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»

Информатика

Отчет по лабораторной работе № 3

Вариант № 9

Выполнила бригада группы М3О-111Бк-21

Багиров Э. Р.

Нуриев Н. Н.

Проверил Секретарев Виталий Евгеньевич

Москва 2021 г.

Оглавление

[**Задание** 3](#_Toc89954379)

[**Блок-схема алгоритма** 4](#_Toc89954380)

[**Алгоритмический вид программы** 7](#_Toc89954381)

[**Исходный код программы** 9](#_Toc89954382)

[**Некорректные тесты** 12](#_Toc89954383)

[**Корректные тесты** 14](#_Toc89954384)

[**Вывод** 15](#_Toc89954385)

# **Задание**

Кафедра: 304 Курс: ИНФОРМАТИКА

Задание 3: Одномерные массивы

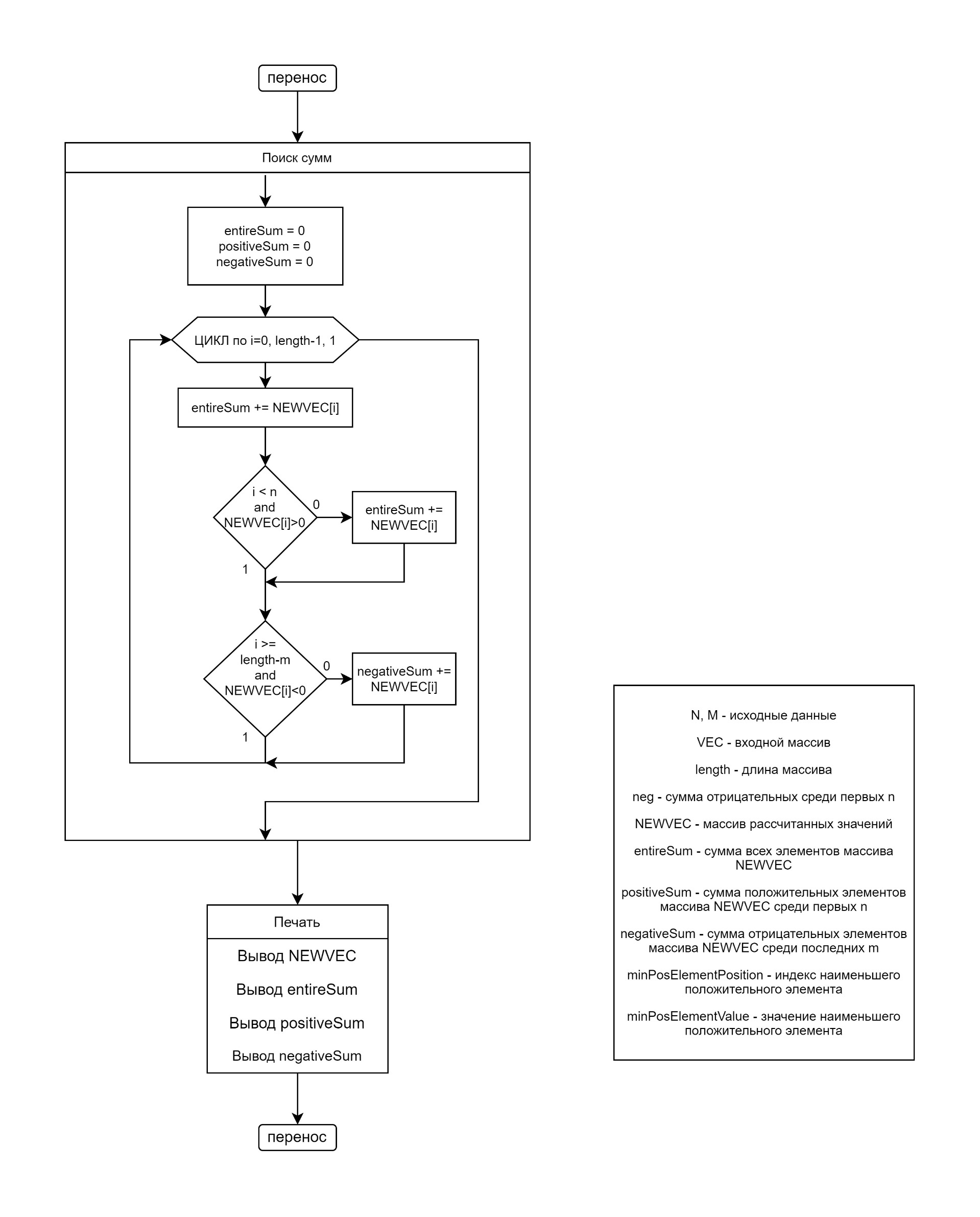
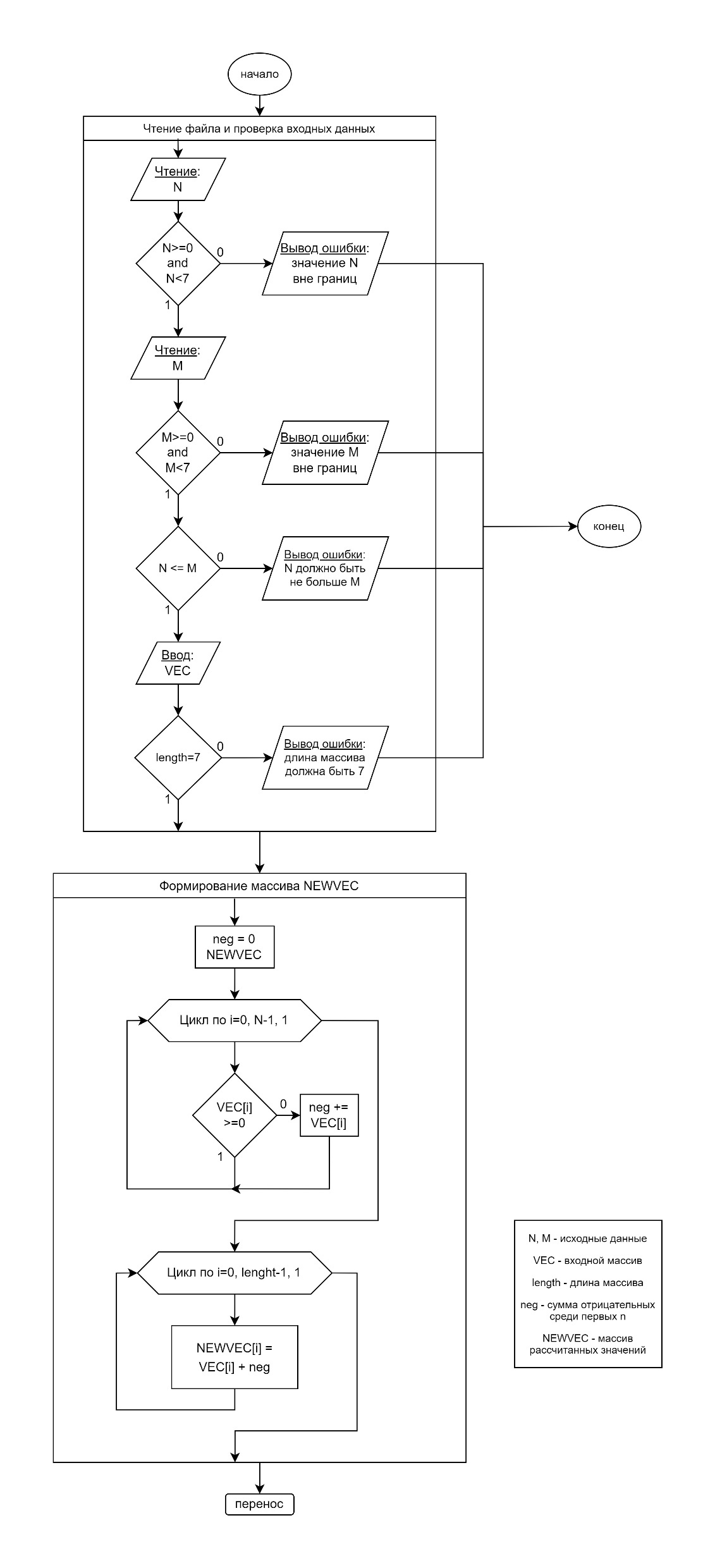
ВАРИАНТ № 9

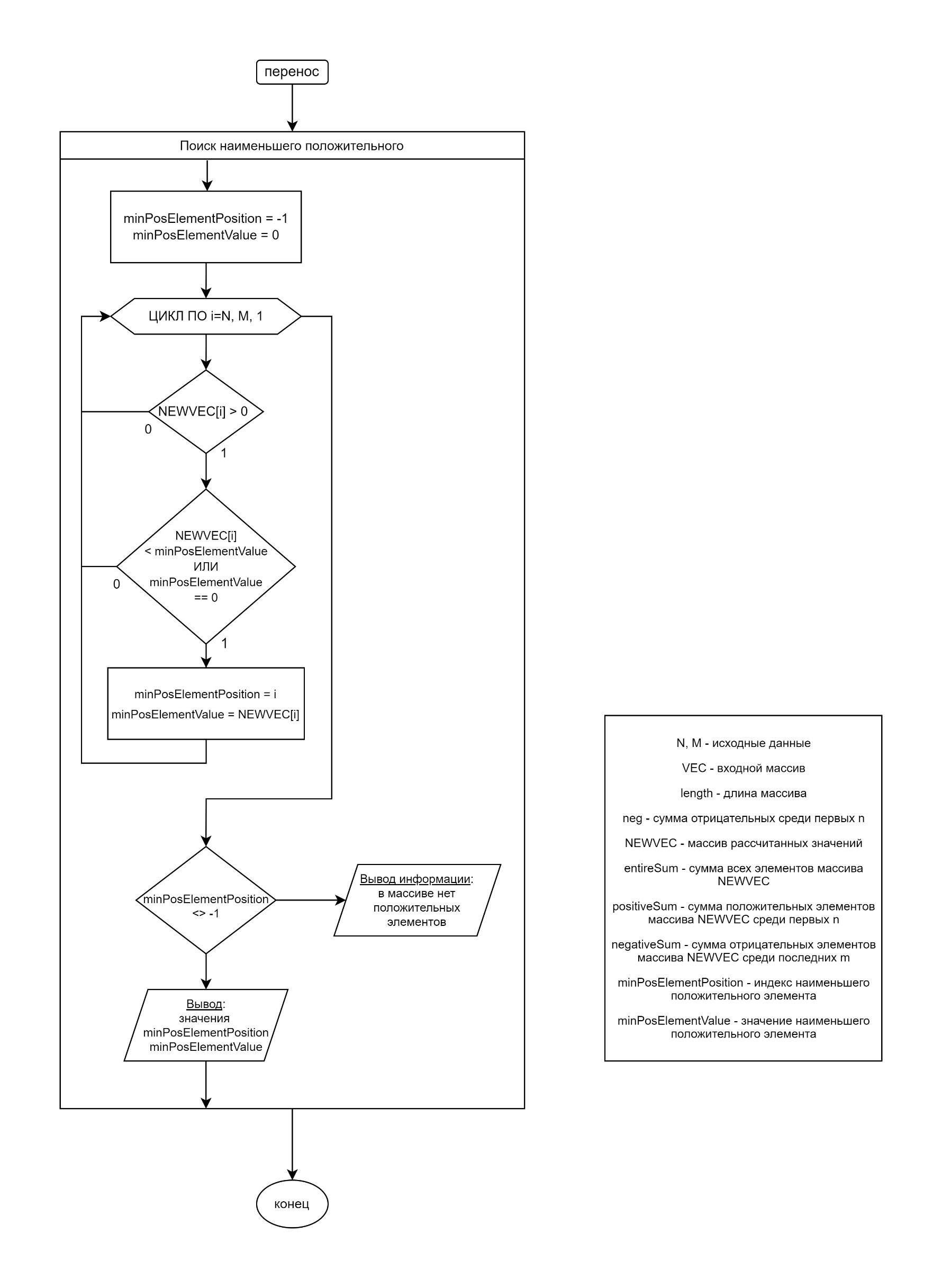
Файл исходных данных содержит значения величин N, M и элементов массива VEC[0],..VEC[6]. Написать, отладить и протестировать программу, выполняющую:

1. Чтение данных из файла.
2. Формирование нового массива NEWVEC, каждый элемент которого есть сумма соответствующего элемента массива VEC со значением, полученным в результате суммирования отрицательных элементов массива VEC среди первых N.
3. Суммирование всех элементов сформированного массива, положительных элементов среди первых N, отрицательных элементов среди последних M.
4. Печать сформированного массива и значений сумм.
5. Печать индекса наименьшего положительного элемента массива NEWVEC (только в диапазоне индексов от N до M), печать индекса и значения указанного элемента.

Программа должна быть параметризована и обеспечивать входной контроль.

# **Блок-схема алгоритма**





# **Алгоритмический вид программы**

Лабораторная работа 3 по информатике

9 вариант

АЛГ Одномерные массивы

ПЕР normal\_length, length, N, M, VEC, NEWVEC, neg, entireSum, positiveSum, negativeSum, minPosElementPosition, minPosElementValue

НАЧАЛО

/\* ОБЪЯВЛЕНИЕ ПЕРЕМЕННЫХ \*/

normal\_length = 7 // нормальная длина массива

N // первое входное число

M // второе входное число

VEC // входной массив

NEWVEC // создаваемый массив

neg = 0 // сумма отрицательных элементов VEC среди первых N

entireSum = 0 // полная сумма элементов NEWVEC

positiveSum = 0 // сумма отрицательных элементов NEWVEC среди первых N

negativeSum = 0 // сумма положительных элементов NEWVEC среди последних M

minPosElementPosition = -1 // расположение наименьшего положительного элемента

minPosElementValue = 0 // значение наименьшего положительного элемента

/\* КОНЕЦ ОБЪЯВЛЕНИЯ ПЕРЕМЕННЫХ \*/

/\* ЧТЕНИЕ ФАЙЛА И ПРОВЕРКА ДАННЫХ \*/

СОЗДАНИЕ ОБЪЕКТА КЛАССА ifstream И СВЯЗЫВАНИЕ ЕГО С ФАЙЛОМ

ЧТЕНИЕ N ИЗ ФАЙЛА

ЕСЛИ КОНЕЦ ФАЙЛА // ПРОВЕРКА НА НАЛИЧИЕ ДАННЫХ В ФАЙЛЕ

ТО ПЕЧАТЬ (сообщение об ошибке)

КОНЕЦ

КОНЕСЛИ

ЕСЛИ N<0 ИЛИ N>7 // ПРОВЕРКА N

ТО ПЕЧАТЬ (сообщение об ошибке)

КОНЕЦ

КОНЕСЛИ

ЧТЕНИЕ M ИЗ ФАЙЛА

ЕСЛИ M<0 ИЛИ M>7 // ПРОВЕРКА M

ТО ПЕЧАТЬ (сообщение об ошибке)

КОНЕЦ

КОНЕСЛИ

ЕСЛИ N>M

ТО ПЕЧАТЬ (сообщение об ошибке)

КОНЕЦ

КОНЕСЛИ

ОБЪЯВЛЕНИЕ ПЕРЕМЕННОЙ length = 0 ДЛЯ ПОДСЧЕТА КОЛИЧЕСТВА ЭЛЕМЕНТОВ В МАССИВЕ

ПОКА НЕ КОНЕЦ ФАЙЛА // СЧИТЫВАНИЕ ДАННЫХ В МАССИВ VEC И СОХРАНЕНИЕ ЕГО ДЛИНЫ В length

НЦ

ЧТЕНИЕ VEC[length] ИЗ ФАЙЛА

УВЕЛИЧЕНИЕ length на 1

КЦ

КОНПОКА

ЕСЛИ length != normal\_length // ПРОВЕРКА ДЛИНЫ МАССИВА VEC

ТО ПЕЧАТЬ (сообщение об ошибке)

КОНЕЦ

КОНЕСЛИ

ЗАКРЫТИЕ ФАЙЛА

/\* КОНЕЦ ЧТЕНИЯ ФАЙЛА И ПРОВЕРКИ ДАННЫХ \*/

/\* ФОРМИРОВАНИЕ МАССИВА NEWVEC \*/

ДЛЯ i ОТ 0 ДО N-1 // НАХОЖДЕНИЕ СУММЫ ОТРИЦАТЕЛЬНЫХ ЭЛЕМЕНТОВ СРЕДИ ПЕРВЫХ N

НЦ

ЕСЛИ VEC[i] < 0

ТО neg += VEC[i]

КОНЕСЛИ

КЦ

ДЛЯ i ОТ 0 до normal\_length - 1 // НАХОЖДЕНИЕ ЭЛЕМЕНТОВ МАССИВА NEWVEC

НЦ

NEWVEC[i] = VEC[i] + neg

КЦ

/\* КОНЕЦ ФОРМИРОВАНИЯ МАССИВА NEWVEC \*/

/\* ПОИСК СУММ \*/

ДЛЯ i ОТ 0 ДО normal\_length - 1 // ПОСИК СУММ

НЦ

entireSum += NEWVEC[i]

ЕСЛИ i < N И NEWVEC[i] > 0

ТО positiveSum += NEWVEC[i]

КОНЕСЛИ

ЕСЛИ i>=normal\_length - M И NEWVEC[i] < 0

ТО negativeSum += NEWVEC[i]

КОНЕСЛИ

КЦ

/\* КОНЕЦ ПОИСКА СУММ \*/

/\* ПЕЧАТЬ NEWVEC И СУММ \*/

ПЕЧАТЬ МАССИВА NEWVEC

ПЕЧАТЬ entireSum

ПЕЧАТЬ positiveSum

ПЕЧАТЬ negativeSum

/\* КОНЕЦ ПЕЧАТИ NEWVEC И СУММ \*/

/\* ПОИСК НАИМЕНЬШЕГО ПОЛОЖИТЕЛЬНОГО \*/

ДЛЯ i ОТ N ДО M // ПОИСК НАИМЕШЬШЕГО ПОЛОЖИТЕЛЬНОГО ЭЛЕМЕНТА

НЦ

ЕСЛИ NEWVEC[i]>0

ТО ЕСЛИ NEWVEC[i] < minPosElementValue ИЛИ minPosElementValue = 0

ТО minPosElementValue = NEWVEC[i]

minPosElementPosition = i

КОНЕСЛИ

КОНЕСЛИ

КЦ

ЕСЛИ minPosElementPosition != -1 // ПРОВЕРКА НА НАЛИЧИЕ НАИМЕНЬШЕГО ПОЛОЖИТЕЛЬНОГО

ТО ПЕЧАТЬ minPosElementPosition И minPosElementValue

ИНАЧЕ ПЕЧАТЬ (В заданном интервале нет положительных элементов)

/\* КОНЕЦ ПОИСКА НАИМЕНЬШЕГО ПОЛОЖИТЕЛЬНОГО \*/

КОНЕЦ

# **Исходный код программы**

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

\* КАФЕДРА № 304 1 КУРС CPP ЛАБОРАТОРНАЯ РАБОТА \*

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

\* Project Type : Win32 Console Application \*

\* Project Name : lr3 \*

\* File Name : lr3.cpp \*

\* Language : C/C++ \*

\* Programmer(s) : Багиров Э., Нуриев Н. \*

\* Modifyed By : \*

\* Created : 11/28/21 \*

\* Last Revision : 12/09/21 \*

\* Comment(s) : Код к ЛР№3 - "Одномерные массивы" \*

\* \*

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

#include <iostream>

#include <fstream>

using namespace std;

void PrintArray(int array[], int length) {

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

cout << array[i] << " ";

}

cout << endl;

}

int main()

{

setlocale(LC\_ALL, "Russian"); // подключение русского языка

system("color F0"); // приведение цвета консоли к каноничному

/\* ОБЪЯВЛЕНИЕ ПЕРЕМЕННЫХ \*/

const int normal\_length = 7; // нормальная длина массива

int N; // первое входное число

int M; // второе входное число

int VEC[normal\_length]; // входной массив

int NEWVEC[normal\_length]; // создаваемый массив

int neg; // сумма отрицательных элементов VEC среди первых n

int entireSum; // полная сумма элементов массива NEWVEC

int positiveSum; // сумма отрицательных элементов NEWVEC среди первых n

int negativeSum; // сумма положительных элементов NEWVEC среди последних m

int minPosElementPosition; // расположение наименьшего положительного элемента

int minPosElementValue; // значение наименьшего положительного элемента

// имя файла

const string filename = "cor.test.2.txt";

/\* КОНЕЦ ОБЪЯВЛЕНИЯ ПЕРЕМЕННЫХ \*/

/\* ЧТЕНИЕ ФАЙЛА И ПРОВЕРКА ДАННЫХ \*/

ifstream fl; // создание объекта класса ifstream

fl.open(filename); // связывание объекта с файлом

fl >> N; // сохранение первого входного числа в N

// проверка на наличие данных в файле

if (fl.eof()) {

cout << "Ошибка: файл пуст.\n";

return 1;

}

if (fl.fail()) //в файле не число

{

cout << "Ошибка: сбой при чтении файла.\n" << endl;

fl.close(); //закрыть файл

return 3; //выход по ошибке

}

// проверка N

if (N < 0 || N > 7) {

cout << "Ошибка: значение N вне границ.\n";

return 2;

}

fl >> M; // сохранение второго входного числа в M

// проверка M

if (M < 0 || M>7) {

cout << "Ошибка: значение M вне границ.\n";

return 2;

}

// взаимная проверка N, M

if (N > M) {

cout << "Ошибка: N должно быть не больше M.\n";

return 3;

}

int length = 0; // для подсчета количества элементов в массиве

// считывание данных в массив VEC и сохранение его длины в length

while (!fl.eof()) {

fl >> VEC[length];

length++;

}

if (length != normal\_length) {

cout << "Ошибка: длина массива должна быть равна 7.\n";

return 4;

}

fl.close(); // отпуск файла

PrintArray(VEC, length); // эхо-печать входного массива

/\* КОНЕЦ ЧТЕНИЯ ФАЙЛА И ПРОВЕРКА ДАННЫХ \*/

/\* ФОРМИРОВАНИЕ МАССИВА NEWVEC \*/

neg = 0;

// нахождение суммы отрицательных элементов среди первых N

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

if (VEC[i] < 0) {

neg += VEC[i];

}

}

// нахождение элементов массива NEWVEC

for (int i = 0; i < normal\_length; i++) {

NEWVEC[i] = VEC[i] + neg;

}

/\* КОНЕЦ ФОРМИРОВАНИЯ МАССИВА NEWVEC \*/

/\* ПОИСК СУММ \*/

entireSum = 0;

positiveSum = 0;

negativeSum = 0;

for (int i = 0; i < normal\_length; i++) {

entireSum += NEWVEC[i]; // прибавление любого элемента к полной сумме

// проверка элемента на позицию и значение

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

positiveSum += NEWVEC[i];

}

// проверка элемента на позицию и значение

if (i >= normal\_length - M && NEWVEC[i] < 0) {

negativeSum += NEWVEC[i];

}

}

/\* КОНЕЦ ПОИСКА СУММ \*/

/\* ПЕЧАТЬ NEWVEC И СУММ\*/

PrintArray(NEWVEC, normal\_length);

cout << "entireSum: " << entireSum << "\n";

cout << "positiveSum: " << positiveSum << "\n";

cout << "negativeSum: " << negativeSum << "\n";

/\* КОНЕЦ ПЕЧАТИ NEWVEC И СУММ\*/

/\* ПОИСК НАИМЕНЬШЕГО ПОЛОЖИТЕЛЬНОГО \*/

minPosElementPosition = -1;

minPosElementValue = 0;

if (M == 7) {

M = 6;

}

if (N == 7) {

N = 6;

}

for (int i = N; i <= M; i++) {

if (NEWVEC[i] > 0) {

if (NEWVEC[i] < minPosElementValue || minPosElementValue == 0) {

minPosElementValue = NEWVEC[i];

minPosElementPosition = i;

}

}

}

cout << "Наименьший положительный элемент в интервале [" << N << ";" << M << "):\n";

if (minPosElementPosition != -1) {

cout << "индекс - " << minPosElementPosition << "\nзначение - "

<< minPosElementValue << "\n";

}

else {

cout << "В заданном интервале нет положительных элементов";

}

/\* КОНЕЦ ПОИСКА НАИМЕНЬШЕГО ПОЛОЖИТЕЛЬНОГО \*/

return 0;

}

# **Некорректные тесты**

**Тест №1**

Цель теста: проверить работу программы при пустом файле

Исходные данные: (пустой файл)

Ожидаемый результат: вывод сообщения об ошибке: «Ошибка: файл пуст.»

Полученный результат:



Вывод: полученный результат совпал с ожидаемым. Тест ошибки не обнаружил.

**Тест №2**

Цель теста: проверить работу программы с некорректными данными в файле

Исходные данные: ф а 1 2 3 4 5 6 7

Ожидаемый результат: вывод сообщения об ошибке: «Ошибка: сбой при чтении файла.»

Полученный результат:



Вывод: полученный результат совпал с ожидаемым. Тест ошибки не обнаружил.

**Тест №3**

Цель теста: проверить работу программы в некорректной области данных

Исходные данные: 10 5 1 2 3 4 5 6 7

Ожидаемый результат: вывод сообщения об ошибке: «Ошибка: значение N вне границ.»

Полученный результат:



Вывод: полученный результат совпал с ожидаемым. Тест ошибки не обнаружил.

**Тест №4**

Цель теста: проверить работу программы в некорректной области данных

Исходные данные: 5 10 1 2 3 4 5 6 7

Ожидаемый результат: вывод сообщения об ошибке: «Ошибка: значение M вне границ.»

Полученный результат:



Вывод: полученный результат совпал с ожидаемым. Тест ошибки не обнаружил.

**Тест №5**

Цель теста: проверить работу программы в некорректной области данных

Исходные данные: 6 5 1 2 3 4 5 6 7

Ожидаемый результат: вывод сообщения об ошибке: «Ошибка: N должно быть не больше M.»

Полученный результат:



Вывод: полученный результат совпал с ожидаемым. Тест ошибки не обнаружил.

**Тест №6**

Цель теста: проверить работу программы в некорректной области данных

Исходные данные: 5 6 1 2 3 4 5 6

Ожидаемый результат: вывод сообщения об ошибке: «Ошибка: длина массива должна быть равна 7.»

Полученный результат:



Вывод: полученный результат совпал с ожидаемым. Тест ошибки не обнаружил.

# **Корректные тесты**

**Тест №1**

Цель теста: проверить работу программы в корректной области исходных данных

Исходные данные: 5 6 1 2 3 4 5 6 7

Ожидаемый результат:

VEC: 1 2 3 4 5 6 7

NEWVEC: 1 2 3 4 5 6 7

entireSum: 28

positiveSum: 15

negativeSum: 0

Наименьший положительный элемент в интервале [5;6]:

индекс - 5

значение - 6

Расчет ожидаемого результата:

1. Прочитаны данные из файла: N=5, M=6, VEC={1,2,3,4,5,6,7}. Эхо-печать VEC: 1 2 3 4 5 6 7
2. Сумма отрицательных элементов среди первых N neg=0

Сформирован массив NEWVEC: 1 2 3 4 5 6 7

1. Полная сумма элементов NEWVEC entireSum=1+2+3+4+5+6+7=28

Сумма положительных элементов среди первых N positiveSum=1+2+3+4+5=15

Сумма отрицательных элементов среди последних M элементов negativeSum=0

1. Печать NEWVEC, entireSum, positiveSum, negativeSum
2. Найден наименьший положительный элемент в промежутке [5;6] – значение 6, индекс - 5

Полученный результат:

Изображение выглядит как текст

Автоматически созданное описание

Вывод: полученный результат совпал с ожидаемым. Тест ошибки не обнаружил.

**Тест №2**

Цель теста: проверить работу программы в корректной области исходных данных

Исходные данные: 2 4 1 -2 3 -4 5 -6 7

Ожидаемый результат:

VEC: 1 -2 3 -4 5 -6 7

NEWVEC: -1 -4 1 -6 3 -8 5

entireSum: -10

positiveSum: 0

negativeSum: -14

Наименьший положительный элемент в интервале [2;4]:

индекс - 2

значение – 1

Расчет ожидаемого результата:

1. Прочитаны данные из файла: N=2, M=4, VEC={1,-2,3,-4,5,-6,7}. Эхо-печать VEC: 1 -2 3 -4 5 -6 7 Сумма отрицательных элементов среди первых N neg=-2

Сформирован массив NEWVEC: -1 -4 1 -6 3 -8 5

1. Полная сумма элементов NEWVEC entireSum=-1-4+1-6+3-8+5=-10

Сумма положительных элементов среди первых N positiveSum=0

Сумма отрицательных элементов среди последних M элементов negativeSum=-6-8=-14

1. Печать NEWVEC, entireSum, positiveSum, negativeSum
2. Найден наименьший положительный элемент в промежутке [2;4] – значение 1, индекс - 2

Полученный результат:

Изображение выглядит как текст

Автоматически созданное описание

Вывод: полученный результат совпал с ожидаемым. Тест ошибки не обнаружил.

**Тест №3**

Цель теста: проверить работу программы в корректной области исходных данных

Исходные данные: 1 5 -1 -2 -3 -4 -5 -6 -7

Ожидаемый результат:

VEC: -1 -2 -3 -4 -5 -6 -7

NEWVEC: -2 -3 -4 -5 -6 -7 -8

entireSum: -35

positiveSum: 0

negativeSum: -30

Наименьший положительный элемент в интервале [1;5]:

В заданном интервале нет положительных элементов

Расчет ожидаемого результата:

1. Прочитаны данные из файла: N=1, M=5, VEC={-1,-2,-3,-4,-5,-6,-7}.

Эхо-печать VEC: -1 -2 -3 -4 -5 -6 -7. Сумма отрицательных элементов среди первых N neg=-1

Сформирован массив NEWVEC: -2 -3 -4 -5 -6 -7 -8

1. Полная сумма элементов NEWVEC entireSum=-2-3-4-5-6-7-8=-35

Сумма положительных элементов среди первых N positiveSum=0

Сумма отрицательных элементов среди последних M элементов negativeSum=-4-5-6-7-8=-30

1. Печать NEWVEC, entireSum, positiveSum, negativeSum
2. Не найден наименьший положительный элемент в промежутке [1;5]

Полученный результат:

Изображение выглядит как текст

Автоматически созданное описание

Вывод: полученный результат совпал с ожидаемым. Тест ошибки не обнаружил.

# **Вывод**

Проанализирован алгоритм по работе с элементами одномерного массива. В ходе выполнения лабораторной работы были составлены блок-схема и псевдокод для дальнейшего написания программы. Проведены тесты для проверки соответствия исходных данных и корректные тесты. В результате проведения тестов ошибок в программе не обнаружено. Разработка программы завершена на основании того, что полученные результаты совпали с ожидаемыми, набор тестов считаем полным.