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 14

Выполнила бригада группы М3О-119Бк-21

Багиров Э. Р.

Нуриев Н. Н.

Проверила доцент, к.т.н., Дмитриева Е.А.

Москва 2021 г.

**Содержание**

Задание3

Схема алгоритма4

Текст программы5

Псевдокод7

Некорректные тесты8

Корректные тесты9

Вывод11

**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:

**Схема алгоритма вычисления суммы бесконечного арифметического ряда.**

1123 Ввод Eps

s

**Текст программы.**

#include <iostream>

#include <iomanip>

using namespace std;

int main()

{

setlocale(LC\_ALL, "Russian");

// Входной контроль

long double Eps;

cout << "Введите Eps: ";

cin >> Eps;

if (Eps <= 0 ) {

cout << "ОШИБКА! Eps должен быть больше 0 и меньше 1!";

return 0;

}

// Инициализация переменных

double Sum = 0;

unsigned long long N = 1;

double prevDelta = 1;

bool interrupted = false;

// Рассчет суммы

do {

Sum += 1.0 / N / (N + 1);

N++;

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

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

break; // достигнут предел точности

} //

prevDelta = 1 - Sum; //

} while ((1 - Sum) > Eps);

// Вывод результатов

cout.precision(60);

if (interrupted) {

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

}

cout << "Количество просумированных элементов N = " << N << endl;

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

cout << fixed << "Разность суммы и точного значения (1 - Sum) = " << (1 - Sum) << endl;

return 1;

}

**Псевдокод.**

ПЕР N, Sum, Eps, interrupted, prevDelta

НАЧАЛО

//ВХОДНОЙ КОНТРОЛЬ

ВВОД Eps

ЕСЛИ Eps <= 0

ТО ПЕЧАТЬ ОБ ОШИБКЕ "Eps должен быть больше 0!"

КОНЕЦ

КОНЕСЛИ

//ИНИЦИАЛИЗАЦИЯ ПЕРЕМЕННЫХ

N = 1

Sum = 0

prevDelta = 1

interrupted = false

//РАССЧЕТ СУММЫ РЯДА

ПОВТОРЯТЬ

НЦ

Sum += 1.0 / N / (N + 1) // Если точность не увеличилась

N = N + 1 //

ЕСЛИ (prevDelta + Sum - 1 == 0) // за последнюю итерацию,

ТО НЦ //

interrupted = true // достигнут предел точности

ПРЕРВАТЬ ЦИКЛ //

КЦ //

prevDelta = 1 - Sum //

КЦ // ДО ТЕХ ПОР ПОКА (1 - Sum) > Eps //

//ВЫВОД РЕЗУЛЬТАТА

ЕСЛИ interrupted = true

ТО ПЕЧАТЬ "Достигнуто ограничение по точности подсчета. Разность достигнутой точности и Eps =" 1 - Sum - Eps

ПЕЧАТЬ N, Sum, (1 - Sum)

КОНЕЦ

**Некорректные тесты.**

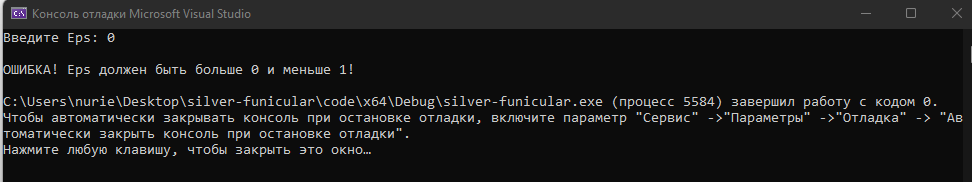
**Тест №1**

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.

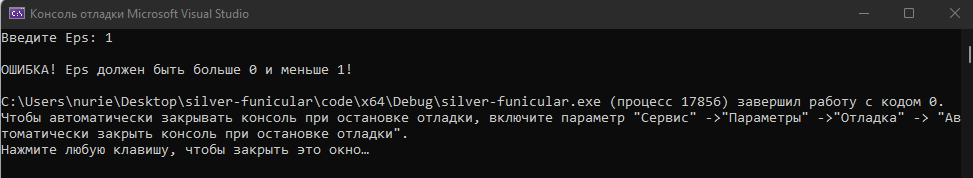
**Тест №2**

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.

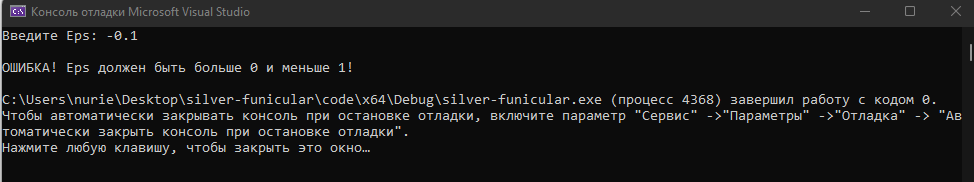
**Тест №3**

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

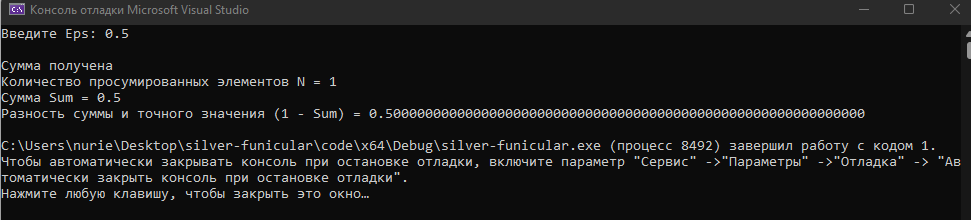
Ожидаемый результат: Сообщение - “Cумма получена”

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

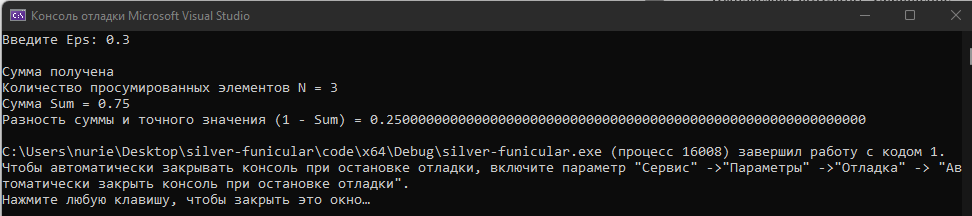
Ожидаемый результат: Сообщение - “Cумма получена”

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

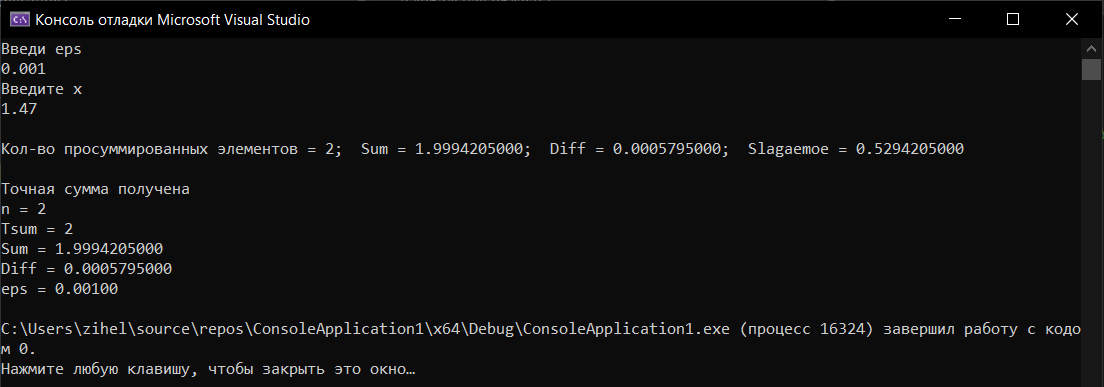
Ожидаемый результат: Сообщение - “Требуемая сумма получена”

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

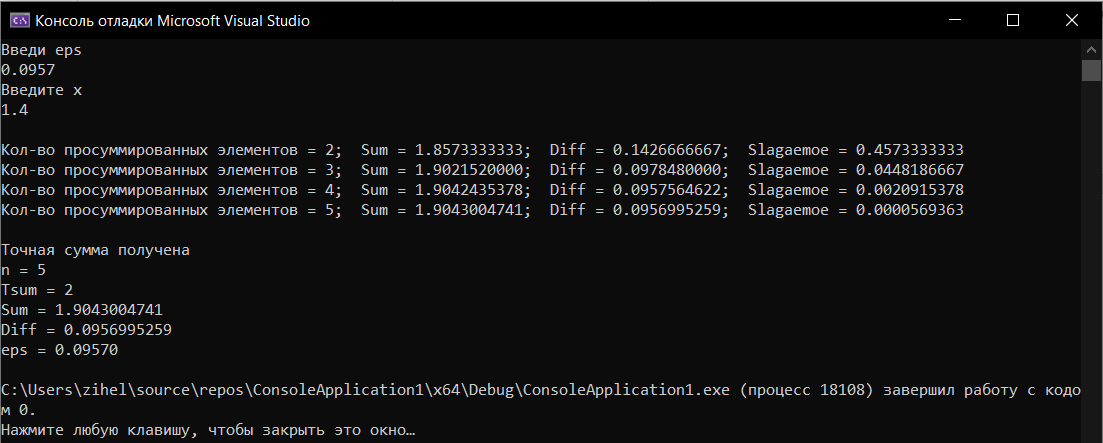
Ожидаемый результат: Сообщение - “Требуемая сумма получена”

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

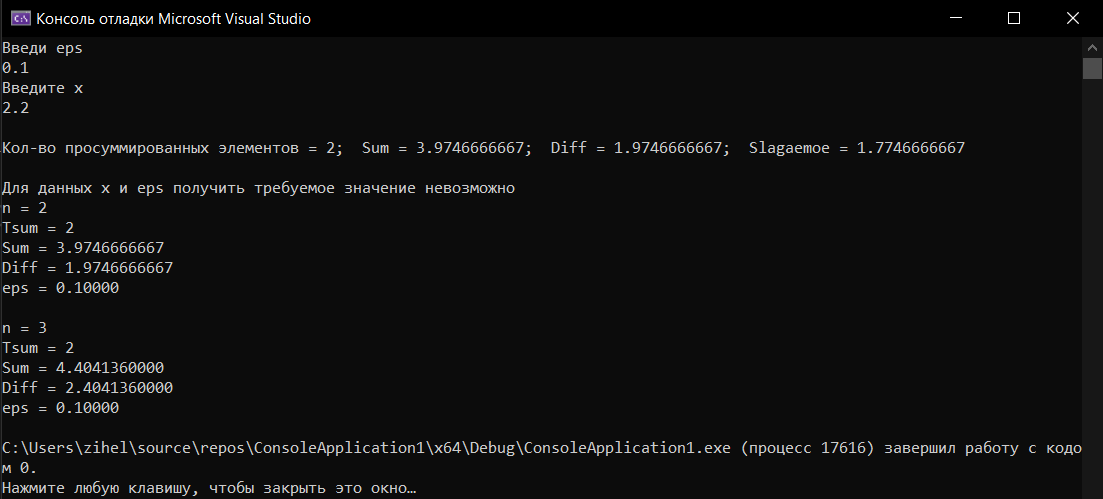
Ожидаемый результат: Сообщение - “Для данных х и eps получить требуемое значение невозможно”

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.