МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

### федеральное государственное автономное образовательное учреждение высшего образования

«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ»

(ГУАП)

КАФЕДРА ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ И СЕТЕЙ

Преподаватель

канд. техн. наук, доцент Л.Н. Бариков

Отчёт

по лабораторной работе №5

по дисциплине ОСНОВЫ ПРОГРАММИРОВАНИЯ

на тему: «Суммирование рядов»

Работу выполнил

студент гр. 4941 Н.С. Горбунов

Санкт-Петербург

2020

**Цель лабораторной работы:** изучение концепций и освоение технологии процедурного программирования, приобретение навыков программирования на языке C/C++ циклических вычислений при решении задач суммирования рядов.

**Задание на программирование:** используя технологию процедурного программирования разработать программу вычисления суммы ряда с заданной точностью в заданном интервале допустимых значений аргумента.

Программа **должна формировать таблицу**, содержащую значения аргумента ряда, суммы ряда, количество слагаемых и контрольные значения суммы, полученные с помощью стандартных функций библиотеки.

***Порядок выполнения работы:***

1. Получить у преподавателя индивидуальное задание. Выполнить постановку задачи: сформулировать условие, определить входные данные и их ограничения, определить вид выходной таблицы значений. При выборе типов переменных необходимо учитывать их возможные значения и не допускать превышения ограничений для соответствующего типа данных. Если переменная описана как целая беззнаковая, то превышение границы допустимого диапазона *K* приведёт к тому, что переменная будет хранить значение, взятое по модулю числа *K*+1. Если значение знаковое, то переполнение приведёт к смене его знака.

2. Разработать математическую модель:

- вывести рекуррентную формулу для расчета величины очередного слагаемого;

- описать начальные установки номера слагаемого, величины слагаемого, исходного значения суммы;

- описать процесс накопления суммы.

3. Построить схему алгоритма вычисления значения суммы ряда. Обосновать выбор циклических управляющих структур.

4. Составить программу на языке *C/C++*.

5. Входные данныевводить с клавиатуры по запросу. При этом под точностью вычисления результата понимается величина *ε*, значение которой принадлежит диапазону (0, 0.1].

6. Выходные данные выводить на экран в форме таблицы с графами:

аргумент, сумма, количество слагаемых, контрольное значение суммы.

7. Использовать стандартные потоковые объекты ввода/вывода ***cin*** и ***cout***.

8. Проверить и продемонстрировать преподавателю работу программы, при этом значение суммы должно совпадать с соответствующим контрольным значением (с заданной точностью). Выходная таблица должна содержать от 5 до 10 строк.

9. Оформить отчет о лабораторной работе в составе: постановка задачи, математическая модель, схема алгоритма вычисления значения суммы ряда, текст программы, контрольные примеры (скриншоты).

**Задание 13**

0…

**Математическая модель**

Вычисление суммы в каждой итерации цикла происходит по формуле , где - значение формулы . Каждую итерацию сравниваем абсолютную величину этого слагаемого с заданной точностью вычисления. Если  то добавляем это значение к текущему значению суммы, увеличиваем номер *n* очередного слагаемого на 1 и определяем его величину. Все перечисленные действия оформляем в виде отдельной функции, аргументами которой будут значения *x* и точности *ɛ*. Функция будет возвращать вычисленное значение суммы. Число слагаемых, попавших в сумму, будет передаваться в функцию *main*() через указатель.

По заданию необходимо получить **таблицу значений** суммы ряда в зависимости от значения аргумента *x*, вычисленных с заданной точностью *ɛ*. Поэтому в качестве исходных данных, помимо значения точности вычисления результата, вводятся начальное и конечное значения аргумента (*xn, xk*), а также шаг изменения значения аргумента от начального значения до конечного (*h*). По условию решаемой задачи допустимые значения аргумента заданы неравенством 0<*x*<1, а значение точности вычисления *ɛ* должно принадлежать интервалу (0, 0.1]. Поэтому при вводе исходных данных проверяем указанные соотношения.

После этого за текущее значение аргумента *x* принимаем значение *xn*, вычисляем для этого значения величину суммы ряда и выводим одну строку искомой таблицы (текущее значение аргумента *x*, значение вычисленной суммы, количество слагаемых *n*, попавших в сумму и контрольное значение, полученное с использованием стандартных функций). Контрольное значение в рассматриваемом примере – это величина Вычисленное значение суммы ряда для текущего значения аргумента и контрольное значение не должны различаться более чем на величину точности вычисления.

Затем увеличиваем значение аргумента *x* на величину шага изменения *h*, проверяем не превышение аргументом *x* величины *xk* и вновь вычисляем для этого значения величину суммы ряда и выводим очередную строку искомой таблицы.

Эти действия повторяются до момента, когда текущее значение *x* превысит значение *xk*. Построение таблицы закончено. Задача решена.

**Блок-схема алгоритма**

*Sum\_f*(*x*,*eps*,&*n*)

*sum*=0

*n*=0

sl=(x-1)/x

|*sl*|>*eps*&&

*n*<INT\_MAX

нет

да

*sum*+=*sl*

*n*++

sl=pow(x-1,n+1)/((n+1.)\*pow(x,n+1))

*return*(*sum*)

**Текст программы**

#include <iostream>

#include <math.h>

using namespace std;

double Sum\_f(int& n, double x, double eps);

int main()

{

int var, N;

double EPS, Xn, Xk, h,X ,rez;

setlocale(LC\_ALL, "Russian");

for (;;)

{

cout << "\n Вид действия:\n";

cout << " 1 - получение таблицы значений\n";

cout << " 2 - завершение работы программы\n";

cout << " Выберите вид действия -> ";

cin >> var;

switch (var)

{

case 1:

//ввод данных и проверка на правильность

cout << " Введите начальное значение аргумента ";

cin >> Xn;

if (fabs(Xn) >= 1 || Xn <= 0)

{

cout << " Ошибка! Значение д.б. xn>0.5";

continue;

}

cout << " Введите конечное значение аргумента ";

cin >> Xk;

cout << " Введите шаг изменения аргумента h ";

cin >> h;

if (h <= 0)

{

cout << " Ошибка! Значение h д.б. >0";

continue;

}

cout << " Введите точность вычисления eps ";

cin >> EPS;

if ((EPS <= 0) || (EPS > 0.1))

{

cout << " Ошибка! Значение eps д.б. >0 и <=0.1";

continue;

}

//Заголовок

cout << "\n Аргумент| Сумма |Кол.слаг.|Контрольное значение" << endl;

X = Xn;

do {

N = 0;

rez = Sum\_f(N,X,EPS);

cout.width(7); cout.precision(2); cout << X;

if (N < INT\_MAX)

{

cout.width(11);

cout.precision(5);

cout << rez;

cout.width(6);

cout << N;

cout.width(15);

cout.precision(4);

cout << log(X) << endl;

}

else cout << " Точность не достигнута\n";

X += h;

} while (X <= Xk);

break;

default: return 0;

}

}

}

double Sum\_f(int &n, double x, double eps) //функция вычисляет значение по формуле

{

double sum = 0;

double sl = (x - 1) / x;

while (fabs(sl) > eps && n < INT16\_MAX)

{

sum = sum + sl;

n++;

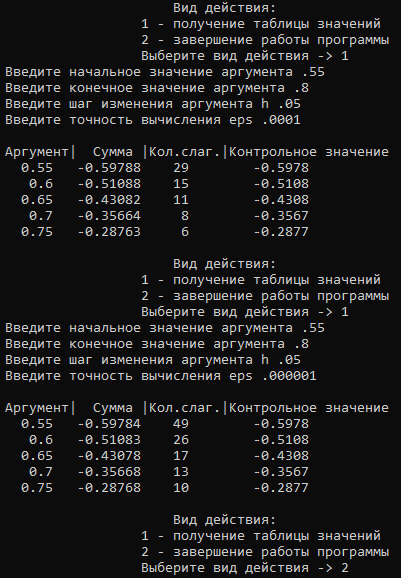
sl = sl\*(x-1)\*n/((n+1)\*x);

}

return sum;

}

**Пример работы программы**

****

**Вывод:** используя технологию процедурного программирования разработал программу вычисления суммы ряда с заданной точностью в заданном интервале допустимых значений аргумента.