max\_neg\_i[size] //номера наибольшего максимального числа среди первых N

VEC[size] //Имя изначального массива

NEWVEC[size] //Имя преобразованного массива

sum //Сумма нечетных элементов преобразованного массива

max\_neg //Наибольшее отрицательное среди первых N элементов

del //Делитель

max\_N //Метка, говорящая о наличии отрицательного элемента

temp //Переменная для хранения временных данных

f\_name = “input.txt” // Имя открываемого файла

//ОТКРЫТИЕ ФАЙЛА И ПРОВЕРКА ЕГО НА КОРРЕКТНОСТЬ ДАННЫХ

**ОТКРЫТЬ ФАЙЛ** f\_name

**ЕСЛИ** Файл не открыт

**ТО**

**КОНЕЦ РАБОТЫ АЛГОРИТМА**

**КОНЕЦ**

**ПОКА** Файл не закончился

**ВВОД С ФАЙЛА В** temp

I = I + 1

**ЕСЛИ** Считать не удалось

**ТО**

**ЕСЛИ** temp пуст

**ТО**

**КОНЕЦ РАБОТЫ АЛГОРИТМА**

**ИНАЧЕ**

**КОНЕЦ РАБОТЫ АЛГОРИТМА**

**КОНЕЦ**

**КОНЕЦ**

**ВЕРНУТЬСЯ В НАЧАЛО ФАЙЛА**

**//**ВВОД ДАННЫХ И ВХОДНОЙ КОНТРОЛЬ

**ЕСЛИ** i **НЕ** = size + 2

**ТО**

**ЕСЛИ** size = 2

**ТО**

**КОНЕЦ РАБОТЫ АЛГОРИТМА**

**КОНЕЦ**

**ЕСЛИ** I **НЕ** = 1

**ТО**

**ЕСЛИ** I = 2

**КОНЕЦ РАБОТЫ АЛГОРИТМА**

**КОНЕЦ**

**ЕСЛИ** I <= size +2

**КОНЕЦ РАБОТЫ АЛГОРИТМА**

**ИНАЧЕ**

**КОНЕЦ РАБОТЫ АЛГОРИТМА**

**КОНЕЦ**

**КОНЕЦ РАБОТЫ АЛГОРИТМА**

**КОНЕЦ**

**ВВОД С ФАЙЛА** M

**ЕСЛИ** M < 0 **ИЛИ** M > size

**ТО**

**КОНЕЦ РАБОТЫ ПРОГРАММЫ**

**КОНЕЦ**

**ВВОД С ФАЙЛА** N

**ЕСЛИ** N < 0 **ИЛИ** N > size

**ТО**

**КОНЕЦ РАБОТЫ АЛГОРИТМА**

**КОНЕЦ**

**ВЫВОД** M, N

**ДЛЯ** I = 0; I < size; I = I + 1

**НЦ** //Считываем массив и задаем новый

**ВВОД С ФАЙЛА** VEC[i]

NEWVEC[size – I -1] = VEC[i]

**ВЫВОД** VEC[i]

**КЦ**

**ЕСЛИ** NEWVEC[M] = 0

**ТО**

**КОНЕЦ РАЬОТЫ АЛГОРИТМА**

**КОНЕЦ**

Del = NEWVEC[M]

**ЗАКРЫТЬ ФАЙЛ**

max\_neg = 0

**ДЛЯ** I = 0; I < size; i++

**НЦ** //Преобразуем новый массив

**ЕСЛИ** NEWVEC[i] >= 0

**ТО**

NEWVEC[i] = NEWVEC[i]/del

**ИНАЧЕ ЕСЛИ** I < N **И** (max\_neg < NEWVEC[i] **ИЛИ НЕ** max\_n)

**ТО** Ищем наибольшее отрицательное среди первых N

Max\_N = true

Max\_neg = NEWVEC[i]

**КОНЕЦ**

**ВЫВОД** NEWVEC[i]

**КОНЕЦ**

Sum = 0

**ДЛЯ** I = 1; I < size; I = I + 2

**НЦ //**Считаем сумму элементов с нечетными номерами

Sum = sum + NEWVEC[i]

**КЦ**

**ЕСЛИ** max\_N = true

**ТО**

**ВЫВОД** max\_neg

**ДЛЯ** I = 0 ; i<size;i++

**НЦ**

**ДЛЯ** I = 0; I < N; i++

**ЕСЛИ** max\_neg == NEWVEC[i]

Max\_neg\_i[b] = i

**ВЫВОД** max\_neg[b]

b = b+1

**КЦ**

**ИНАЧЕ**

**ВЫВОД:** Отрицательных элементов не обнаружено.

**КОНЕЦ**

**Код Программы.**

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

\* Курс ИНФОРМАТИКА \*

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

\*Project type :Win64 Console Application \*

\*Project name :Lab3 \*

\*File name :Лабораторная работа.exe \*

\*Language :CPP, MSVS 2021 \*

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

\*Modified By : \*

\*Created :10.12.2021 \*

\*Last revision:17.12.2021 \*

\*Comment :Одномерные массивы \*

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

#include <iostream>

#include <fstream>

using namespace std;

int main()

{

    setlocale(LC\_ALL, "ru");

    const int size = 10;     //Размер массива

    int M;                   //Номер элемента преобразованного массива, на который мы и будеи делить положительные элементы этого массива.

    int N;                   //Количесство первых элементов, среди которых и ищем максимальный отрицательный элемент

    int i = 0;               //Счетчик циклов

    int max\_neg\_i[size];     //номера максимального отрицательного числа из первых N элементов преобразованного массива

    double VEC[size];        //Имя исходного массива

    double NEWVEC[size];     //Имя нового массива

    double sum;              //Сумма нечетных элементов преобразованного массива

    double max\_neg;          //Максимальный отрицательный элемент из первых N элементов преобразованного массива

    double del;              //Делитель

    double temp;             //Временная переменная для хранения данных

    bool max\_N = false;      //Метка, говорящая нам о том, найден ли хотя бы 1 отрицательный элемент из первых N элементов

    string f\_name;           //переменная, в которой хранится имя открываемого файла.

    //f\_name = "input\_ERR\_404.txt";

    f\_name = "input.txt";

    //f\_name = "input1.txt";

    //f\_name = "input2.txt";

    //f\_name = "input3.txt";

    //f\_name = "input4.txt";

    //f\_name = "input5.txt";

    //f\_name = "input6.txt";

    //f\_name = "input7.txt";

    //f\_name = "input8.txt";

    //f\_name = "input9.txt";

    //f\_name = "input10.txt";

    ifstream ifs(f\_name);    //Открываем файл

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

    ОТКРЫТИЕ ФАЙЛА И ПРОВЕРКА ЕГО НА КОРРЕКТНОСТЬ ДАННЫХ

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

    if (!ifs)                                                //Проверка открытия файла

    {

        cout << "Ошибка открытия файла." << endl;

        exit(1);

    }//Конец if

    while (!ifs.eof()) {                                     //проверка содержания файла

        ifs >> temp;

        if (ifs.fail()) {

            if (temp) {

                cout << "Файл пуст." << endl;

                ifs.close();

                exit(2);

            }

            else {

                cout << "Обнаружены некорректные символы.\nПроверьте исходный файл с данными " << f\_name << endl;

                ifs.close();

                exit(2);

            }

        }

        i++;

    }

    ifs.seekg(0, ios::beg);                                  //Возращение в начало файла

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

   ---ВВОД ДАННЫХ И ВХОДНОЙ КОНТРОЛЬ---

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

    if (i != size + 2) {

        if (i == 2) {

            cout << "Где массив, жмотяра?" << endl;

            exit(3);

        }

        cout << "Неверное количество элементов в файле" << endl;

        exit(3);

    }

    ifs >> M;

    M = M - 1;

    if (M < 0 || M >= size) {

        cout << "M д.б. целым положительным числом, которое меньше либо равно размеру массивов." << endl;

        exit(1);

    }//Конец if

    ifs >> N;

    N = N - 1;

    if (N < 0 || N >= size) {

        cout << "N д.б. целым положительным числом, которое меньше либо равно размеру массивов. " << endl;

        exit(1);

    }//Конец if

    cout << "M = " << M + 1 << "\nN = " << N + 1 << endl;

    cout << "Исходный массив:" << endl;

    for (i = 0; i < size; i++)                                 //Считываем массив и задаем новый

    {

        ifs >> VEC[i];

        NEWVEC[size - i - 1] = VEC[i];

        cout << VEC[i] << " ";

    }

    cout << endl;

    if (NEWVEC[M] == 0) {

        cout << "Делить на ноль не стоит..." << endl;

        exit(-1);

    }//Конец if

    del = NEWVEC[M];

    ifs.close();                                            //Закрываем файл

    cout << endl << "Преобразованный массив" << endl;

    max\_neg = 0;

    int b = 0;

    for (i = 0; i < size; i++)                                 //Преобразуем новый массив

    {

        if (NEWVEC[i] >= 0) {

            NEWVEC[i] /= del;

        }

        else if (i < N && (max\_neg <= NEWVEC[i] || !max\_N)) // и ищем максимальное отрицательное среди первых N

        {

            max\_N = true;

            max\_neg = NEWVEC[i];

        }//Конец if else if

        cout << NEWVEC[i] << " ";

    }

    cout << endl;

    sum = 0;

    for (i = 1; i < size; i += 2)                              //Считаем сумму элементов с нечетными номерами

    {

        sum += NEWVEC[i];

    }

    cout << "\nsum: " << sum << endl;

    if (max\_N) {

        cout << "max\_neg = " << max\_neg << endl;

        for (i = 0; i < N; i++) {

            if (max\_neg == NEWVEC[i]) {

                max\_neg\_i[b] = i;

                cout << "\nmax\_neg\_" << b << " = " << max\_neg\_i[b] << endl;

                b++;

            }

        }

    }

    else {

        cout << "Отрицательных элементов в заданой зоне не обнаружено." << endl;

    }//Конец if else

    return 0;

}

**Тесты программы.**

* + - 1. **Некорректные тесты**

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

Ожидаемый результат: Ошибка открытия файла

Полученный результат:



1.2 Цель: проверить работоспособность программы при пустом файле.

Ожидаемый результат: Файл пуст.

Полученный результат:



1.3 Цель: проверить работоспособность программы при наличии текста в файле. Ожидаемый результат: Обнаружены некорректные символы.

Проверьте исходный файл с данными input.txt

Полученный результат: 

1.4 Цель: проверить работоспособность программы при M < 0

Ожидаемый результат: M д.б. целым положительным числом, которое меньше либо равно размеру массивов.

Полученный результат: 

1.5 Цель: проверить работоспособность программы при M >= n

Ожидаемый результат: M д.б. целым положительным числом, которое меньше либо равно размеру массивов.

Полученный результат: 

1.6 Цель: проверить работоспособность программы при N < 0

Ожидаемый результат: N д.б. целым положительным числом, которое меньше либо равно размеру массивов.

Полученный результат: 

1.7 Цель: проверить работоспособность программы при N >= n Ожидаемый результат: N д.б. целым положительным числом, которое меньше либо равно размеру массивов.

Полученный результат: 

1.8 Цель: проверить работоспособность программы при NEWVEC[M] = 0

Ожидаемый результат: Делить на ноль не стоит… Полученный результат: 

1.9 Цель: проверить работоспособность программы при количестве элементов, не равным заданному в алгоритме.

Ожидаемый результат: Кто бросил погрызанный массив?

Количество элементов в файле <= size + 2 и не равно 1

Полученный результат: 

1.10 Цель: проверить работоспособность программы при количестве элементов, не равным заданному в алгоритме.

Ожидаемый результат: Какой-то длинный массив попался…

Количество элементов в файле > size + 2 и не равно 1

Полученный результат:



1.11 Цель: проверить работоспособность программы при остутствии массива.

Ожидаемый результат: Где массив, жмотяра?

Полученный результат: 

1.11 Цель: проверить работоспособность программы при остутствии массива и одного из элементов ( M либо N).

Ожидаемый результат: Неверное количество элементов в файле

Полученный результат: 

* + - 1. **Некорректные тесты**

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

Исходные данные:

M = 9

N = 1

Исходный массив: {1 2 3 4 5 6 7 8 9 10}

Ожидаемый результат:

Исходный массив: 1 2 3 4 5 6 7 8 9 10

Преобразованный массив:

10/2 = 5

9/2 = 4,5

8/2 = 4

7/2 =3,5

6/2 = 3

5/2 = 2,5

4/2 = 2

3/2 = 1,5

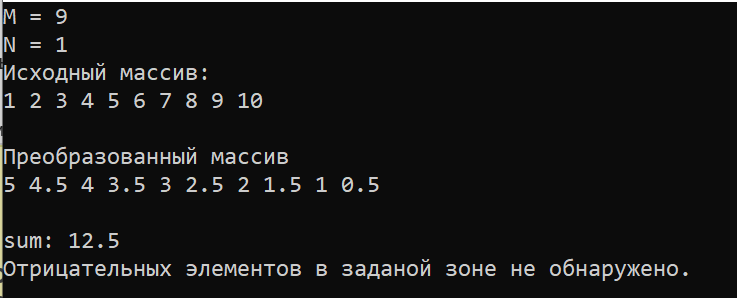
2/2 = 1

1/2=0,5

Sum = 4.5 + 3.5 + 2.5 + 1.5 + 0.5 = 12.5

Отрицательных элементов в заданной зоне не обнаружено.

Полученные данные:



2.2 Цель: проверить работоспособность программы при наличии отрицательных, дробных и дробных отрицательных элементов.

Исходные данные:

M = 9

N = 9

Исходный массив: {1 2 3 -4 5.6 -6.8 7 8 -9 10}

Ожидаемый результат:

Исходный массив: 1 2 3 -4 5.6 -6.8 7 8 -9 10

Преобразованный массив:

10/2 = 5

-9 (<0)

8/2 = 4

7/2 =3,5

-6.8 (<0)

5,6/2 = 2,8

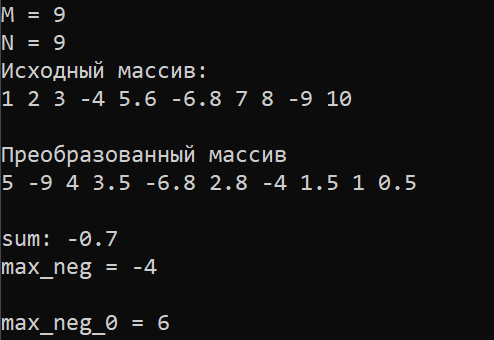
-4 (<0)

3/2 = 1,5

2/2 = 1

1/2=0,5

Sum = -9 + 3,5 + 2,8 + 1,5 + 0,5 = -0,7



2.3 Цель: проверить работоспособность программы при наличии отрицательных, дробных и дробных отрицательных элементов.

Исходные данные:

M = 9

N = 9

Исходный массив: {1 2 -4 3 -4 5.6 -6.8 7 8 -9}

Ожидаемый результат:

Исходный массив: 1 2 3 -4 5.6 -6.8 7 8 -9 10

Преобразованный массив:

-9 (<0)

8/2 = 4

7/2 =3,5

-6.8 (<0)

5,6/2 = 2,8

-4 (<0)

3/2 = 1,5

-4 (<0)

2/2 = 1

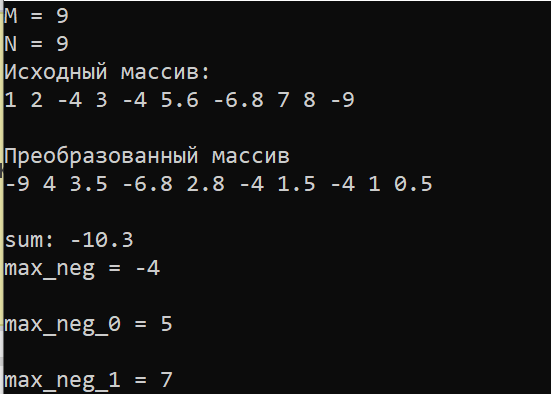
1/2=0,5

Sum = -9 + 3,5 + 2,8 + 1,5 + 0,5 = -0,7

Max\_neg = -4

Max\_neg\_0 = 5

Max\_neg\_1 = 7



**Вывод по работе.**

Разработка программы завершена на том основании, что:

1. Полученные результаты совпали с ожидаемыми.

2. считаем набор тестов полным.