**Московский авиационный институт**

**(национальный исследовательский институт)**

Институт №3

Системы управления, информатики и электроэнергетики

Кафедра 304

Вычислительный машины, системы и сети

**Отчет по лабораторной работе**

**по учебной дисциплине**

**«Компьютерные технологии»**

**на тему:**

***«Одномерные массивы»***

Группа: М30-109Б-19

Вариант: №2

Выполнили:

*Пенькова А.О., Кузнецов И.И.*

Приняли:

*Давыдкина Е.А.,*

*Секретарев В.Е.*

*Москва 2019*

**Содержание.**

**Задание**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_3

**Блок-схема**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_4-8

**Псевдокод**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_9-11

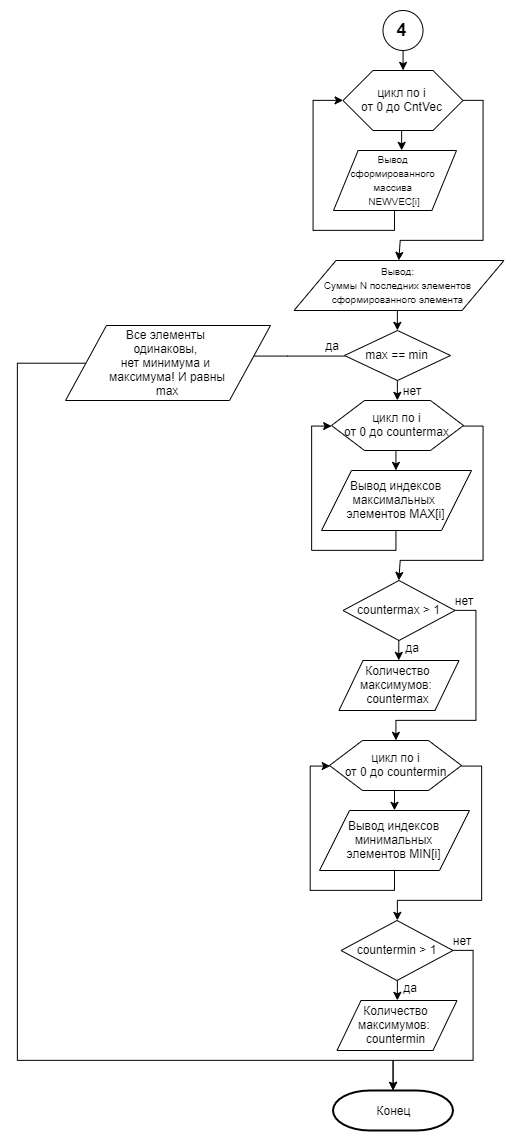
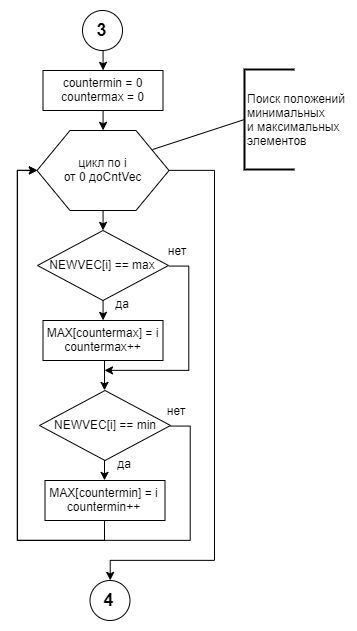
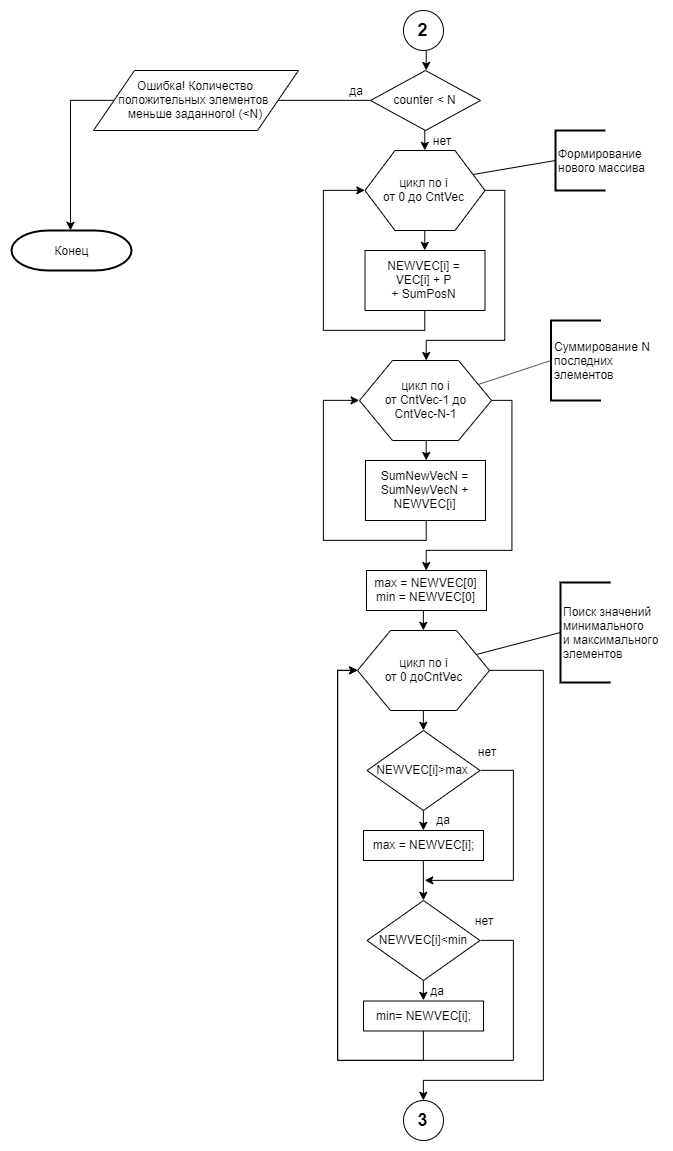
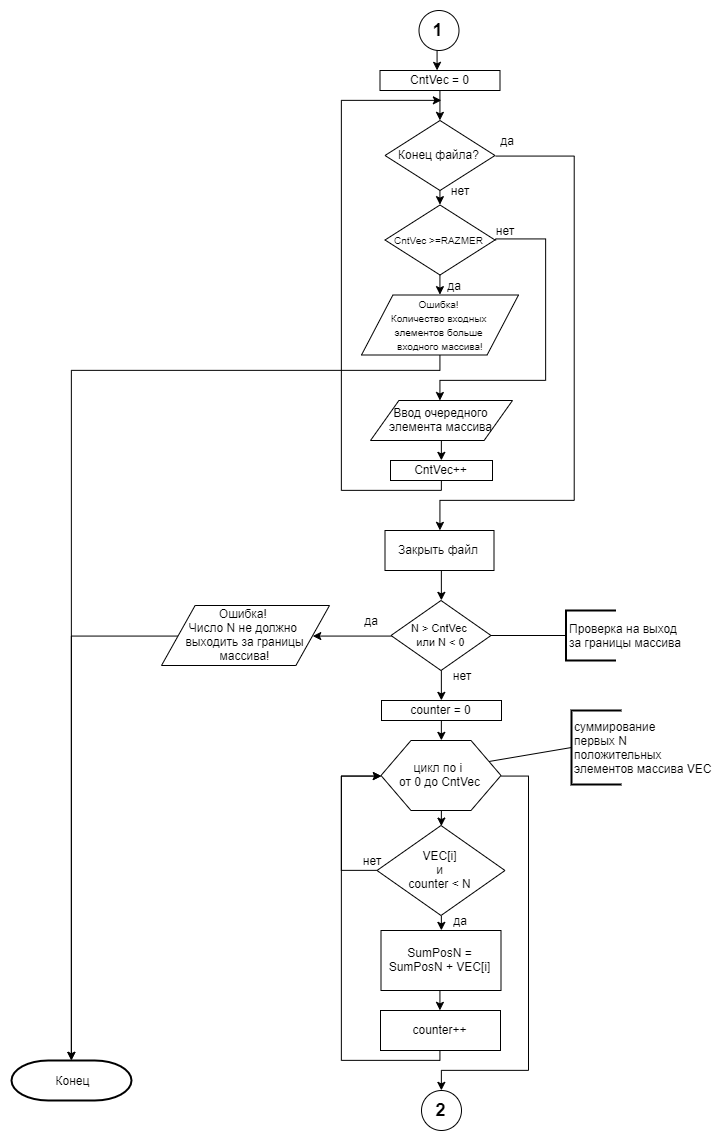
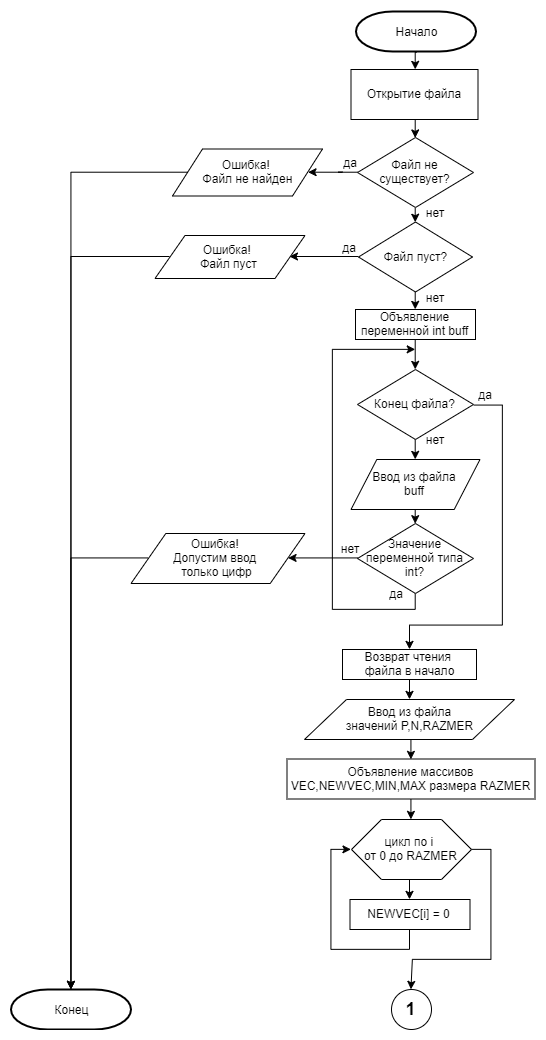
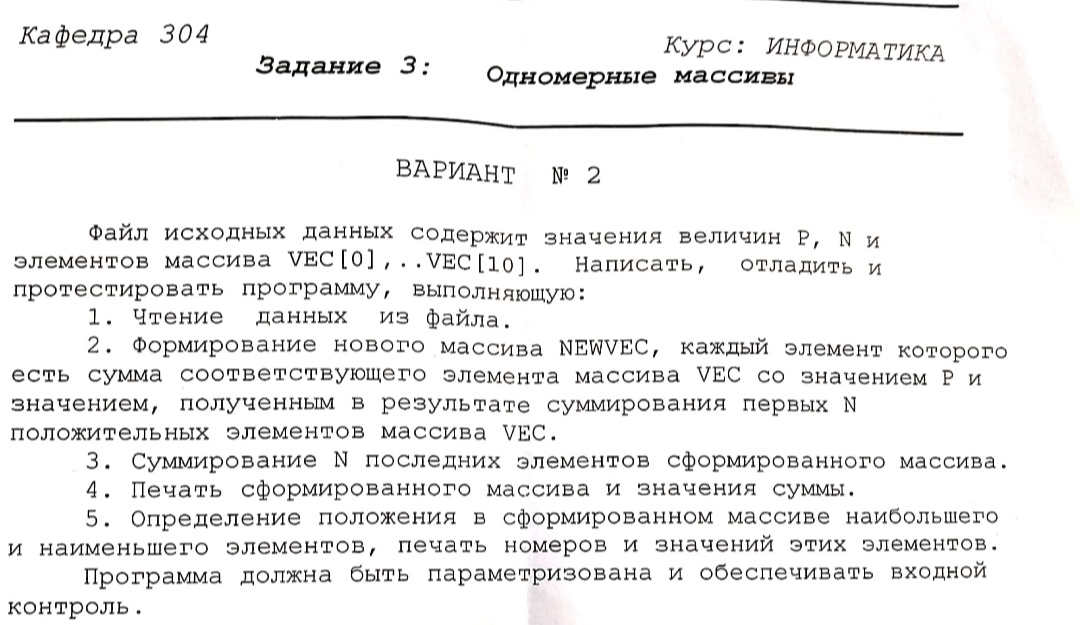
**Код программы**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_12-14

**Ручной счет**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_15

**Тесты**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_16-27

**Вывод**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_28

**Задание**



**Псевдокод:**

**АЛГ**

**НАЧАЛО**

Натуральные аргументы:

P // значение P

N // значение N

SumPosN = 0 //счетчик реального размера массива

SumNewVecN = 0 // счетчик для подсчета N положительных элементов в массиве

*Создать объект класса fin*

*Открытие файла*

**ЕСЛИ** ФАЙЛ НЕ НАЙДЕН

**ТО**

*Вывести: Ошибка! Файл не найден!*

*Закрыть программу*

**КОНЕЦ ЕСЛИ**

**ЕСЛИ** НАЙДЕННЫЙ ФАЙЛ ПУСТ

**ТО**

*Вывести: Ошибка! Файл пуст!*

*Закрыть программу*

**КОНЕЦ ЕСЛИ**

Объявление переменной int buff

**НАЧАЛО ЦИКЛА ПОКА** ЭЛЕМЕНТЫ НЕ КОНЧАТСЯ

Считывание buff

**НАЧАЛО ЕСЛИ** ЗНАЧЕНИЕ ПЕРЕМЕННОЙ НЕ ТИПА INT

*Вывести: Ошибка! Допустим ввод только цифр!*

*Закрыть программу*

**КОНЕЦ ЕСЛИ**

**КОНЕЦ ЦИКЛА**

*Возврат чтения файла на самое начало*

Считывание P, N, RAZMER;

Объявление массивов VEC, NEWVEC, MIN, MAX размера RAZMER

**НАЧАЛО ЦИКЛА** ДЛЯ i ОТ 0 ДО RAZMER

NEWVEC[i] = 0

**КОНЕЦ ЦИКЛА**

Инициализация переменной CntVec = 0

**НАЧАЛО ЦИКЛА ПОКА** ЭЛЕМЕНТЫ НЕ КОНЧАТСЯ

**НАЧАЛО ЕСЛИ** (CntVec >= RAZMER) //ПРОВЕРКА НА ВЫХОД ЗА ГРАНИЦЫ МАССИВА

*Вывести: Ошибка! Количество входных элементов больше входного массива!*

*Закрыть программу*

**КОНЕЦ ЕСЛИ**

считываниеVEC[CntVec]// ЧТЕНИЕ МАССИВА ИЗ ФАЙЛА

CntVec++

**КОНЕЦ ЦИКЛА**

*Закрыть файл*

**ЕСЛИ** (N > CntVec или N < 0)// ПРОВЕРКА НА ВЫХОД ЗА ГРАНИЦЫ МАССИВА

**ТО**

*Вывести*: *Ошибка! Размер массива не соответствует условиям!*

*Закрыть программу*

**КОНЕЦ ЕСЛИ**

Инициализация переменной counter = 0

**НАЧАЛО ЦИКЛА** ДЛЯ i ОТ 0 ДО CntVec //СУММИРОВАНИЕ ПЕРВЫХ N ПОЛОЖИТЕЛЬНЫХ

**ЕСЛИ** (VEC[i] > 0) **И** (counter < N) ЭЛЕМЕНТОВ МАССИВА VEC

**ТО**

SumPosN += VEC[i];

counter++;

**КОНЕЦ ЕСЛИ**

**КОНЕЦ ЦИКЛА**

**НАЧАЛО ЕСЛИ** COUNTER < N

*Вывести*: *Ошибка! Количество положительных элементов меньше заданного! (<N)!*

*Закрыть программу*

**КОНЕЦ ЕСЛИ**

**НАЧАЛО ЦИКЛА** ДЛЯi ОТ 0 ДО CntVec //ФОРМИРОВАНИЕ НОВОГО МАССИВА

NEWVEC[i] = VEC[i] + P + SumPosN;

**КОНЕЦ ЦИКЛА**

**НАЧАЛО ЦИКЛА** ДЛЯ i ОТ CntVec-1 ДО CntVec-N-1 //СУММИРОВАНИЕ N ПОСЛЕДНИХ ЭЛЕМЕНТОВ

SumVecN += NEWVEC[i]

**КОНЕЦ ЦИКЛА**

Инициализация переменных int: max = NEWVEC[0], min = NEWVEC[0]

**НАЧАЛО ЦИКЛА** ДЛЯi ОТ 0 ДО CntVec //ПОИСК ЗНАЧЕНИЙ МИНИМАЛЬНОГО

**ЕСЛИ** NEWVEC[i] > max И МАКСИМАЛЬНОГО ЭЛЕМЕНТОВ

**ТО**

max = NEWVEC[i];

**КОНЕЦ ЕСЛИ**

**ЕСЛИ** NEWVEC[i] < min

**ТО**

min = NEWVEC[i];

**КОНЕЦ ЕСЛИ**

**КОНЕЦ ЦИКЛА**

**НАЧАЛО ЦИКЛА** ДЛЯi ОТ 0 ДО CntVec //ПОИСК ПОЛОЖЕНИЯ И ЗНАЧЕНИЙ МИНИМАЛЬНОГО

**ЕСЛИ** NEWVEC[i] == max И МАКСИМАЛЬНОГО ЭЛЕМЕНТОВ

**ТО**

MAX[COUNTERMAX] = i

countermax++

**КОНЕЦ ЕСЛИ**

**ЕСЛИ** NEWVEC[i] == min

**ТО**

MIN[COUNTERMAX] = i

countermin++

**КОНЕЦ ЕСЛИ**

**КОНЕЦ ЦИКЛА**

**НАЧАЛО ЦИКЛА** ДЛЯi ОТ 0 ДО CntVec

*Вывести:**NEWVEC[i] // сформированный массив*

**КОНЕЦ ЦИКЛА**

*Вывести: N // сумма последних элементов сформированного элемента*

***НАЧАЛО ЕСЛИ*** *max == min*

*Вывести:* Все элементы одинаковы, нет минимума и максимума! И равны: max

*Закрыть программу*

***КОНЕЦ ЕСЛИ***

**НАЧАЛО ЦИКЛА** ДЛЯi ОТ 0 ДО countermax

*Вывести:**MAX[i] // положения максимумов*

**КОНЕЦ ЦИКЛА**

**НАЧАЛО ЕСЛИ** countermax > 1

*Вывести: Количество максимумов: countermax*

**КОНЕЦ ЕСЛИ**

**НАЧАЛО ЦИКЛА** ДЛЯi ОТ 0 ДО countermin

*Вывести:**MIN[i] // положения минимумов*

**КОНЕЦ ЦИКЛА**

**НАЧАЛО ЕСЛИ** countermin > 1

*Вывести: Количество максимумов: countermin*

**КОНЕЦ ЕСЛИ**

**Код программы:**

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

\* Project name: lab3\_ \*

\* Project type: Win32 Console Application \*

\* File name: lab3\_.cpp \*

\* Language: CPP, MSVS 2019 and above \*

\* Programmer: Кузнецов Илья Игоревич, Пенькова Алина Олеговна М3О-109Б \*

\* Modified by: Пенькова Алина Олеговна\*

\* Created: 18.11.2019 \*

\* Last revision: 02.12.2019 \*

\* Comment: Одномерные массивы \*

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

#include <iostream>

#include <cmath>

#include <fstream>

#include <vector>

using namespace std;

int main() {

system("color F0");

setlocale(LC\_ALL, "Russian");

int RAZMER; //размер массива

int P; //значение P (исходные данные)

int N; //значение N (исходные данные)ч

int SumPosN = 0; //сумма первых n пололожительных элементов

int SumNewVecN = 0; //сумма vec, p и суммы первых n пололожительных элементов

ifstream fin;

fin.open("Text01.txt"); // начинаем чтение файла

//НАЧАЛО ВВОДА И КОРРЕКТНОСТИ ВВЕДЕННЫХ ДАННЫХ

if (!fin) // если файл не найден

{

cout << "Ошибка! файл не найден" << endl; //вывод сообщения об ошибки

system("pause");

return 1;

}

if (fin.peek() == EOF) //если найденный файл пуст

{

cout << "Ошибка! файл пуст!" << endl;// вывод сообщения об ошибке

system("pause");

return 1;

}

int buff; // буффер для проверки корректности вводимых данных

while (!fin.eof()) { //проверка на ввод цифр

fin >> buff;

if (!fin){

cout << "Ошибка! Допустим ввод только цифр!" << endl;

system("pause");

return 1;

}

}

fin.seekg(0); //возврат чтения файла на самое начало

fin >> P >> N >> RAZMER; //чтение значений величин P и N и размера массива

vector <int> VEC(RAZMER); //объявление массива (исходные данные)

vector <int> NEWVEC(RAZMER); //объявление нового массива

vector <int> MIN(RAZMER); //массив с индексами минимумов

vector <int> MAX(RAZMER); //массив с индексами максимумов

for (int i = 0; i < RAZMER; i++) { // обнуление массива

NEWVEC[i] = 0;

}

int CntVec = 0; //счетчик реального размера массива

while (!fin.eof()) //чтение элементов массива, пока элементы не закончатся

{

if (CntVec >= RAZMER) //проверка на выход за границы массива

{

cout << "Ошибка! Количество входных элементов больше входного массива!" << endl;

system("pause");

return 1;

}

fin >> VEC[CntVec]; //чтение массива из файла

CntVec++; //подсчет количества элементов массив

}

fin.close(); //закрыть файл

if ((N > CntVec) || (N < 0)) { // проверка на границы массива

cout << "Ошибка! Число N не должно выходить за границы массива!" << endl;

system("pause");

return 1;

}

//КОНЕЦ ВВОДА И КОРРЕКТНОСТИ ВВЕДЕННЫХ ДАННЫХ

//НАЧАЛО ОСНОВНОЙ ЧАСТИ ПРОГРАММЫ

int counter = 0; //счетчик для подсчета N положительных элементов в массиве

for (int i = 0; i < CntVec; i++) { //суммирование первых N положительных элементов массива VEC

if ((VEC[i] > 0) && (counter < N)) {

SumPosN += VEC[i];

counter++;

}

}

if (counter < N) {

cout << "Ошибка! Количество положительных элементов меньше заданного! (<N)" << endl;

system("pause");

return 1;

}

for (int i = 0; i < CntVec; i++) { //формирование нового массива

NEWVEC[i] = VEC[i] + P + SumPosN;

}

for (int i = CntVec - 1; i > CntVec - N - 1; i--) { //суммирование N последних элементов

SumNewVecN += NEWVEC[i];

}

int max = NEWVEC[0]; // значение максимального элемента

int min = NEWVEC[0]; // значение минимального элемента

for (int i = 0; i < CntVec; i++) { //поиск значений минимального и максимального элементов

if (NEWVEC[i] > max) { //максимальное значение

max = NEWVEC[i];

}

if (NEWVEC[i] < min) { //минимальное значение

min = NEWVEC[i];

}

}

int countermin = 0; //счетчик минимумов

int countermax = 0; //счетчик максимумов

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

if (NEWVEC[i] == max) {

MAX[countermax] = i;

countermax++;

}

if (NEWVEC[i] == min) {

MIN[countermin] = i;

countermin++;

}

}

//КОНЕЦ ОСНОВНОЙ ЧАСТИ ПРОГРАММЫ

//ВЫВОД НАЧАЛО

cout << "Сформированный массив: " << endl;

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

cout << "NEWVEC[" << i << "]: " << NEWVEC[i] << endl;

}

cout << endl;

cout << "Сумма N{" << N << "} последних элементов сформированного элемента: " << SumNewVecN << endl << endl;

if (max == min) {

cout << "Все элементы одинаковы, нет минимума и максимума! И равны: " << max << endl;

system("pause");

return 1;

}

cout << "Положения наибольшего элемента: ";

for (int i = 0; i < countermax; i++) { cout << "[" << MAX[i] << "] "; }

cout << endl;

cout << "Значения наибольшего элемента: " << max << endl;

if (countermax > 1) {

cout << "Количество максимумов: " << countermax << endl;

}

cout << endl;

cout << "Положения наименьшего элемента: ";

for (int i = 0; i < countermin; i++) { cout << "[" << MIN[i] << "] "; }

cout << endl;

cout << "Значения наименьшего элемента: " << min << endl;

if (countermin > 1) {

cout << "Количество минимумов: " << countermin << endl;

}

cout << endl;

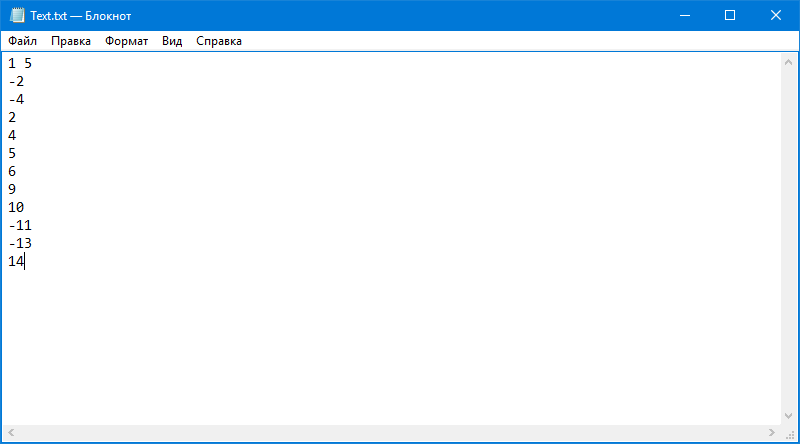
//ВЫВОД КОНЕЦ

system("pause");

return 0; // завершение программы

}

**Ручной счет**



Сумма первых положительных элементов массива **VEC**:

2+4+5+6+9=**26**

Формирование нового элемента **NEWVEC**:

[0]: -2 + 1 + 26 = 25 [1]: -4 + 1 + 26 = 23

[2]: 2 + 1 + 26 = 29 [3]: 4 + 1 + 26 = 31

[4]: 5 + 1 + 26 = 32 [5]: 6 + 1 + 26 = 33

[6]: 9 + 1 + 26 = 36 [7]: 10 + 1 + 26 = 37

[8]: -11 + 1 + 26 = 16 [9]: -13 + 1 + 26 = 14

[10]: 14 + 1 + 26 = 41

Сумма **N(5)** последних элементов сформированного элемента: 41 + 14 + 16 + 37 + 36 = **144**

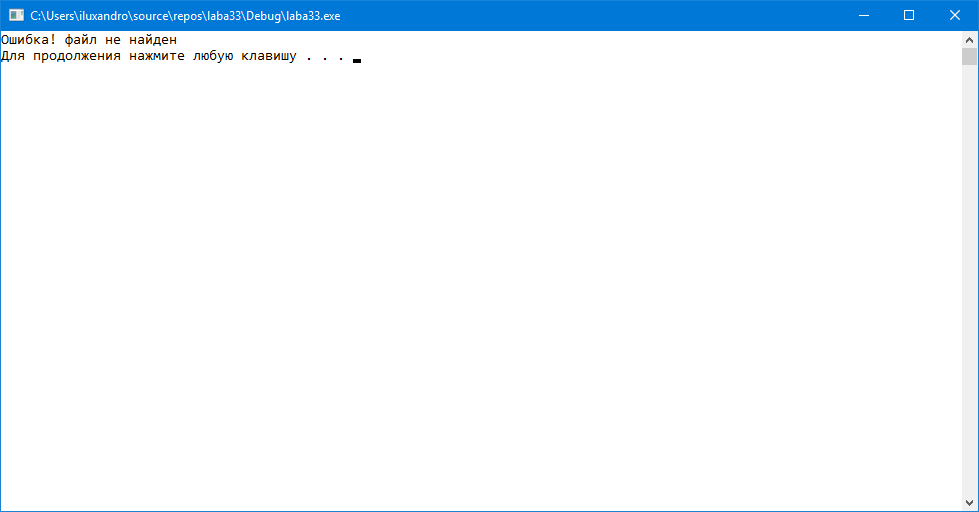
**Тесты**

**Тест №1.**

**Цель:** проверить работу программы с отсутствующим файлом

**Ожидаемые результаты:** Вывод сообщения об ошибке

**Полученные результаты:**



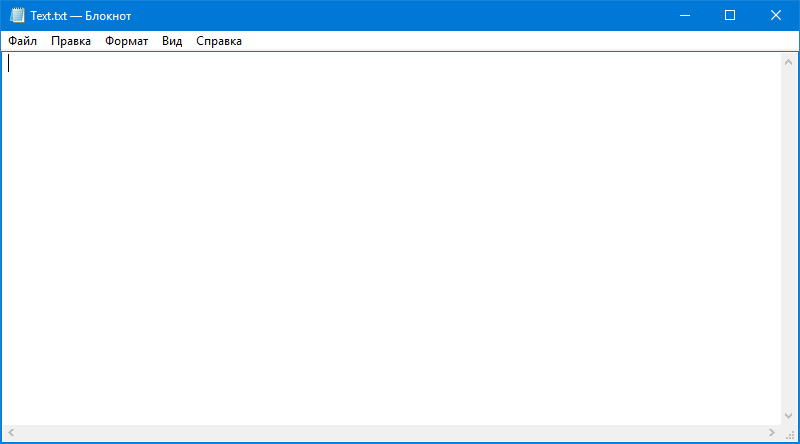
**Вывод**: Полученные результаты равны ожидаемым.

Тест не выявил ошибок. Работа программы корректна

**Тест №2.**

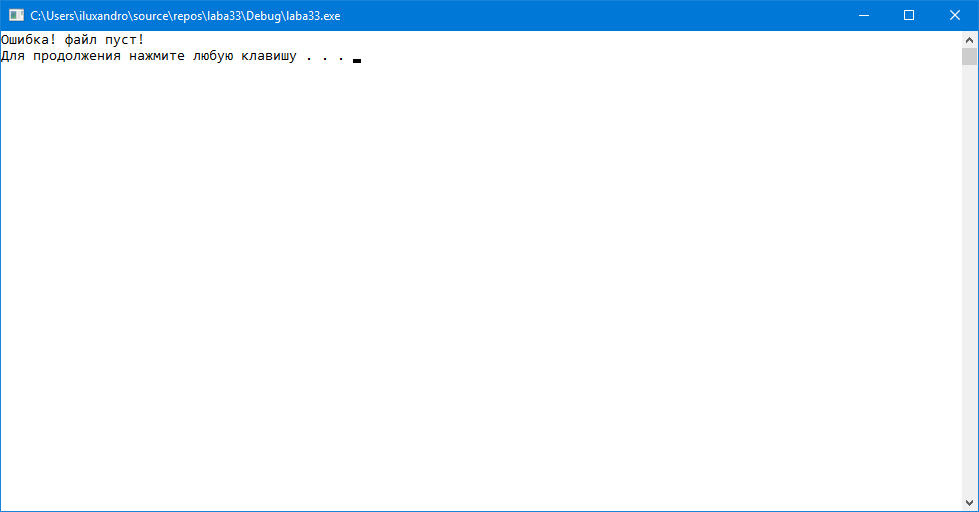
**Цель:** проверить работу программы с пустым файлом

**Ввод:**

****

**Ожидаемые результаты:** Вывод сообщения об ошибке

**Полученные результаты:**



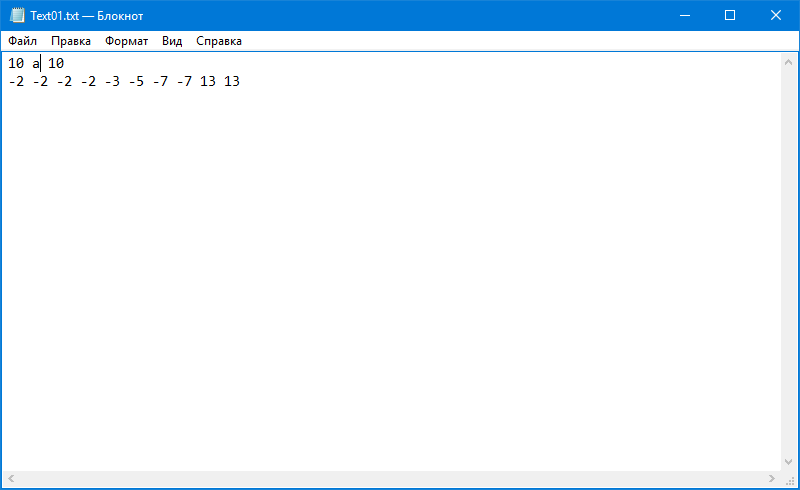
**Вывод**: Полученные результаты равны ожидаемым.

Тест не выявил ошибок. Работа программы корректна

**Тест №3.**

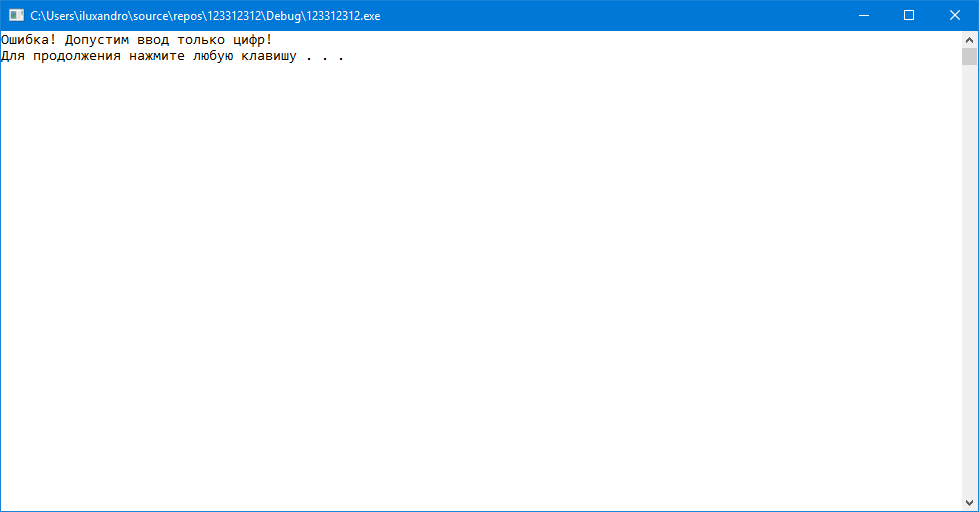
**Цель:** проверить работу программы с буквами в исходных файлах

**Ввод:**



**Ожидаемые результаты:** Вывод сообщения об ошибке

**Полученные результаты:**



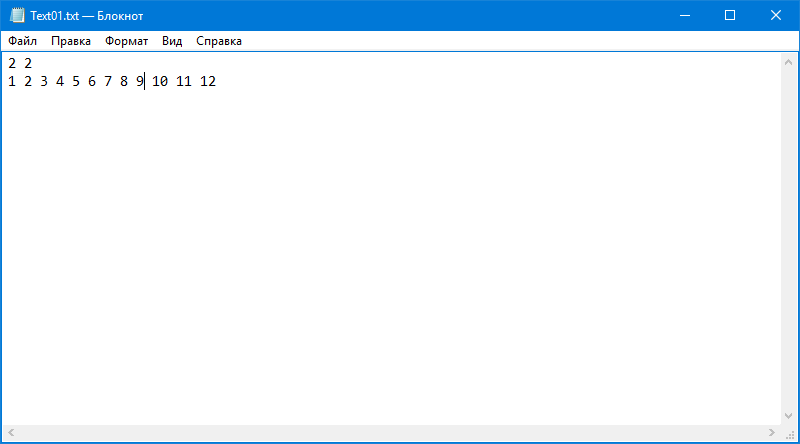
**Вывод**: Полученные результаты равны ожидаемым.

Тест не выявил ошибок. Работа программы корректна

**Тест №4.**

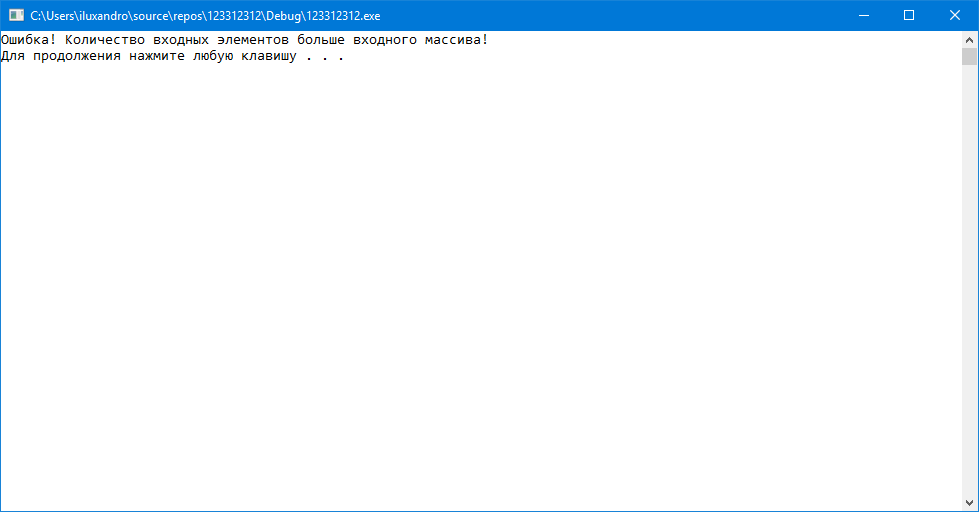
**Цель:** проверить работу программы с лишними значениями в исходных данных

**Ввод:**



**Ожидаемые результаты:** Вывод сообщения об ошибке

**Полученные результаты:**



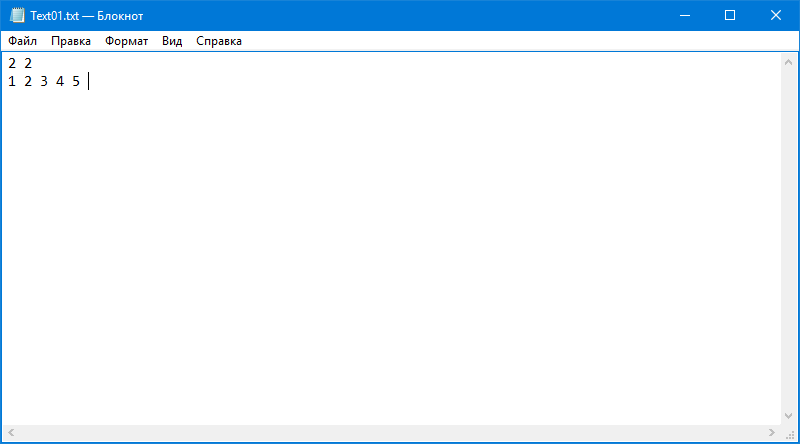
**Вывод**: Полученные результаты равны ожидаемым.

Тест не выявил ошибок. Работа программы корректна

**Тест №5.**

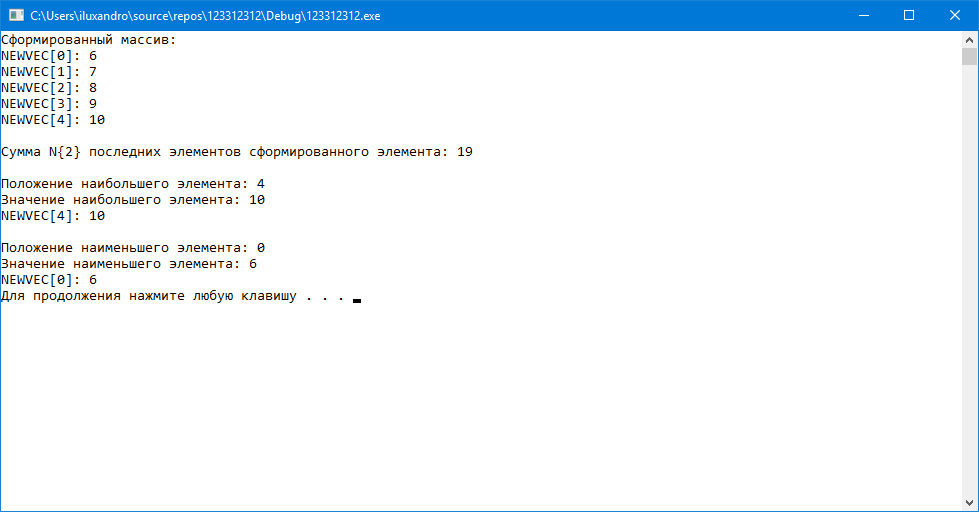
**Цель:** проверить работу программы при уменьшении количества элементов в массиве

**Ввод:**



**Ожидаемые результаты:** Корректная работа программы

**Полученные результаты:**



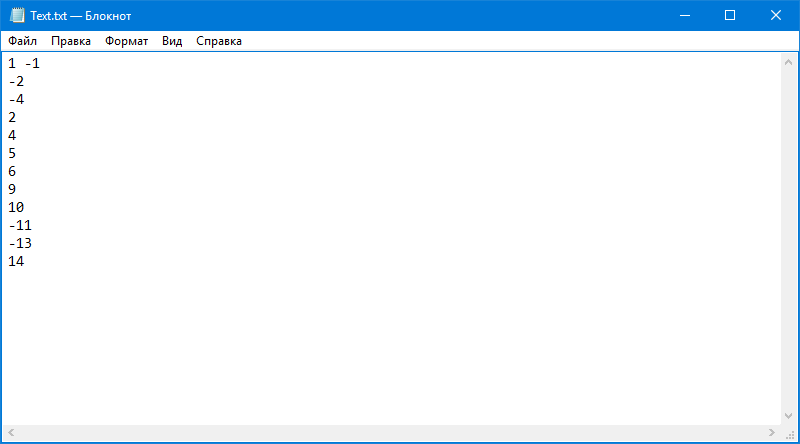
**Вывод**: Полученные результаты равны ожидаемым.

Тест не выявил ошибок. Работа программы корректна

**Тест №6.**

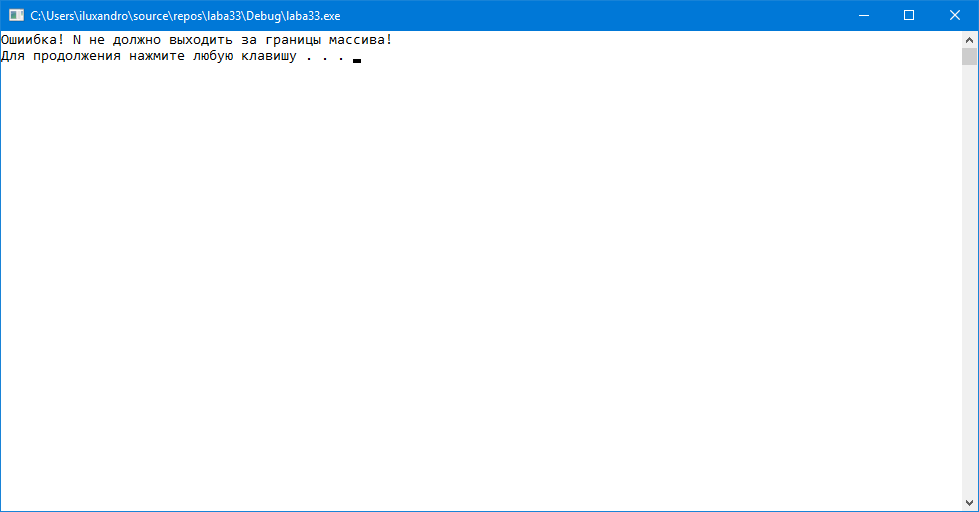
**Цель:** проверить работу программы с N<0

**Ввод:**



**Ожидаемые результаты:** Вывод сообщения об ошибке

**Полученные результаты:**



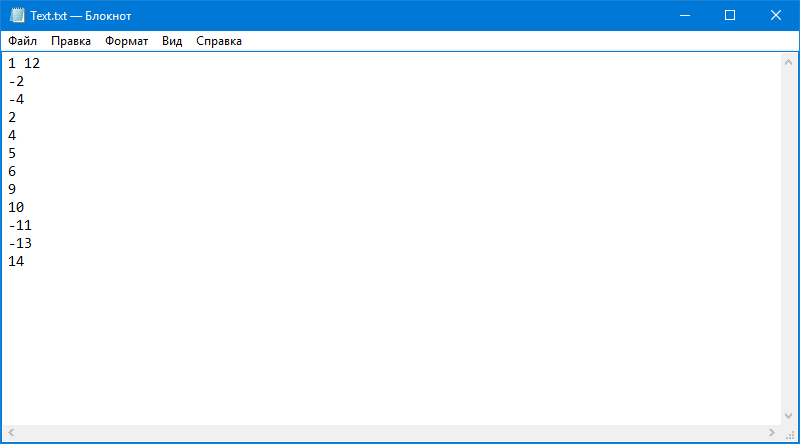
**Вывод**: Полученные результаты равны ожидаемым.

Тест не выявил ошибок. Работа программы корректна

**Тест №7.**

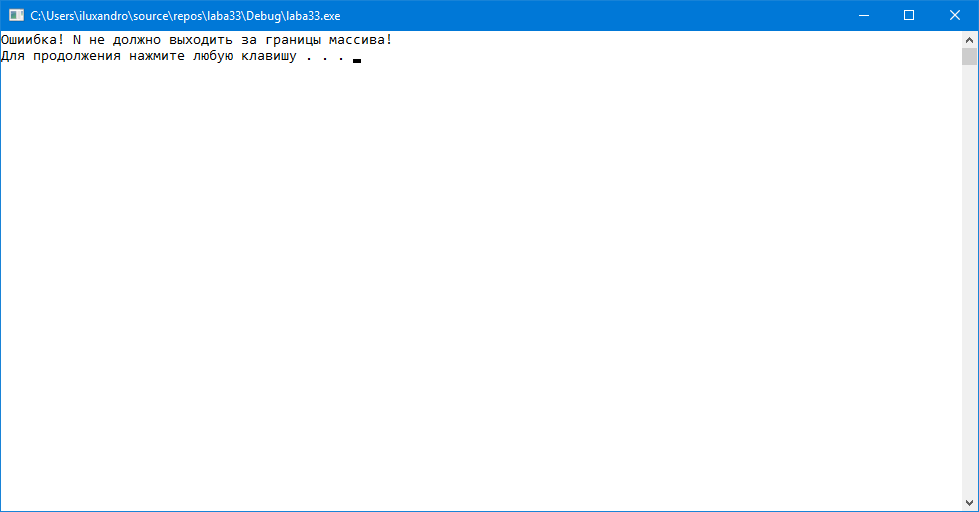
**Цель:** проверить работу программы с N выходящим за границы массива в положительную сторону (N>CntVec)

**Ввод:**



**Ожидаемые результаты:** Вывод сообщения об ошибке

**Полученные результаты:**



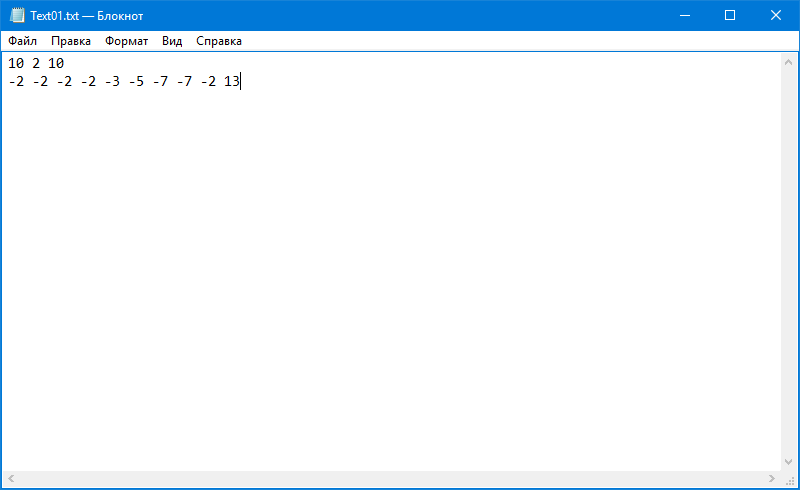
**Вывод**: Полученные результаты равны ожидаемым.

Тест не выявил ошибок. Работа программы корректна

**Тест №8.**

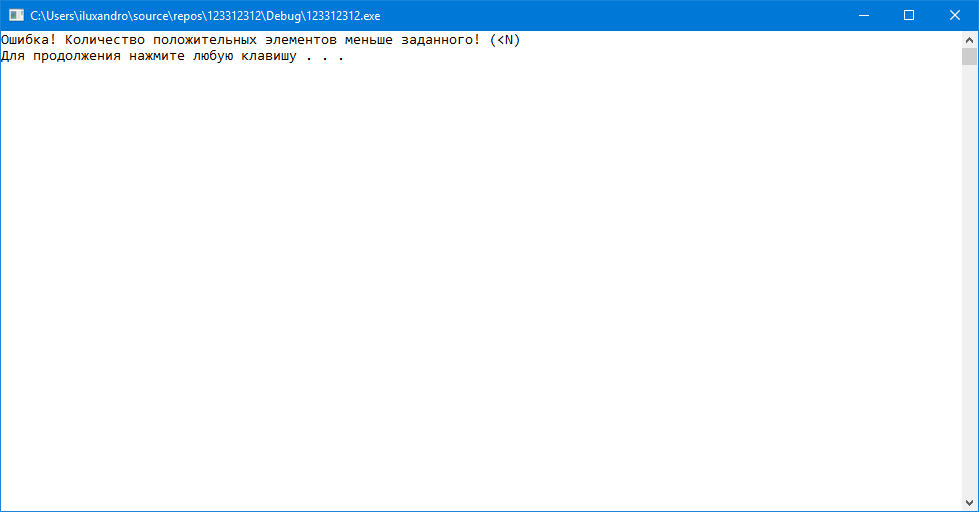
**Цель:** проверить работу программу с количеством положительных элементов < N

**Ввод:**



**Ожидаемые результаты:** Вывод сообщения о ошибке

**Полученные результаты:**



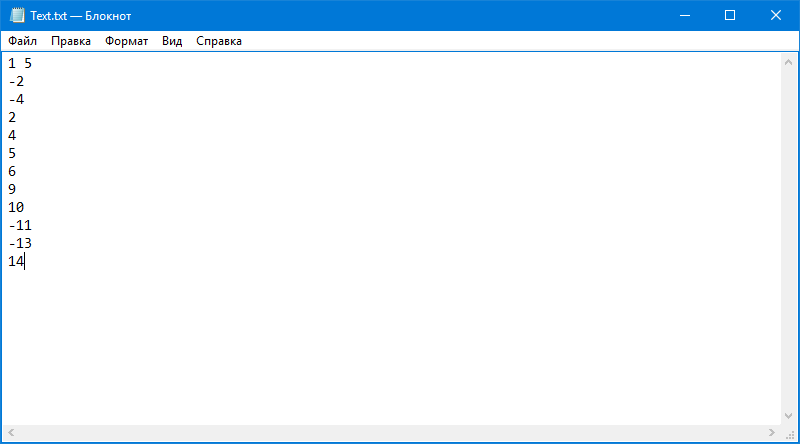
**Вывод**: Полученные результаты равны ожидаемым.

Тест не выявил ошибок. Работа программы корректна

**Тест №9.**

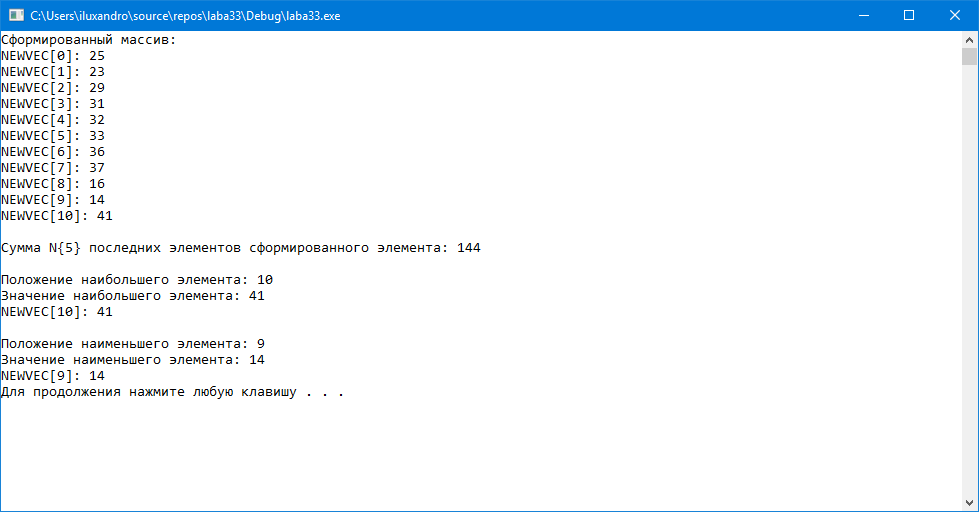
**Цель:** проверить работу программы

**Ввод:**



**Ожидаемые результаты:** Вывод корректных значений

**Полученные результаты:**



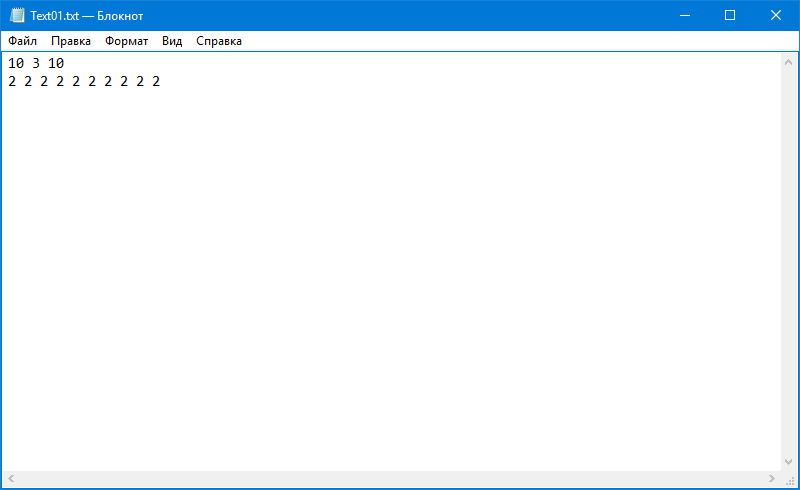
**Вывод**: Полученные результаты равны ожидаемым.

Тест не выявил ошибок. Работа программы корректна

**Тест №10.**

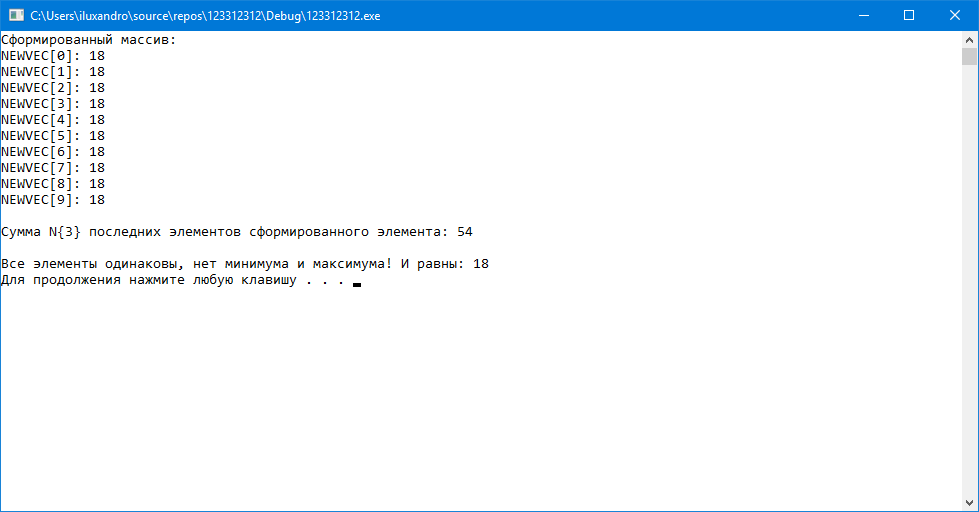
**Цель:** проверить работу программы с одинаковыми элементами массива

**Ввод:**



**Ожидаемые результаты:** Вывод корректных значений и сообщения о отсутствии максимумов и минимумов

**Полученные результаты:**



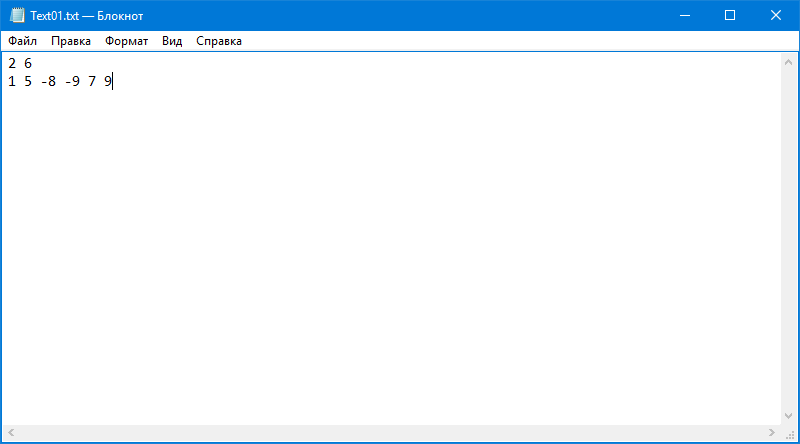
**Вывод**: Полученные результаты равны ожидаемым.

Тест не выявил ошибок. Работа программы корректна

**Тест №11.**

**Цель:** проверить работу программы с N = количеству элементов в массиве

**Ввод:**



**Ожидаемые результаты:** Вывод корректных значений

**Полученные результаты:**



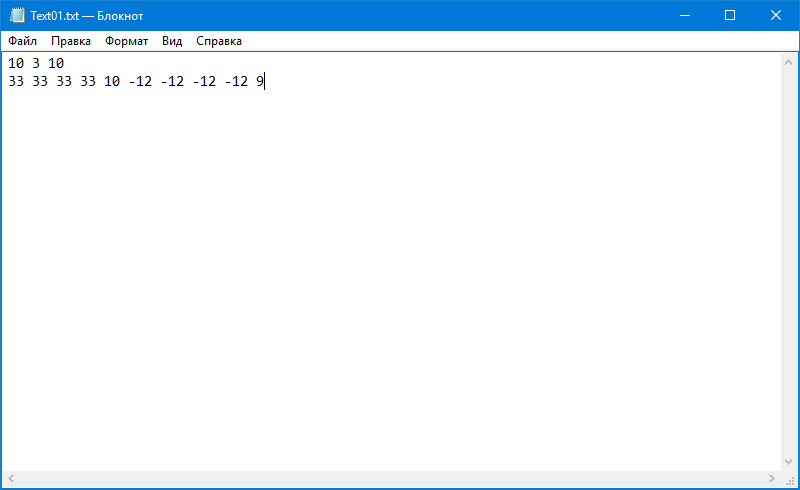
**Вывод**: Полученные результаты равны ожидаемым.

Тест не выявил ошибок. Работа программы корректна

**Тест №12.**

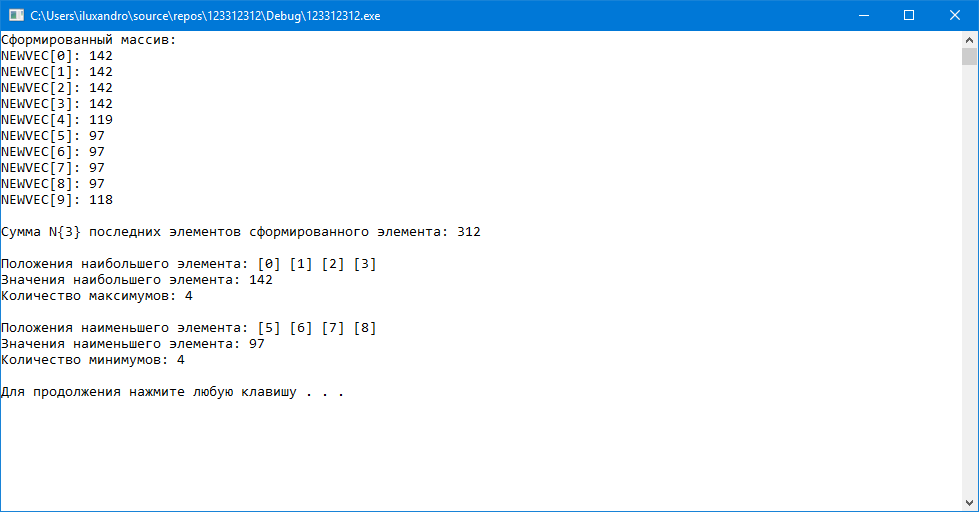
**Цель:** проверить корректность работы с несколькими минимумами и максимумами

**Ввод:**



**Ожидаемые результаты:** Вывод корректных значений

**Полученные результаты:**



**Вывод**: Полученные результаты равны ожидаемым.

Тест не выявил ошибок. Работа программы корректна

**Вывод:**

Разработка программы успешно завершена, так как:

**1)**Полученные результаты совпадают с ожидаемыми.

**2)**Набор тестов считаем полным.