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](#_Toc86736331)

[**Диаграмма алгоритма** 4](#_Toc86736332)

[**Алгоритмический вид программы** 5](#_Toc86736333)

[**Исходный код программы** 6](#_Toc86736334)

[**Некорректные тесты** 7](#_Toc86736335)

[**Корректные тесты** 8](#_Toc86736336)

[**Протокол расчета ожидаемого результата** 11](#_Toc86736337)

[**Вывод** 12](#_Toc86736338)

# **Задание**

Кафедра: 304 Курс: ИНФОРМАТИКА

Задание 3: Одномерные массивы

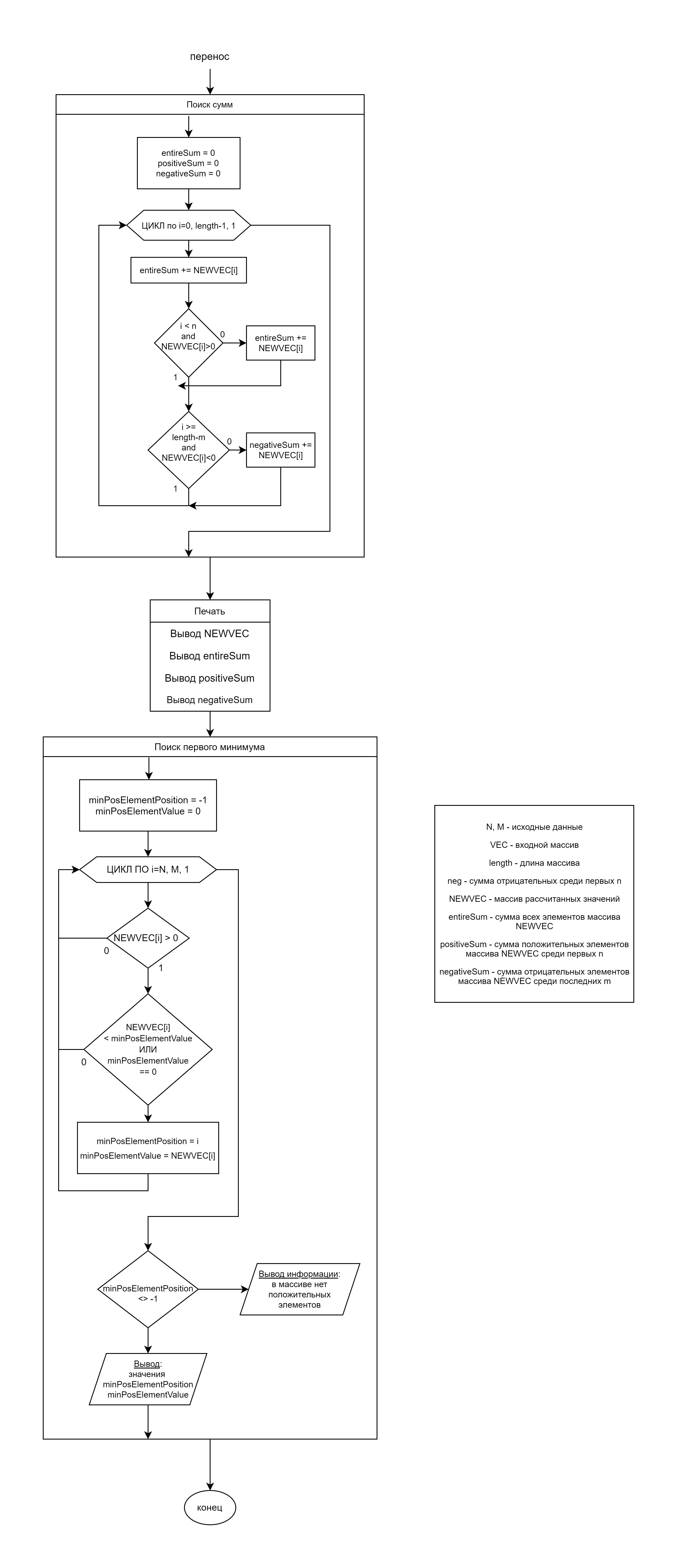
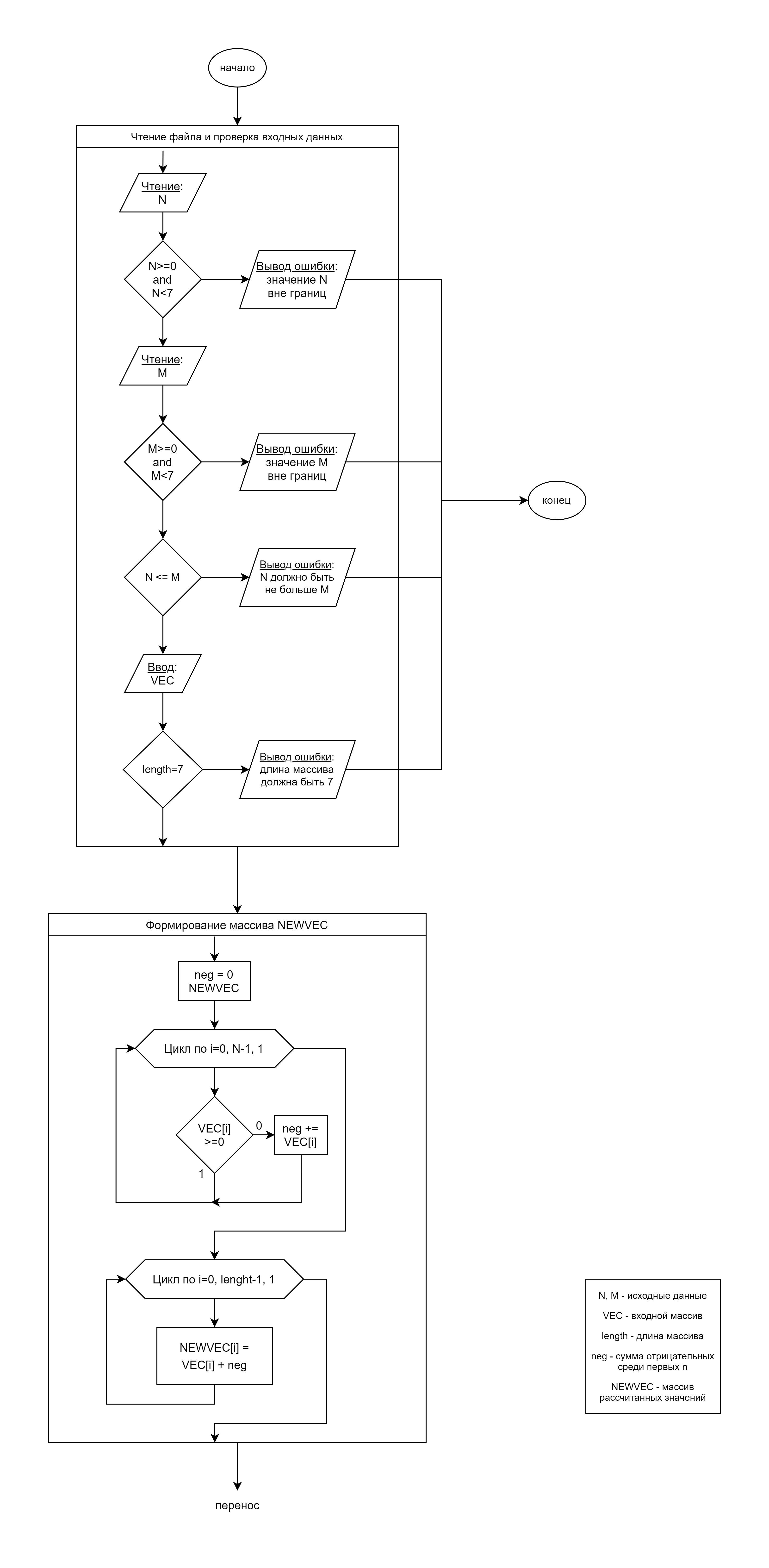
ВАРИАНТ № 9

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.

# **Диаграмма алгоритма**



# **Алгоритмический вид программы**

Лабораторная работа 2 по информатике

9 вариант

АЛГ Задание 2: Табулирование функции

ПЕР A, B, N, x, h, i

НАЧАЛО

//ИНИЦИАЛИЗАЦИЯ ПЕРЕМЕННЫХ

А // координата начала отрезка [A;B]

В // координата конца отрезка [A;B]

N // количество интервалов разделения [A;B]

h // длина шага

x // аргумент приращения

i // индекс приращения

//ВВОД ДАННЫХ

ВВОД A, B

//ПРОВЕРКА ВХОДНЫХ ДАННЫХ

ЕСЛИ А > B

ТО ПЕЧАТЬ (сообщение об ошибке)

КОНЕЦ

КОНЕСЛИ

ЕСЛИ A <> B

ТО

ВВОД N

ЕСЛИ N <= 0

ТО ПЕЧАТЬ (сообщение об ошибке)

КОНЕЦ

КОНЕСЛИ

h = (B - A) / N // расчет величины шага

ИНАЧЕ

N = 0

h = 0

КОНЕСЛИ

ДЛЯ i ОТ 0 ДО N

НЦ

x = A + i \* h //Приращение аргумента

F = (x - 1) \*\* 3 //Вычисление F(x)

G = ((х + 5) \*\* 3) / (1 + sin(x) \*\* 2) //Вычисление G(x)

//ВЫВОД

ВЫВОД В ТАБЛИЦЕ i, x, F, G

КЦ

КОНЕЦ

# **Исходный код программы**

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

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

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

\* Project Type : Win32 Console Application \*

\* Project Name : laba2 \*

\* File Name : laba2.cpp \*

\* Language : C/C++ \*

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

\* Modifyed By : \*

\* Created : 11/10/21 \*

\* Last Revision : 11/24/21 \*

\* Comment(s) : Код к ЛР№2 - "Табулирование функций" \*

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

#include <iostream>

#include <iomanip>

using namespace std;

// псевдографика - заголовок таблицы

void HeadTable() {

cout << "\t"

<< char(218) << setfill(char(196)) << setw(20)

<< char(194) << setfill(char(196)) << setw(20)

<< char(194) << setfill(char(196)) << setw(20)

<< char(194) << setfill(char(196)) << setw(20)

<< char(191) << endl;

cout << "\t"

<< char(179) << setfill(' ') << setw(19)<< "#"

<< char(179) << setfill(' ') << setw(19)<< "x"

<< char(179) << setfill(' ') << setw(19)<< "F(x)"

<< char(179) << setfill(' ') << setw(19)<< "G(x)"

<< char(179) << endl;

cout << "\t"

<< char(195) << setfill(char(196)) << setw(20)

<< char(197) << setfill(char(196)) << setw(20)

<< char(197) << setw(20)

<< char(197) << setw(20) << char(180) << endl;

cout.setf(ios::showpoint | ios::fixed);

}

// псевдографика - завершение таблицы

void BottomTable() {

cout << "\t"

<< char(192) << setfill(char(196)) << setw(20)

<< char(193) << setfill(char(196)) << setw(20)

<< char(193) << setfill(char(196)) << setw(20)

<< char(193) << setfill(char(196)) << setw(20)

<< char(217) << endl;

}

// псевдографика - тело таблицы

void BetweenTheRaws() {

cout << "\t"

<< char(195) << setfill(char(196)) << setw(20)

<< char(197) << setfill(char(196)) << setw(20)

<< char(197) << setfill(char(196)) << setw(20)

<< char(197) << setfill(char(196)) << setw(20)

<< char(180) << endl;

}

// имплемантация возведения числа в степень

double power(double a, int n) {

double b = a;

for (int i = 1; i < n; i++) {

b \*= a;

}

return b;

}

// расчет значения функции F(x) в точке x

double F(double x) {

return power(x - 1, 3);

}

// расчет значения функции G(x) в точке x

double G(double x) {

return power(x + 5, 3) / (1 + power(sin(x), 2));

}

// вывод строк таблицы с автоматическим закрытием таблицы

void PrintTable(int n, double x, double f, double g, bool last) {

cout << "\t" << char(179) << setw(19) << setfill(char(255)) << n

<< char(179) << setw(19) << setprecision(4) << x

<< setfill(char(255))

<< char(179) << setw(19) << setprecision(4) << f

<< setfill(char(255))

<< char(179) << setw(19) << setprecision(4) << g

<< setfill(char(255)) << char(179) << endl;

if (last) {

BottomTable();

}

else {

BetweenTheRaws();

}

}

int main()

{

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

system("color F0");

//Инициализация переменных

double A; // начало отрезка

double B; // конец отрезка

int N; // количество интервалов, на которое делится отрезок [A;B]

double h; // шаг

double x; // аргумент приращения

int i; // индекс приращения

// Ввод и проверка данных

cout << "Введите А - координату начала отрезка [A;B]: \t";

cin >> A;

cout << "Введите B - координату конца отрезка [A;B]: \t";

cin >> B;

if (A > B) {

cout << "\nОШИБКА! А должен быть меньше или равен В!";

return 1; // выход из программы возвращением кода 1

}

if (A != B) {

cout << "Введите N - количество разделений отрезка [A;B]: \t";

cin >> N;

if (N <= 0) {

cout << "\nОШИБКА! N должно быть больше 0! ";

return 2; // выход из программы возвращением кода 2

}

h = (B - A) / N; // Расчет величины шага

}

else {

// Если входной отрезок есть точка, изменить количество разделений

N = 0;

h = 0;

}

cout << "Величина шага h = " << h << "\n";

// Подготовка таблицы

setlocale(LC\_ALL, "C"); // отключение русского языка

HeadTable(); // вывод заголовка таблицы

for (i = 0; i <= N; i++) { // посторная печать таблицы в цикле

x = A + i \* h; // вычисление инкремируемой переменной

double Fx = F(x); // расчет значения функции F(x) в точке x

double Gx = G(x); // расчет значения функции G(x) в точке x

bool last = i == N; // флаг последней итерации цикла

// функции-принтеру передаются значения:

// i - индекс итерации

// x - аргумент приращения

// Fx - значение функции F(x) в точке x

// Gx - значение функции G(x) в точке x

// last - флаг последней итерации цикла

PrintTable(i, x, Fx, Gx, last);

}

return 0; // выход из программы возвращением кода 0

}

# **Некорректные тесты**

**Тест №1**

Цель теста: проверить работу программы на границе некорректной области исходных данных

Исходные данные: A = 5, B = 2

Ожидаемый результат: вывод сообщения об ошибке:“ ОШИБКА! А должен быть меньше или равен В!”

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 некорректной области исходных данных

Исходные данные: A = 5, B = 10, N = -1

Ожидаемый результат: вывод сообщения об ошибке: “ОШИБКА! N должно быть больше 0!”

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 на границе корректной области исходных данных

Исходные данные: A = 4, B = 100, N = 0

Ожидаемый результат: вывод сообщения об ошибке: “ОШИБКА! N должно быть больше 0!”

Полученный результат:

Изображение выглядит как текст

Автоматически созданное описание

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

# **Корректные тесты**

**Тест №1**

Цель теста: проверить работу программы на границе корректной области исходных данных

Исходные данные: A = 4, B = 4

Ожидаемый результат: вывод значения функций F и G в точке x=4

Ожидаемые значения функций:

F(4) = (4-1)^3 = 3^3 = 27

G(4) = (4+5)^3 / (1+sin^2(4))= 9^3 / (1+0.0698^2) = 729/1.57275 = 463.5193

Полученный результат:

Изображение выглядит как текст, стол

Автоматически созданное описание

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

**Тест №2**

Цель теста: проверить работу программы в корректной области

Исходные данные: A = 2, B = 5, N = 3

Ожидаемый результат:

Вывод таблицы:

|  |  |  |  |
| --- | --- | --- | --- |
| # | x | F(x) | G(x) |
| 0 | 2 | 1 |  |
| 1 | 3 |  |  |
| 2 | 4 |  |  |
| 3 | 5 |  |  |

Полученный результат:

Изображение выглядит как стол

Автоматически созданное описание

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

**Тест №3**

Цель теста: проверить работу программы в корректной области

Исходные данные: А = 0, В = π, N = 10

Ожидаемый результат:

Вывод таблицы:

|  |  |  |  |
| --- | --- | --- | --- |
| # | x | F(x) | G(x) |
| 0 | 0 |  |  |
| 1 | 0,3142 |  |  |
| 2 | 0,6284 |  |  |
| 3 | 0,9425 |  |  |
| 4 | 1,2566 |  |  |
| 5 | 1,5708 |  | 141,56146 |
| 6 | 1,8850 |  | 171,39555 |
| 7 | 2,1991 |  | 225,57507 |
| 8 | 2,5133 |  | 315,32855 |
| 9 | 2,8274 |  | 437,92007 |
| 10 | 3,1416 |  | 539,75071 |

Полученный результат:

Изображение выглядит как стол

Автоматически созданное описание

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

**Тест №4**

Цель теста: проверить работу программы в корректной области

Исходные данные: А = 12, В = 27, N = 5

Ожидаемый результат:

Вывод таблицы:

|  |  |  |  |
| --- | --- | --- | --- |
| # | x | F(x) | G(x) |
| 0 | 12 | 1331 |  |
| 1 | 15 | 2744 | 5622,42226 |
| 2 | 18 | 4913 | 7779,50207 |
| 3 | 21 | 8000 |  |
| 4 | 24 | 12167 | 13400,01809 |
| 5 | 27 |  |  |

Полученный результат:

Изображение выглядит как стол

Автоматически созданное описание

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

# **Вывод**

Проанализирован алгоритм по разбитию отрезка на интервалы, вычислению значения функций в найденных концах интервалов, форматированию выходных данных в виде таблицы. 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.