|  |  |
| --- | --- |
|  | МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ (НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)» |

Институт № 3 «Системы управления, информатика и электроэнергетика»

Кафедра 304 «Вычислительные машины, системы и сети»

Лабораторная работа № 3  
по дисциплине «Компьютерные технологии»  
на тему «Одномерные массивы»

Выполнили  
студенты группы М3О-125БВ-24

Вариант №4  
Егоров А.В.,

Федоров А.И.

Приняли

доцент каф. 304, канд. физ.-мат. наук Чечиков Ю.Б.,

ст. преп. каф. 304 Татаринкова Е.М.,

Москва  
2024

Содержание:

Постановка задачи……………………………………………………………….3

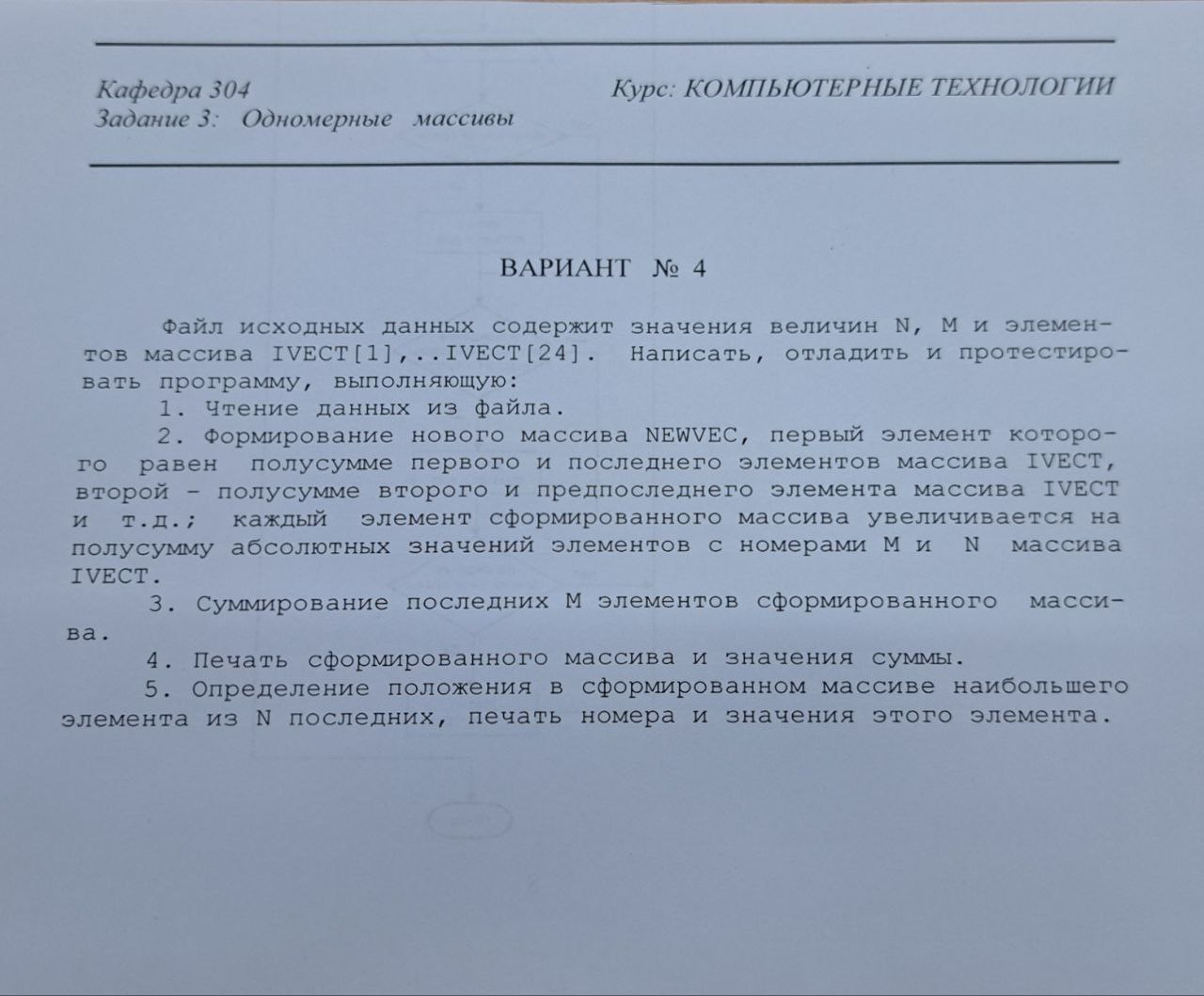
Блок-Схема……………………………………………………………………….4

Код программы…………………………………………………………………..6

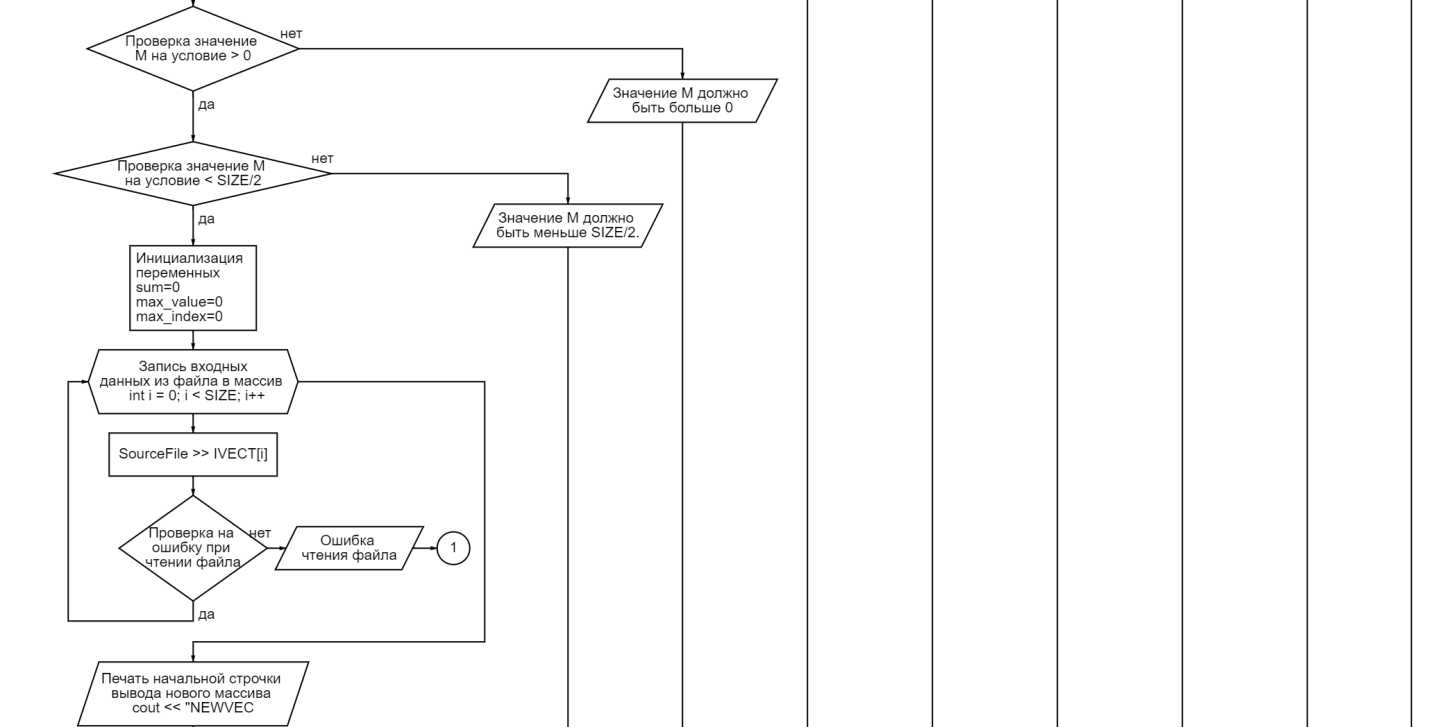
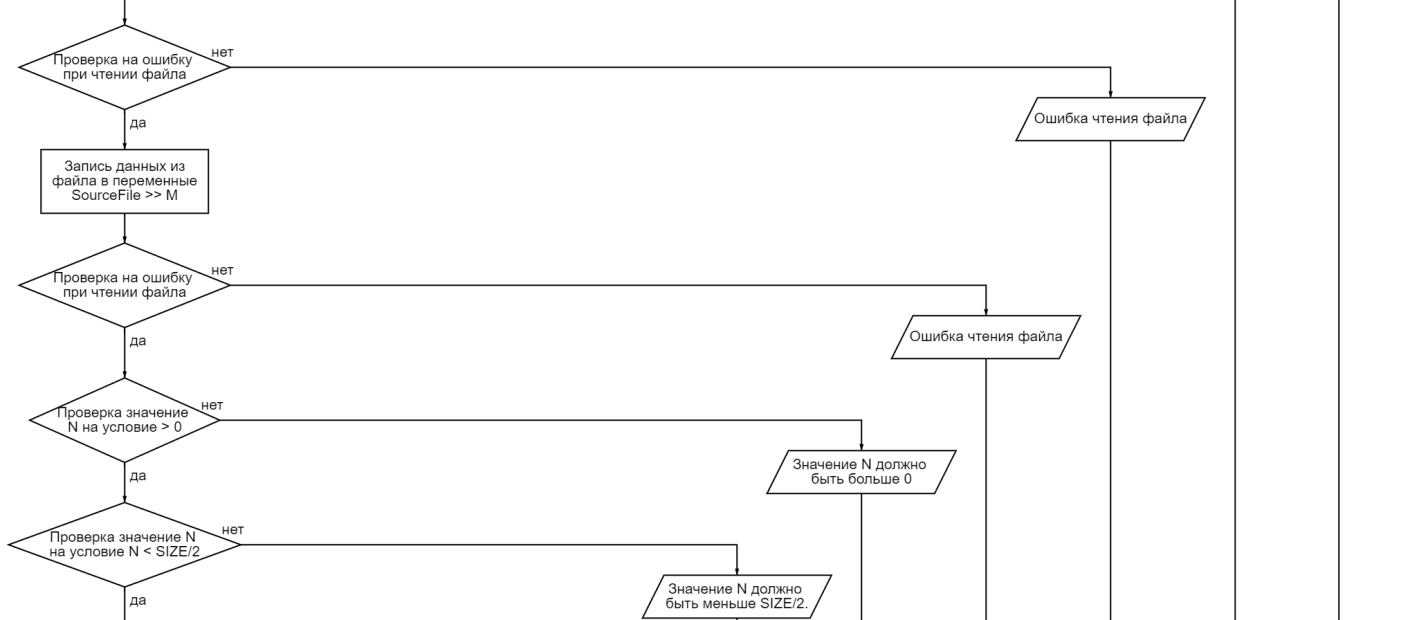
Тестирование…………………………………………………………………….10

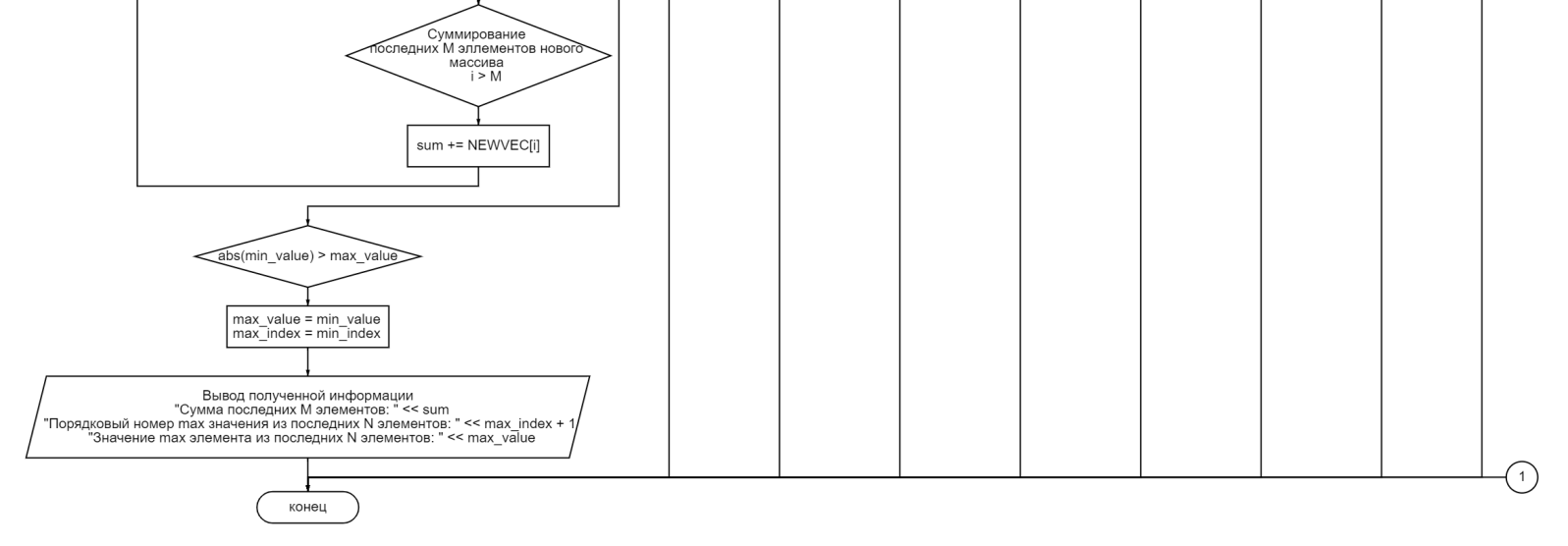
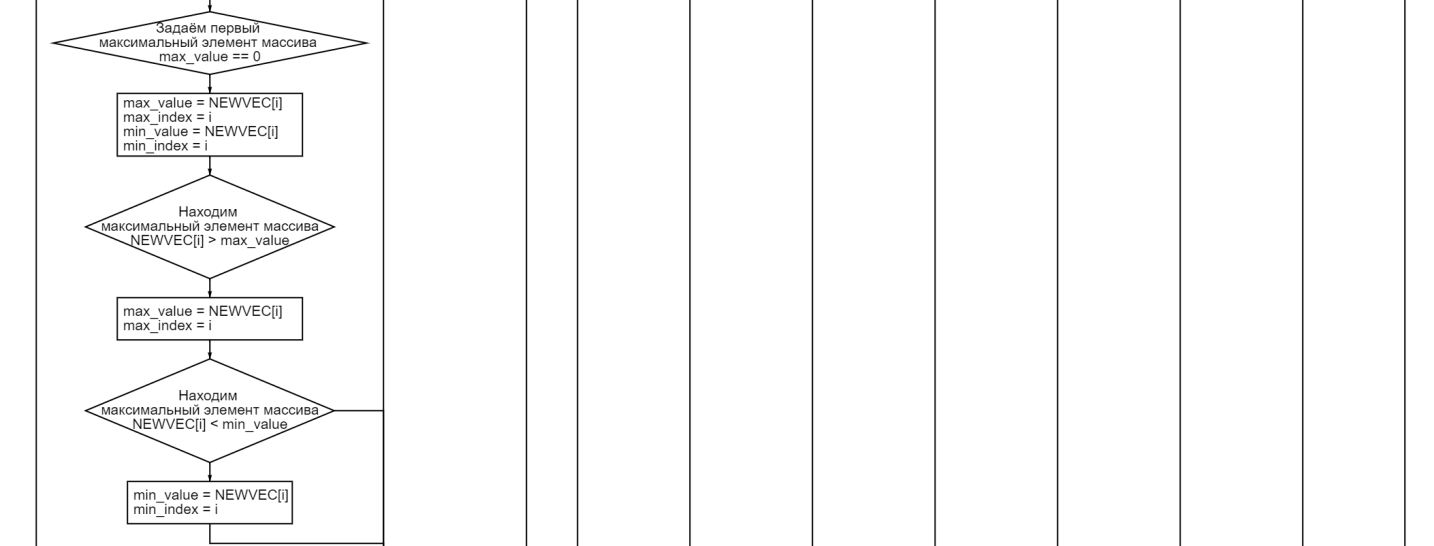
Вывод…………………………………………………………………….………15

Постановка задачи:



Блок-схема:





**Код программы:**

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

\*                     КАФЕДРА № 304 1 КУРС                      \*

\*---------------------------------------------------------------\*

\* Project Type  : GNU/Linux Console Application                 \*

\* Project Name  : proj\_3                                 \*

\* File Name     : main.cpp                                     \*

\* Language      : C/C++                                         \*

\* Programmer(s) : Егоров А.В (swrneko), Федоров А.И.   \*

\* Modifyed By   : Егоров А.В (swrneko)                 \*

\* Edited by : Neovim, Visual Studio \*

\* OS : Arch Linux, Windows 11 \*

\* Github url : https://github.com/swrneko/mai\_shit.git \*

\* Created       : 03.12/24                                      \*

\* Last Revision : 10/12/24                                      \*

\* Comment(s)   : Одномерные массивы.                 \*

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

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

\* Подключение препроцессора и библиотек \*

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

#include <iostream>

#include <fstream>

#include <cmath>

using namespace std;

int main() {

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

\* Объявление переменных \*

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

float N; // Значение N

float M; // Значение M

const int SIZE=24; // Размер входного массива

float IVECT[SIZE]; // Задание размера входного массива

float NEWVEC[SIZE/2]; // Задание размера нового массива

float **sum**; // Сумма последних M элементов нового массива

float max\_value; // Значение наибольшего элемента из N последних

float max\_index; // Порядковый номер наибольшего элемента из N последних

float min\_value; // Значение наибольшего элемента из N последних

float min\_index; // Порядковый номер наибольшего элемента из N последних

const string FileName = "files/source.txt";

/\* Инициализация файла \*/ ifstream SourceFile;

SourceFile.open(FileName);

/\* Проверка файла на фозможность открытия \*/

if (!SourceFile.is\_open()) {

cout << "Ошибка: Не удалось прочитать файл: " << "'" << FileName << "'. " << "Возможно файл удалён или отсутствует." << endl;

SourceFile.close();

exit(1);

}

/\* Проверка файла на пустоту \*/

SourceFile >> N; // Запись данных из файла в переменные

if (SourceFile.eof()) {

cout << "Ошибка: Файл " << "'" << FileName << "'" << " пуст." << endl;

SourceFile.close();

exit(1);

}

/\* Проверка на ошибку при чтении файла \*/

if (SourceFile.fail()) {

cout << "Ошибка: В файле " << "'" << FileName << "'" << " значение N определено неверно (в значении имеются символы отличные от числа)." << endl;

SourceFile.close();

exit(1);

}

SourceFile >> M; // Запись данных из файла в переменные

if (SourceFile.fail()) {

cout << "Ошибка: В файле " << "'" << FileName << "'" << " значение M определено неверно (в значении имеются символы отличные от числа)." << endl;

SourceFile.close();

exit(1);

}

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

\* Входной контроль \*

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

/\* Проверка значение N на условие > 0 \*/

if (N < 0) {

cout << "Значение N должно быть больше 0." << endl;

SourceFile.close();

exit(1);

}

/\* Проверка значение N на условие < SIZE/2 \*/

if (N > SIZE/2.0) {

cout << "Значение N должно быть меньше или равно 12." << endl;

SourceFile.close();

exit(1);

}

/\* Проверка значение M на условие > 0 \*/

if (M < 0) {

cout << "Значение M должно быть больше 0." << endl;

SourceFile.close();

exit(1);

}

/\* Проверка значение M на условие < SIZE/2\*/

if (M > SIZE/2.0) {

cout << "Значение N должно быть меньше или равно 12." << endl;

SourceFile.close();

exit(1);

}

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

\* Входной контроль пройден \*

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

/\* Инициализация переменных \*/

**sum**=0;

max\_value = 0;

max\_index=0;

/\* Запись входных данных из файла в массив \*/

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

SourceFile >> IVECT[i];

if (SourceFile.fail()) {

cout << "Ошибка: В файле " << "'" << FileName << "'" << " одно или несколько значений элементов определены неверно (в значении имеются символы отличные от числа)." << endl;

SourceFile.close();

exit(1);

}

}

/\* Печать начальной строчки вывода нового массива \*/

cout << "NEWVEC: ";

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

\* Нахождение нового массива \*

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

for (int i=0; i < SIZE/2; i++) {

NEWVEC[i] = ((IVECT[i+2] + IVECT[SIZE - (i+2)])/2) + ((M + N)/2); // Формирование элементов нового массива

/\* Печать начального и последующих элементов нового массива \*/

if (i < SIZE/2 - 1) {

cout << NEWVEC[i] << ", ";

}

/\* Печать конечного элемента нового массива \*/

if (i == SIZE/2 - 1) {

cout << NEWVEC[i] << endl;

}

/\* Нахождение максимального элемента в массиве \*/

if (i >= i - N) {

/\* Задаём первый максимальный элемент массива \*/

if (i == 0) {

max\_value = NEWVEC[i];

max\_index = i;

min\_value = NEWVEC[i];

min\_index = i;

}

/\* Находим максимальный элемент массива \*/

if (NEWVEC[i] > max\_value) {

max\_value = NEWVEC[i];

max\_index = i;

}

if (NEWVEC[i] < min\_value) {

min\_value = NEWVEC[i];

min\_index = i;

}

}

/\* Суммирование последних M эллементов нового массива \*/

if (i > i - M) {

**sum** += NEWVEC[i];

}

}

if (**abs**(min\_value) > max\_value) {

max\_value = min\_value;

max\_index = min\_index;

}

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

\* Вывод полученной информации \*

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

cout << "Сумма последних M элементов: " << **sum** << endl;

cout << "Порядковый номер max значения из последних N элементов: " << max\_index + 1 << endl;

cout << "Значение max элемента из последних N элементов: " << max\_value << endl;

exit(0);

}

**Тесты программы**

**1. Некорректные тесты**

**Тест 1**

Цель теста: Проверить работоспособность программы если файл не существует.

Исходные данные: - .

Ожидаемый результат: Ошибка: Неудалось прочитать файл «имя файла».

Полученный результат:

Вывод: полученный результат совпал с ожидаемым.

**Тест 2**

Цель теста: Проверить работоспособность программы при пустом файле с данными.

Исходные данные: - .

Ожидаемый результат: «Ощибка: файл «имя файла» пустой.»

Вывод: полученный результат совпал с ожидаемым.

**Тест 3**

Цель теста: Проверить работоспособность программы если N имеет не верный тип данных.

Исходные данные: **A** 10 12 37 47 23 19 32 48 49 13 31 41 14 25 52 33 88 27 43 28 83 22 42 45 10.

Ожидаемый результат: «Ошибка: В файле «имя файла» значение N пределено неверно.»

Вывод: полученный результат совпал с ожидаемым.

**Тест 4**

Цель теста: Проверить работоспособность программы если M имеет не верный тип данных.

Исходные данные: 11 **B** 12 37 47 23 19 32 48 49 13 31 41 14 25 52 33 88 27 43 28 83 22 42 45 10.

Ожидаемый результат: «Ошибка: В файле «имя файла» значение M пределено неверно.»

Вывод: полученный результат совпал с ожидаемым.

**Тест 5**

Цель теста: Проверить работоспособность программы если среди элементов находится не число.

Исходные данные: 11 10 12 37 47 23 19 32 48 49 13 31 41 14 25 52 **C** 88 27 43 28 83 22 42 45 10.

Ожидаемый результат: «В файле «имя файла» одно или несколько значений элементов определены неверно.»

Вывод: полученный результат совпал с ожидаемым.

**Тест 6**

Цель теста: Проверить работоспособность программы при N < 0.

Исходные данные: **-2** 10 12 37 47 23 19 32 48 49 13 31 41 14 25 52 33 88 27 43 28 83 22 42 45 10.

Ожидаемый результат: «Значение N должно быть больше 0.»



Вывод: полученный результат совпал с ожидаемым.

**# Тест граничных значений N, M = 12  
Тест 7**

Цель теста: Проверить работоспособность программы если N больше размера нового массива NEWVEC.

Исходные данные: **40** 10 12 37 47 23 19 32 48 49 13 31 41 14 25 52 33 88 27 43 28 83 22 42 45 10.

Ожидаемый результат: «Значение N должно быть меньше или равно 12.»

Вывод: полученный результат совпал с ожидаемым.

**Тест 8**

Цель теста: Проверить работоспособность программы если M < 0.

Исходные данные: 11 **-3** 12 37 47 23 19 32 48 49 13 31 41 14 25 52 33 88 27 43 28 83 22 42 45 10.

Ожидаемый результат: «Значение M должно быть больше 0 .»



Вывод: полученный результат совпал с ожидаемым.

**Тест 9**

Цель теста: Проверить работоспособность программы если M больше размера нового массива NEWVEC.

Исходные данные: 11 **40** 12 37 47 23 19 32 48 49 13 31 41 14 25 52 33 88 27 43 28 83 22 42 45 10.

Ожидаемый результат: «Значение M должно быть меньше или равно 12.»

Вывод: полученный результат совпал с ожидаемым.

**2. Корректные тесты**

**Тест 10:**

Цель теста: Проверить корректность работы программы при значении M на границе значений.

Исходные данные: 11 **12** 12 37 47 23 19 32 48 49 13 31 41 14 25 52 33 88 27 43 28 83 22 42 45 10.

Ожидаемый результат:

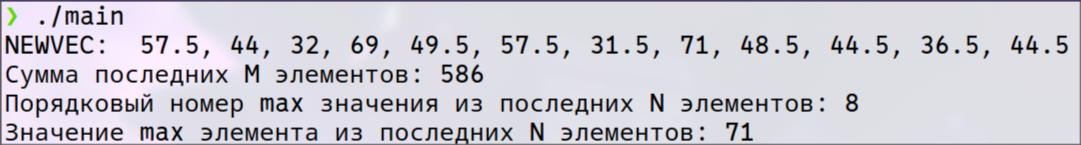
NEWVEC: 57.5, 44, 32, 69, 49.5, 57.5, 31.5, **71**, 48.5, 44.5, 36.5, 44.5

Сумма последних **12** элементов: 586

Порядковый номер max значения из последних **11** элементов: 8

Значение max элемента из последних **11** элементов: 71

Полученный результат:

Вывод: полученный результат совпал с ожидаемым.

**Тест 11:**

Цель теста: Проверить корректность работы программы при значении N на границе значений.

Исходные данные: **12** 10 12 37 47 23 19 32 48 49 13 31 41 14 25 52 33 88 27 43 28 83 22 42 45 10.

Ожидаемый результат:

NEWVEC: 57, 43.5, 31.5, 68.5, 49, 57, 31, **70.5**, 48, 44, 36, 44

Сумма последних **12** элементов: 586

Порядковый номер max значения из последних **11** элементов: 8

Значение max элемента из последних **11** элементов: 70.5

Полученный результат:

**Тест 12**

Цель теста: Проверить корректность работы программы при корректно введённых данных.

Исходные данные: 11 10 12 37 47 23 19 32 48 49 13 31 41 14 25 52 33 88 27 43 28 83 22 42 45 10.

Ожидаемый результат:

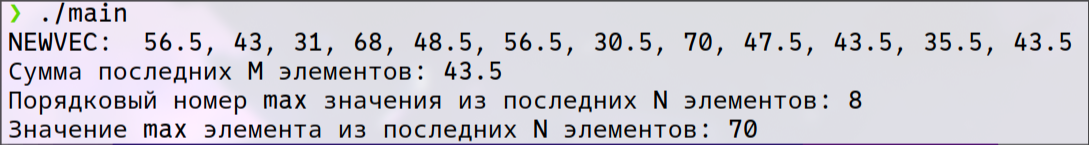
NEWVEC: 56.5, 43, 31, 68, 48.5, 56.5, 30.5, **70**, 47.5, 43.5, 35.5, 43.5

Сумма последних **10** элементов: 43.5

Порядковый номер max значения из последних **11** элементов: 8

Значение max элемента из последних **11** элементов: 70

Полученный результат:

Вывод: полученный результат совпал с ожидаемым.

**Тест 13**

Цель теста: Проверить корректность работы программы при корректно введённых данных.

Исходные данные: 8 1 32 23 44 51 36 94 89 50 78 87 39 99 29 18 63 52 88 13 72 54 43 14 77 16.

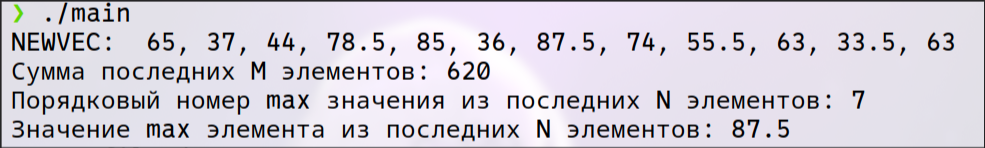
Ожидаемый результат:

NEWVEC: 65, 37, 44, 78.5, 85, 36, 87.5, 74, 55.5, 63, 33.5, 63

Сумма последних **1** элементов: 620

Порядковый номер max значения из последних **8** элементов: 7

Значение max элемента из последних **8** элементов: 87.5

Полученный результат:

Вывод: полученный результат совпал с ожидаемым.

**Вывод по работе**

Разработка программы завершена на том основании, что:

1. Полученные результаты совпали с ожидаемыми.
2. Считаем набор тестов полным.