Государственное ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ   
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ

«Донецкий национальный технический университет»

Д.09.03.04-ПОИС.20-/5847.ЛР

***Кафедра*** искусственного интеллекта

и системного анализа

Лабораторная работа №1

по дисциплине: «Основы программирования»

Тема: «Вычисление суммы бесконечного ряда»

Проверил:

\_\_\_\_\_\_\_\_\_\_\_\_\_ асс. А. П. Семёнова

(дата, подпись)

\_\_\_\_\_\_\_\_\_\_\_\_\_ асс. И. В. Савицкая

(дата, подпись)

Выполнил:

\_\_\_\_\_\_\_\_\_\_\_\_\_ ст.гр. ПИ-20г М. А. Евсеев

(дата, подпись)

Донецк – 2020

Лабораторная работа №1

Вариант №8

## Задание

1.Написать программу на языке Си вычисления суммы ряда с точностью ɛ.

2. Вид уравнения:

# Ход работы

## Постановка Задачи

Требуется найти ту сумму ряда, в которой модуль разности двух последних элементов будет меньше или равен ɛ.

Входные данные: вещественное число .

Промежуточные данные: сленное число – i, вещественные числа: first (предыдущий элемент) и second (текущий элемент), h (уравнение)

Выходные данные: вещественное число – summ – сумма ряда с точностью ɛ.

## Алгоритм решения задачи

Найдем формулу, с помощью которой мы определим элементы суммы ряда. Для этого возьмем уравнение суммы ряда и найдем отношение текущего к предыдущему элементу:

Таким образом, отношение будет иметь следующий вид:

Используя эту формулу и текущий элемент, найдем следующий. Но для этого нам нужен самый первый элемент, который мы найдем по первоначальному уравнению, подставив начальный коэффициент:

Итак, пользователь вводит ɛ, после чего мы, зная первый элемент, находим второй, вычисляем сумму ряда и модуль разности двух последних элементов. Если модуль разности меньше или равен ɛ, то выводим на экран сумму, иначе продолжаем искать тот элемент, модуль разности с предыдущим элементом которого будет равна или меньше ɛ, после чего выводим сумму всех элементов.

Запишем алгоритм в виде блок-схемы:

Начало

eps

eps>0 и eps<1

Конец

summ

|first-second| > eps

first = second

i++

h =

second = second\* h

summ += second

i = 1

summ = 1

second = summ

## Результат контрольного просчета для первых 5 итераций

1)Методом подбора:

2) С помощью отношения текущего к предыдущему элементу:

## Листинг программы

Код программы, созданной в Visual Studio 2019:

#include <stdio.h>

#include <math.h>

//Очищаем поток входных данных

void flush\_input(void)

{

char c;

while (scanf("%c", &c) == 1 && c != '\n');

}

//вводим эпсилум, который соответствует нашим требованиям

float input\_eps(void)

{

float eps;

if (scanf("%f", &eps) == 0||eps>=1||eps<=0)

{

printf("Incorrect epsilum, try again (1>epsilum>0): \n");

flush\_input();

return input\_eps();

}

flush\_input();

return eps;

}

//отношение текущего к предыдущему элементу

float h(int i)

{

return sqrt(i) / (-i \* sqrt(i - 1));

}

void main()

{

printf("Input epsilum (1>epsilum>0):\n");

float eps = input\_eps();

int i = 1;

float summ = 1, first=0, second=1 ; //сумма, первое и последущее значение

while (fabs(first-second)>=eps)

{

first = second;

i++;

second = second \* h(i);

summ += second;

}

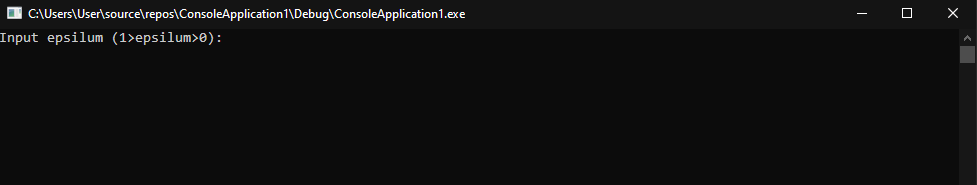
printf("summ = %f\n\n\t", summ);

getch();

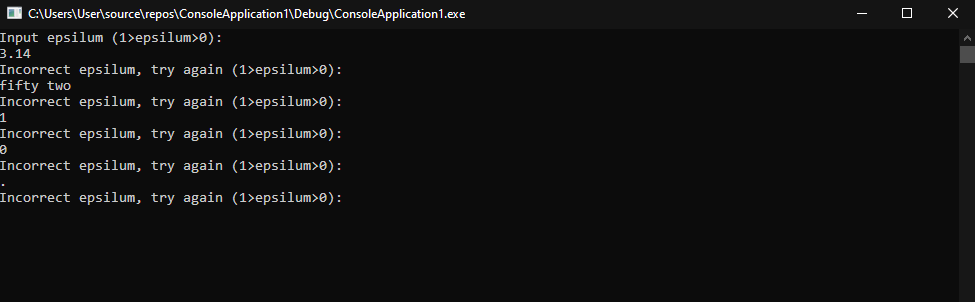
}

## Контрольный пример

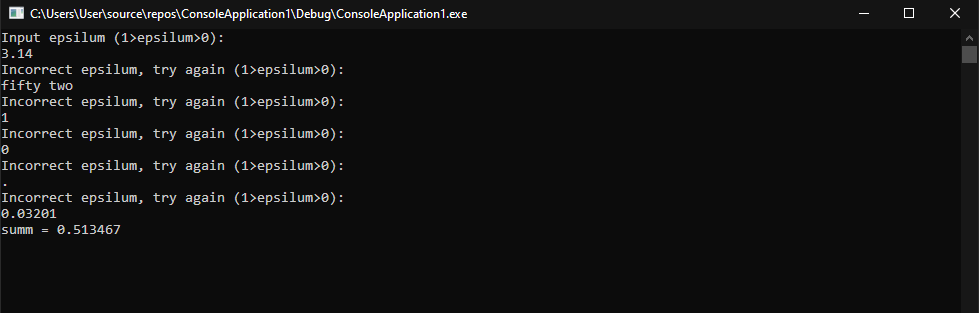
Запустив программу, мы должны ввести ɛ:



При вводе данных, несоответствующих условию , программа обработает их как неверный ввод и потребует повторного ввода до тех пор, пока мы не введем верное число:



После ввода правильного ɛ программа выдаст нам сумму ряда:



Чтобы закрыть программу достаточно нажать любую клавишу.

Вывод: Я научился писать программы на языке Си, составлять для них алгоритмы решения, тестировать их и обрабатывать исключения.