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 № 1

Вариант № 7

Выполнила бригада группы М3О-111Бк-21

Багиров Э. Р.

Нуриев Н. Н.

Проверил \*\*\*

Москва 2021 г.

**Задание**

Кафедра: 304 Курс: ИНФОРМАТИКА

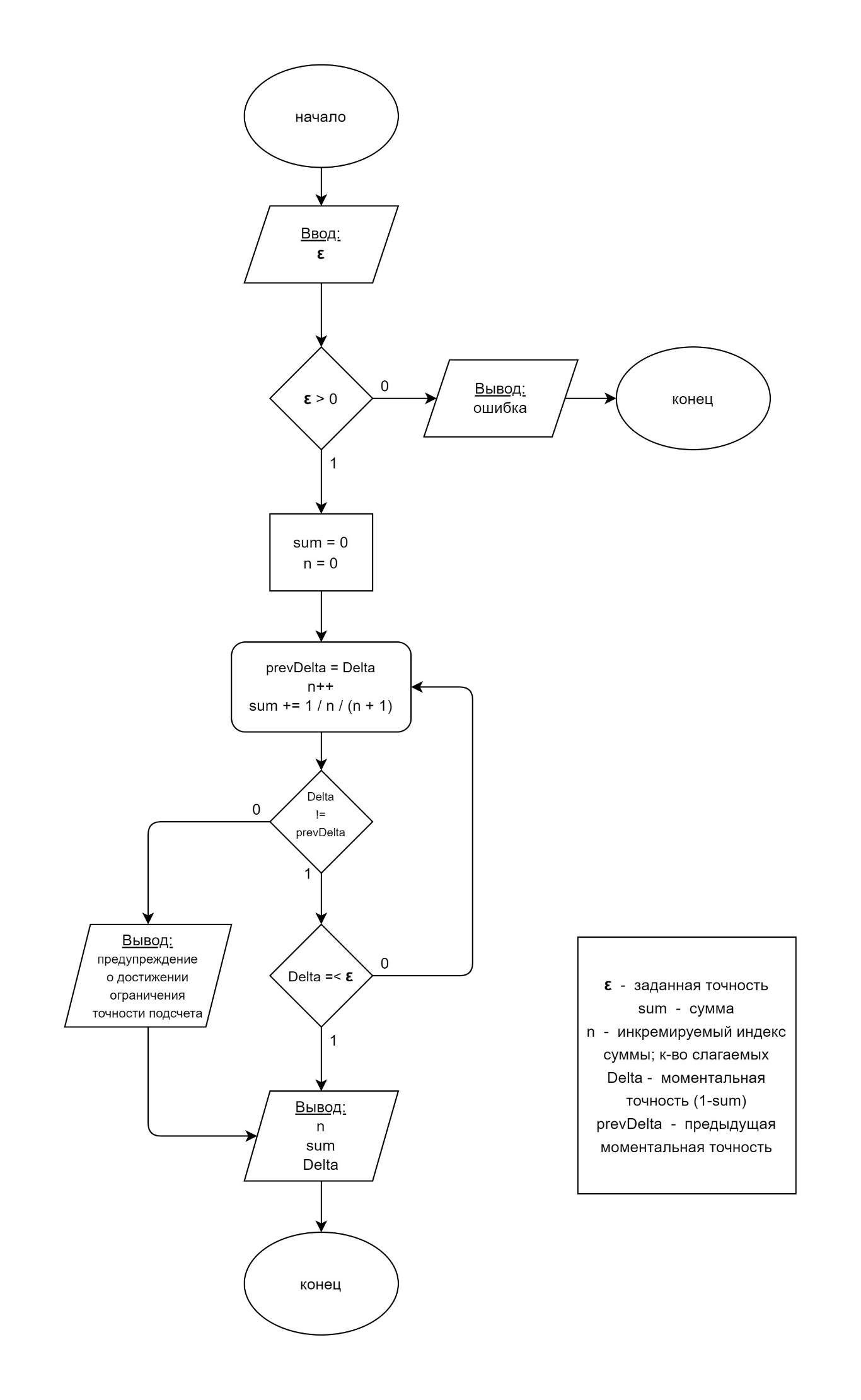
Задание 1: Вычисление суммы бесконечного числового ряда

ВАРИАНТ № 7

Определить с заданной точностью сумму бесконечного степенного ряда:

Предусмотреть ввод точности и печать: количества просуммированных элементов, суммы, разности суммы и точного значения, которое равно: **1**

**Диаграмма алгоритма**



**Исходный код программы**

#include <iostream>

#include <iomanip>

using namespace std;

int main()

{

setlocale(LC\_ALL, "Russian");

// Инициализация переменных

long double Eps; // заданная пользователем точность подсчета Эпсилон

double Sum = 0; // сумма

unsigned long long N = 0; // индекс суммы

double prevDelta = 1; // моментальная точность предыдущей итерации

bool interrupted = false; // флаг о выходе из цикла до достижения необходимой точности

// Входной контроль

cout << "Введите Eps: ";

cin >> Eps;

cout << "\n";

if (Eps <= 0) {

// Предотвращение некорректного ввода точности подсчета

cout << "ОШИБКА - Eps должен быть больше 0." << endl;

return 0;

}

// Расчет суммы

do {

N++; // инкремент индекса

Sum += 1.0 / N / (N + 1); // подсчет нового слагаемого, прибавление его к сумме

if (prevDelta + Sum - 1 == 0) {

interrupted = true; // Если точность не увеличилась за последнюю итерацию,

break; // достигнут предел точности, необходимо завершить цикл

}

prevDelta = 1 - Sum; // сохранение новой моментальной точности

} while (prevDelta > Eps);

// Вывод результатов

cout.precision(60);

if (interrupted) {

cout << fixed << "Достигнуто ограничение по точности подсчета\nРазность достигнутой точности Delta и Eps = " << 1 - Sum - Eps << endl;

}

cout << "Cумма получена\nКоличество просумированных элементов N = " << N << endl;

cout << "Сумма Sum = " << Sum << endl;

cout << fixed << "Достигнутая точность Delta = " << (1 - Sum) << endl;

return 1;

}

**Алгоритмический вид программы**

АЛГ Вариант №7

ПЕР N, Sum, Eps, interrupted, prevDelta

НАЧАЛО

// Инициализация переменных

Eps // заданная пользователем точность подсчета Эпсилон

Sum = 0 // сумма

N = 0 // индекс суммы

prevDelta = 1 // моментальная точность предыдущей итерации

interrupted = false // флаг выхода из цикла до достижения необходимой точности

// Входной контроль

ВВОД Eps

ЕСЛИ Eps <= 0

// Предотвращение некорректного ввода точности подсчета

ТО ПЕЧАТЬ (сообщение об ошибке)

КОНЕЦ

КОНЕСЛИ

// Расчет суммы

НЦ ПОВТОРЯТЬ

N = N + 1 // инкремент индекса

Sum += 1.0 / N / (N + 1) // подсчет нового слагаемого, прибавление его к сумме

ЕСЛИ (prevDelta + Sum - 1 == 0)

// Если точность не увеличилась за последнюю итерацию,

// достигнут предел точности, необходимо завершить цикл

ТО

interrupted = true

ПРЕРВАТЬ ЦИКЛ

КОНЕСЛИ

prevDelta = 1 - Sum // сохранение новой моментальной точности

КЦ ПОКА (1 - Sum) > Eps

// Вывод результатов

ЕСЛИ (interrupted == ИСТИНА)

ТО ПЕЧАТЬ (сообщение о достижении ограничения точности)

КОНЕСЛИ

ПЕЧАТЬ (N, Sum, 1 - Sum)

КОНЕЦ

**Некорректные тесты.**

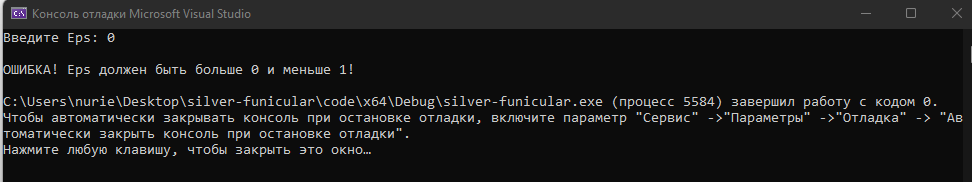
**Тест №1**

Цель теста: проверить работу программы на границе некорректной области

Исходные данные: Eps = 0

Ожидаемый результат: Сообщение об ошибке - “ОШИБКА! Eps должен быть больше 0 и меньше 1!”

Полученный результат:



Вывод: полученный результат совпал с ожидаемым, тест ошибку не обнаружил, программа работает правильно.

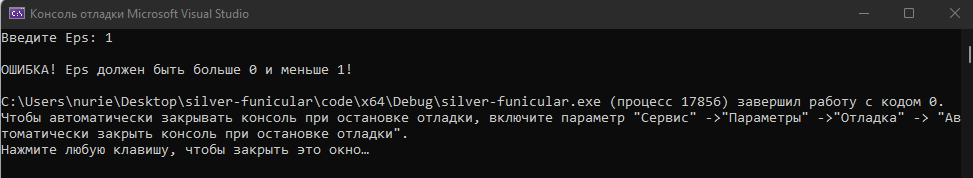
**Тест №2**

Цель теста: проверить работу программы на границе некорректной области

Исходные данные: Eps = 1

Ожидаемый результат: Сообщение об ошибке - “ОШИБКА! Eps должен быть больше 0 и меньше 1!”

Полученный результат:



Вывод: полученный результат совпал с ожидаемым, тест ошибку не обнаружил, программа работает правильно.

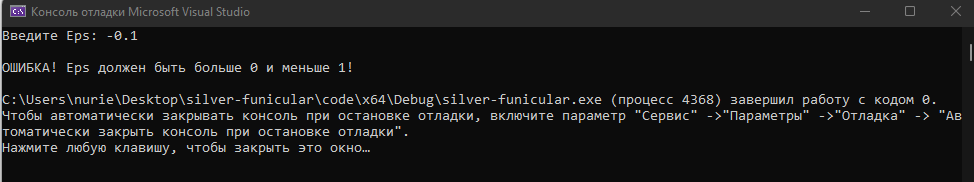
**Тест №3**

Цель теста: проверить работу программы в некорректной области

Исходные данные: Eps = -0.1

Ожидаемый результат: Сообщение об ошибке - “ОШИБКА! Eps должен быть больше 0 и меньше 1!”

Полученный результат:



Вывод: полученный результат совпал с ожидаемым, тест ошибку не обнаружил, программа работает правильно.

**Корректные тесты.**

**Тест №1**

Цель теста: проверить работу программы в корректной области

Исходные данные: Eps = 0.5

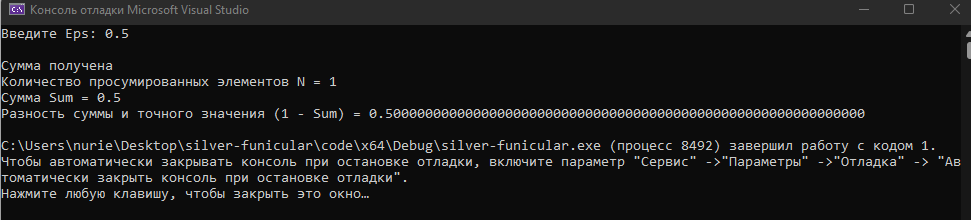
Ожидаемый результат: Сообщение - “Cумма получена”

Количество просуммированных элементов N = 1

Сумма Sum = 0.5

Разность суммы и точного значения (1 – Sum) = 0.5000000…

Полученный результат:



Вывод: полученный результат совпал с ожидаемым, тест ошибку не обнаружил, программа работает правильно.

**Тест №2**

Цель теста: проверить работу программы в корректной области

Исходные данные: Eps = 0.3

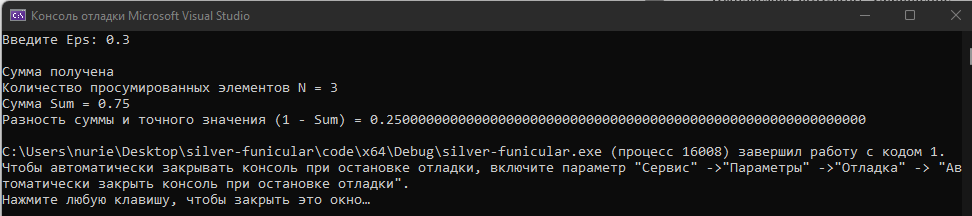
Ожидаемый результат: Сообщение - “Cумма получена”

Количество просуммированных элементов N = 3

Сумма Sum = 0.75

Разность суммы и точного значения (1 – Sum) = 0.2500000…

Полученный результат:



Вывод: полученный результат совпал с ожидаемым, тест ошибку не обнаружил, программа работает правильно.

**Тест №3**

Цель теста: проверить работу программы в корректной области

Исходные данные: eps = 0.001; x = 1.47

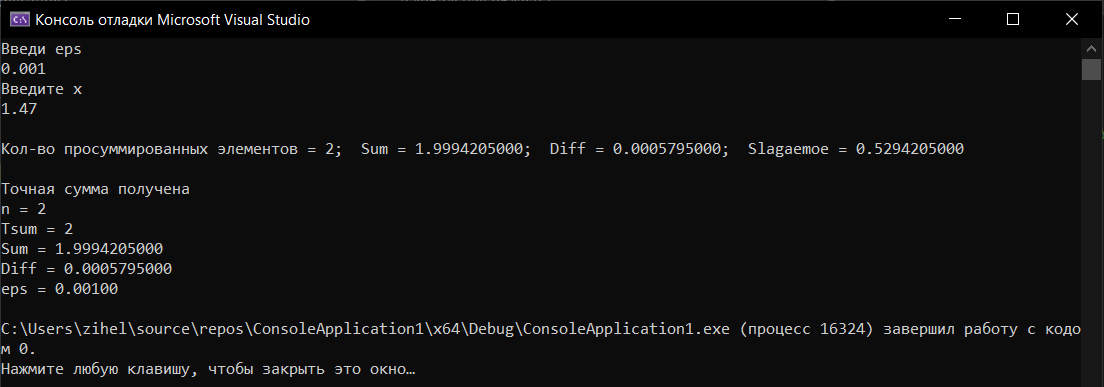
Ожидаемый результат: Сообщение - “Требуемая сумма получена”

Sum = 1.999

n = 2

Diff = 0.0005

Полученный результат:



Вывод: полученный результат совпал с ожидаемым, тест ошибку не обнаружил, программа работает правильно.

**Тест №4**

Цель теста: проверить работу программы в корректной области

Исходные данные: eps = 0.0957; x = 1.4

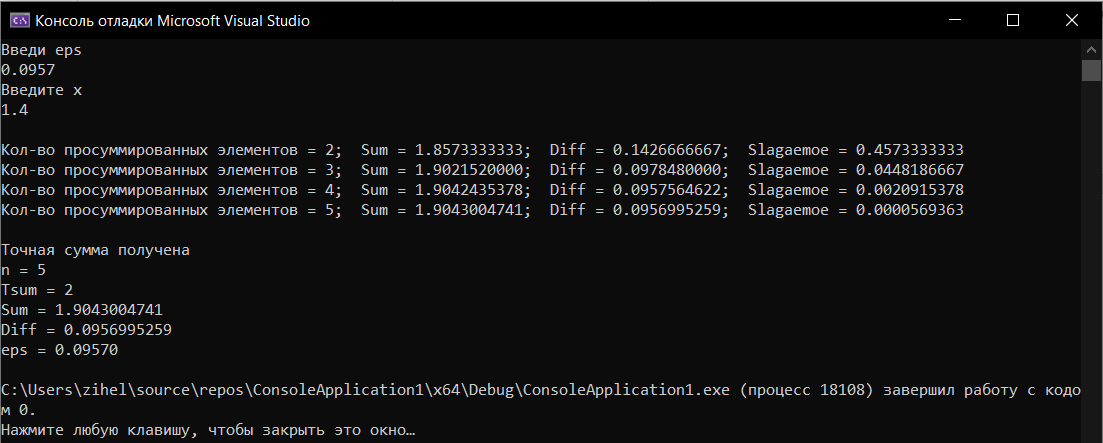
Ожидаемый результат: Сообщение - “Требуемая сумма получена”

Sum = 1.904

n = 5

Diff = 0.0956

Полученный результат:



Вывод: полученный результат совпал с ожидаемым, тест ошибку не обнаружил, программа работает правильно.

**Тест №5**

Цель теста: проверить работу программы в корректной области

Исходные данные: eps = 0.1; x = 2.2

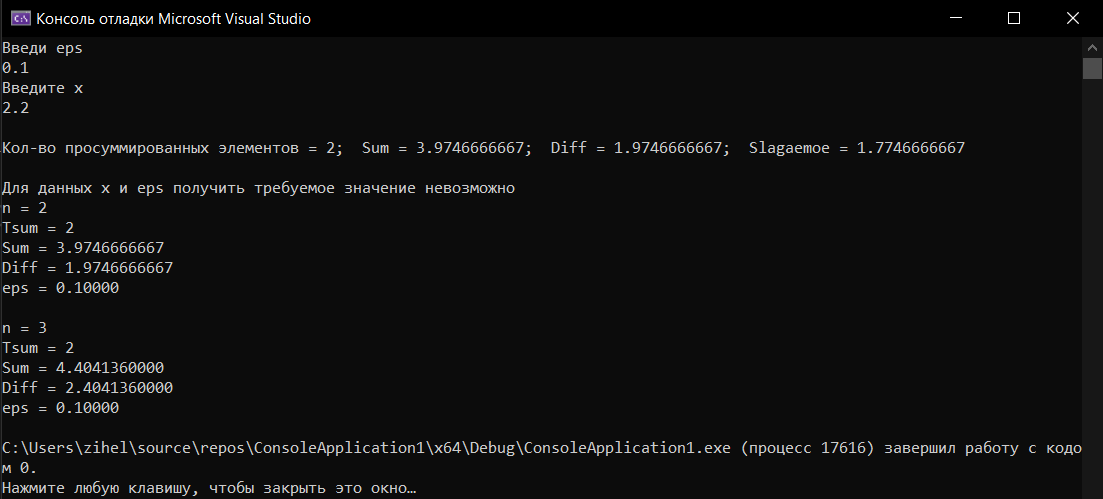
Ожидаемый результат: Сообщение - “Для данных х и eps получить требуемое значение невозможно”

Sum = 2.2

n = 1

Diff = 0.2

Полученный результат:



Вывод: полученный результат совпал с ожидаемым, тест ошибку не обнаружил, программа работает правильно.

Проведены все необходимые тесты, ошибок в программе не обнаружено.

**Вывод.**

Проанализировали алгоритм по вычислению суммы бесконечного числового ряда, в частности определение тригонометрической функции . В ходе выполнения лабораторной работы были составлены блок-схема и псевдокод для дальнейшего написания программы. Проведены некорректные тесты на проверку соответствия исходных данных и корректные, которые показали правильность выполнения алгоритма.