(Ленина)

Кафедр САПР

епап

отчет

**по лабораторной работе №5**

**по дисциплине «Программирование»**

Тема: Функции и файлы языка СИ++.

Вариант: 18

|  |  |  |
| --- | --- | --- |
| Студент гр. 4351 |  | Чмырева В.Д. |
| Преподаватель |  | Рыжов Н.Г. |

Санкт-Петербург

2024

**Цель работы.**

Освоение принципов работы с битовыми операциями, функциями и файлами. Разделение программы на файлы. Знакомство с понятиями класса и объекта класса.

**Задание.**

1. Написать функцию, меняющую в переменной типа unsigned char местами биты с номерами n и n+1, где n = номер варианта % 7. Биты нумеруются справа налево. Младший справа бит имеет номер 0. Выполнить задание с использованием битовых операций. Организовать тестовый вызов функции, после чего вывести на экран результат ее работы, используя символьное представление переменной типа unsigned char.

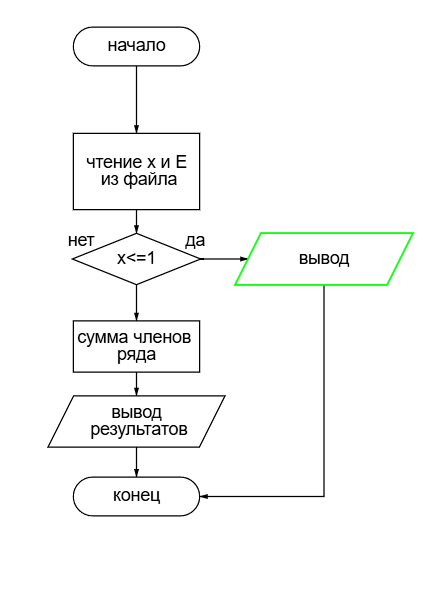
2. Написать функцию sumOfSeries, возвращающую сумму ряда yn=1/x+1/x2+…+1/xn, (x>1) с точностью до ε = 0.001. В качестве параметров по значению передать x и ε, в качестве параметра по ссылке обеспечить возврат количества итераций n. Точность ε вычислять по значению последнего члена ряда ε n = 1/xn. Вычисления заканчиваются при достижении заданной точности ε. Организовать ввод параметров x и ε из текстового файла.

Определение функции sumOfSeries вынести в отдельный файл с расширением \*.cpp. Прототип функции sumOfSeries определить в заголовочном файле и подключить директивой #include "имя\_файла".

Организовать тестовый вызов функции, после чего вывести на экран исходные данные x и ε, а также рассчитанные значения yn, n и εn.

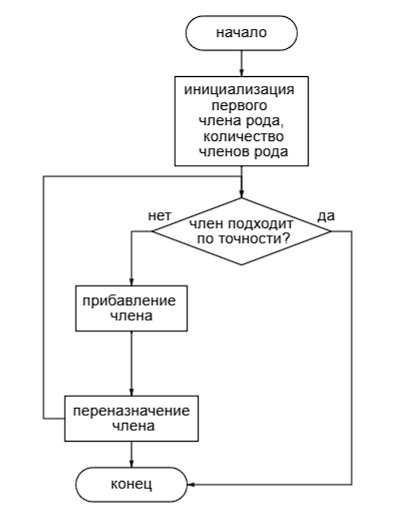
**Схема алгоритма.**

**Сумма ряда.**



нет

**Сумма членов ряда.**



**Код программы.**

***test.cpp:***

#include <iostream>

using namespace std;

*// Функция для перестановки двух соседних битов n и n+1*

unsigned char swapBits(unsigned char value, int n) {

*// Маски для битов n и n+1*

    unsigned char bit1 = (value >> n) & 1;

    unsigned char bit2 = (value >> (n + 1)) & 1;

*// Если биты разные, то нужно их поменять*

    if (bit1 != bit2) {

*// Переключаем биты n и n+1 с помощью исключающего или*

        value ^= (1 << n) | (1 << (n + 1));

    }

    return value;

}

int main() {

    int variant\_number = 18; *// Номер варианта*

    int n = variant\_number % 7;

    unsigned char value = 'A'; *// Тестовая переменная, например, символ 'A'*

    cout << "Original: " << value << " (" << int(value) << ")\n";

    unsigned char result = swapBits(value, n);

    cout << "After swap: " << result << " (" << int(result) << ")\n";

    return 0;

}

***Input.txt:***

***2.5***

***0.001***

***sumOfSeries.h:***

#ifndef SUMOFSERIES\_H

#define SUMOFSERIES\_H

double sumOfSeries(double x, double E, int& n);

#endif

***sumOfSeries.cpp:***

#include "sumOfSeries.h"

#include <cmath>

double sumOfSeries(double x, double E, int& n) {

    double sum = 0.0;

    double term = 1.0 / x;

    n = 1;

*// Добавляем члены ряда, пока последний член больше заданной точности*

    while (term > E) {

        sum += term;

        n++;

        term = 1.0 / pow(x, n);

    }

    return sum;

}

***Main.cpp:***

#include <iostream>

#include <fstream>

#include "sumOfSeries.h"

#include <cmath>

#include <windows.h>

using namespace std;

int main() {

    SetConsoleCP(CP\_UTF8);

    SetConsoleOutputCP(CP\_UTF8);

    double x, E;

    int n = 0; *// Колличество членов*

*// Чтение x и E из файла*

    ifstream inputFile("input.txt");

    if (!inputFile) {

        cerr << "Ошибка. Не удалось считать файл\n";

        return 1;

    }

    inputFile >> x >> E;

    inputFile.close();

    if (x <= 1) {

        cerr << "Ошибка. х не может быть меньше или равен 1\n";

        return 1;

    }

*// Вызов функции для вычисления суммы ряда*

    double result = sumOfSeries(x, E, n);

*// Вывод результатов*

    cout << "x: " << x << ", E: " << E << "\n";

    cout << "Сумма: " << result << "\n";

    cout << "Колличество членов (n): " << n << "\n";

    cout << "Последний член (E\_n): " << 1.0 / pow(x, n) << "\n";

    return 0;

}

**Описание программы.**

Этот проект состоит из двух основных частей:

1. Перестановка битов: Программа меняет местами два соседних бита числа. Биты n и n+1 меняются, если они имеют разные значения. Это реализуется функцией swapBits().

2. Вычисление суммы ряда: Вторая часть программы вычисляет сумму членов ряда с точностью до заданного E. Программа считывает значения x и E из файла input.txt, а затем вычисляет сумму ряда с помощью функции sumOfSeries().

**Руководство пользователя.**

Убедитесь, что рядом с исполняемым файлом находится файл input.txt с двумя числами:

• Первое число: x (должно быть больше 1).

• Второе число: E (определяет точность вычислений).

Запустите программу:

• Первая часть выведет символ value и его значение после перестановки битов.

• Вторая часть прочитает x и E из файла и вычислит сумму ряда, отображая результаты на экране:

• Значения x и E.

• Вычисленная сумма.

• Количество членов в ряду n.

• Последний член, который не превосходит E.

**Пути дальнейшего улучшения.**

1. Добавить обработку ошибок для входных данных: Проверить допустимость значения E (должно быть больше 0) и предусмотреть случаи, когда x может быть близко к 1, чтобы избежать вычислительных ошибок.

2. Оптимизация вычисления рядов: Реализовать кеширование для значений 1/pow(x, n), чтобы избежать повторного вычисления степени при каждой итерации, что ускорит программу.

3. Расширить функционал перестановки битов: Добавить возможность менять произвольные пары битов не только соседние, что может быть полезно для работы с побитовой обработкой данных.

**Вывод.**

В данной лабораторной работе была разработана программа, иллюстрирующая принципы выполнения побитовых операций и вычисления суммы рядов. Программа продемонстрировала, как управлять битами, изменяя их порядок с помощью логических операций. Также была реализована функция, позволяющая вычислять ряд с заданной точностью, что даёт возможность контролировать расчёты.