МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«ПЕРМСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ

ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

(ПНИПУ)

Факультет: Электротехнический

Кафедра: «Информационные технологии и автоматизированные системы» (ИТАС)

Направление: Разработка информационных систем (РИС)

**О Т Ч Ё Т**   
**о лабораторной работе №2**

Дисциплина: «Основы алгоритмизации и программирования»

Тема: «Рекурсии»

Выполнил:

Студент группы РИС-24-3б

Суяргулов А.О.

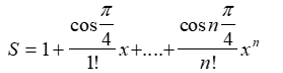
Проверил:

Доцент кафедры ИТАС

Полякова О.А.

Пермь, 2024

ВВЕДЕНИЕ

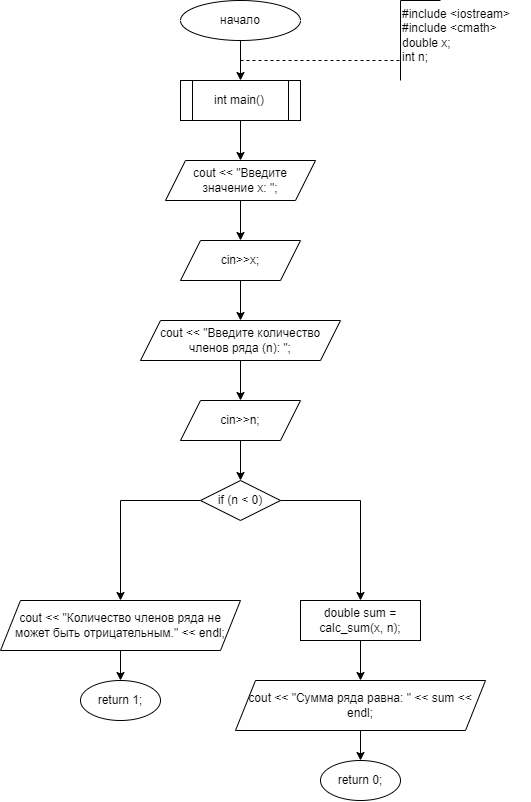
Дано уравнение суммы 

Необходимо найти S с помощью рекурсий на C++.

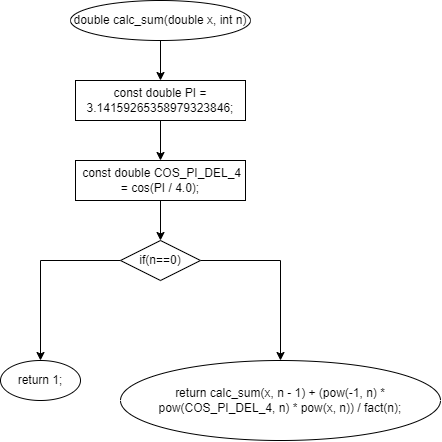
Найдем S при x = 10, n = 5.

ОСНОВНАЯ ЧАСТЬ

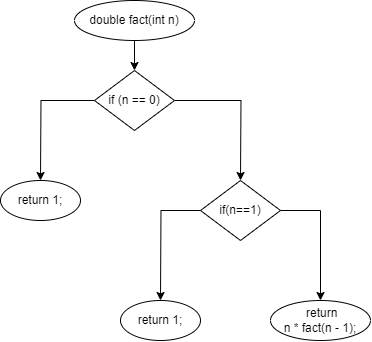
**Блок-схема:**



**Функция, считающая сумму:**



**Функция, считающая факториал:**



**Код:**

#include <iostream>

#include <cmath>

using namespace std; // Используем пространство имен std

// Рекурсивная функция вычисления факториала

double fact(int n) {

if (n == 0) return 1; // факториал 0 равен 1

if (n == 1) return 1; // факториал 1 равен 1

return n \* fact(n - 1); // n! = n \* (n-1)!

}

// Рекурсивная функция вычисления суммы ряда

double calc\_sum(double x, int n) {

const double PI = 3.14159265358979323846; // Объявляем и инициализируем переменную Pi значением числа Pi

const double COS\_PI\_DEL\_4 = cos(PI / 4.0); // Вычисляем cos(π/4)

if (n == 0) {

return 1; // Первый член ряда равен 1

}

else {

return calc\_sum(x, n - 1) + (pow(-1, n) \* pow(COS\_PI\_DEL\_4, n) \* pow(x, n)) / fact(n); // Возвращаем исходную функцию суммы

}

}

int main() {

double x; // Объявляем переменную для значения x

int n; // Объявляем переменную для количества членов ряда

cout << "Введите значение x: "; // Выводим предложение ввода х

cin >> x; // Ввод х

cout << "Введите количество членов ряда (n): "; // Выводим предложение ввода n

cin >> n; // Ввод n

if (n < 0) { // Проверка на отрицательное количество членов ряда

cout << "Количество членов ряда не может быть отрицательным." << endl; // Cообщение об ошибке

return 1;

}

double sum = calc\_sum(x, n); // Вычисляем сумму ряда

cout << "Сумма ряда равна: " << sum << endl; // Выводим результат

return 0;

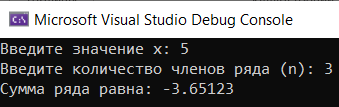
}

**Результат работы кода:**









**Ссылка на GitHub:**

<https://github.com/Drazda1488/Lab_2_Rec>