Министерство науки и высшего образования Российской Федерации Федеральное государственное бюджетное образовательное учреждение высшего образования

«Пермский национальный исследовательский политехнический университет»

Электротехнический факультет

Кафедра «Информационные технологии и автоматизированные системы»

**ЛАБОРАТОРНАЯ РАБОТА**

**«Простая рекурсия»**

Выполнил:

студент группы ИВТ-23-2б

Чудинов Данил Николаевич

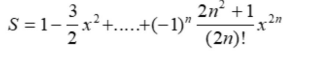
Проверила:

доцент кафедры ИТАС

О. А. Полякова

2024 г.

**Постановка задачи.**

****

**Задачи:**Постановка задачи: приближенно вычислить значение функции двумя способами:

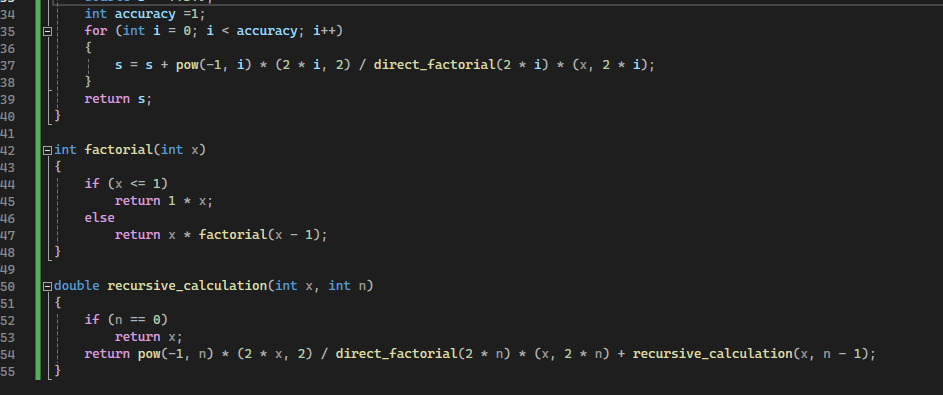
1) Через сумму функционального ряда с помощью рекурсивной функции. Аргументы функции – n и x, где n – количество членов ряда, x – переменная.

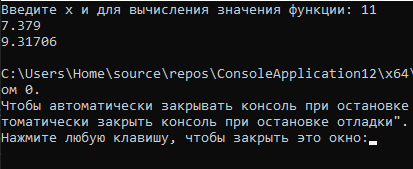
2) Через прямое вычисление значения функции. Аргумент функции - переменная x.

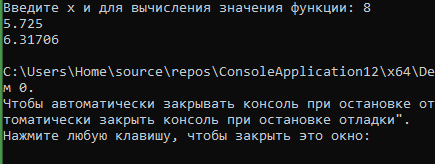
Сравнить полученные результаты.

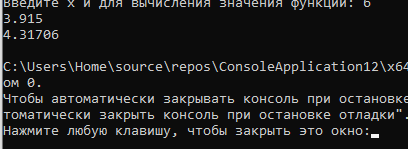
**Код:**

****









**Этот код реализует расчет значения функции с использованием двух различных схем: прямого вычисления и рекурсивного вычисления.**

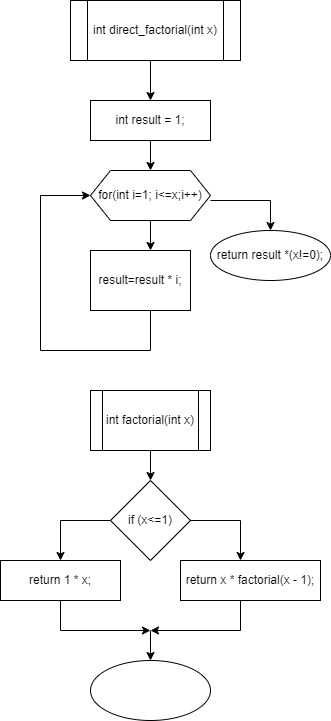
**Функция direct\_factorial - это рекурсивная функция, которая рассчитывает факториал числа n.**

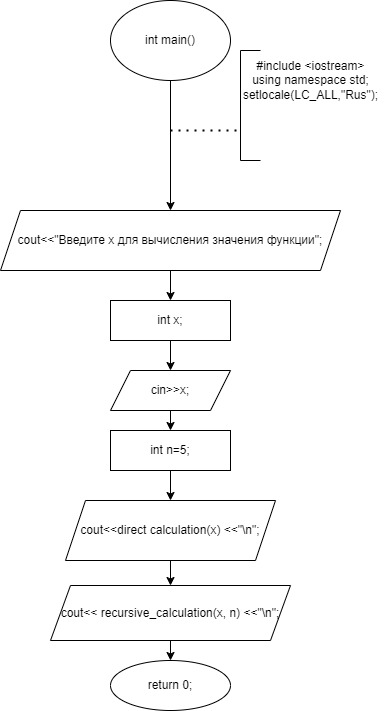
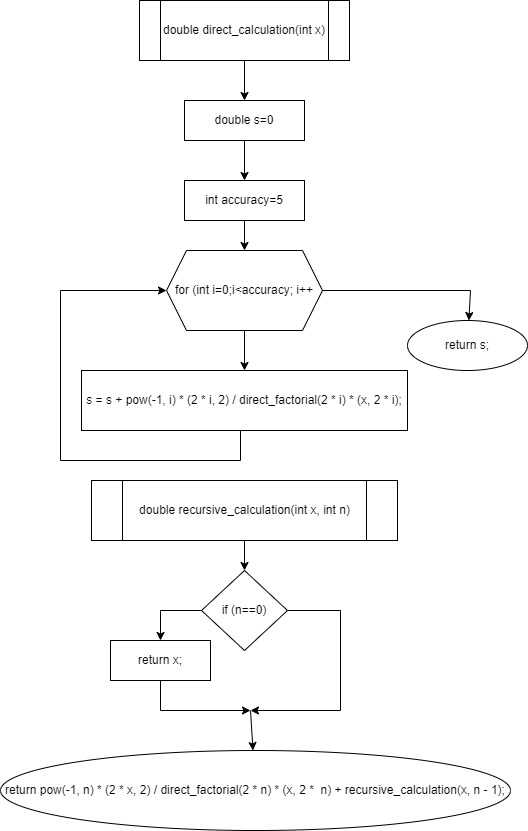
**Функция direct\_calculation - использует цикл для прямого вычисления значения функции. Внутри цикла вычисляется каждый член с помощью формулы pow(-1, i) \* (2 \* i, 2) / direct\_factorial(2 \* i) \* (x, 2 \* i) и суммируется с общей суммой s.**

**Функция factorial - рекурсивно вычисляет факториал числа x.**

**Функция recursive\_calculation - рекурсивно вычисляет значение функции. Она использует формулу pow(-1, n) \* (2 \* x, 2) / direct\_factorial(2 \* n) \* (x, 2 \* n) для вычисления каждого члена и рекурсивно вызывает саму себя с уменьшением значения n. Таким образом, суммируются все члены функции с помощью рекурсивных вызовов.**

**Блок Схемы:**



****

**Числа Фибоначчи**

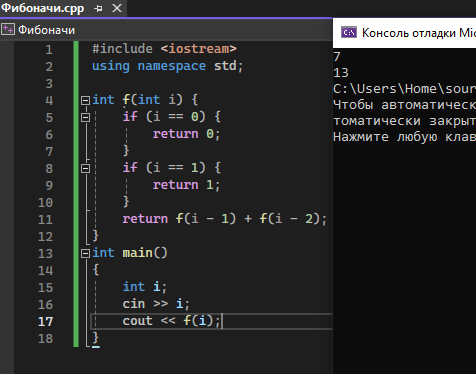
**Задача:**

Вычислить значение которое будет под определенным рядом Фибоначчи.

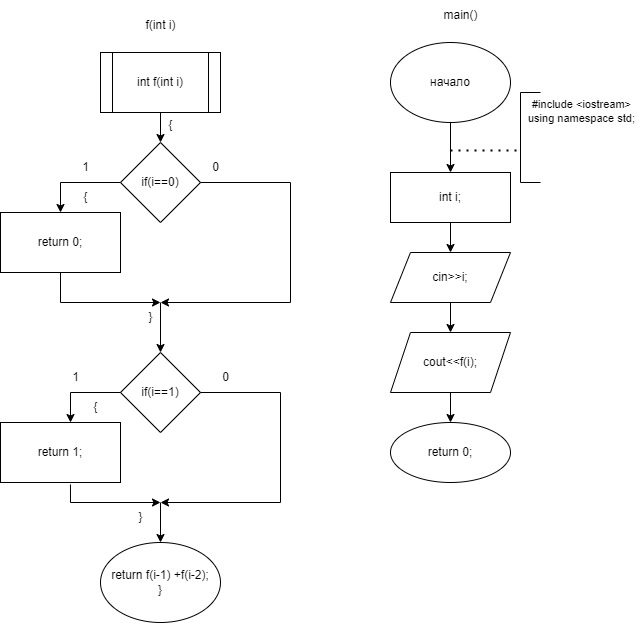
**Анализ задачи:**

1. Последовательность целых чисел
2. Первое число =0, второе число =1
3. Остальные числа являются суммами двух предыдущих

**Код программы:**

****

**Блок-схема:**

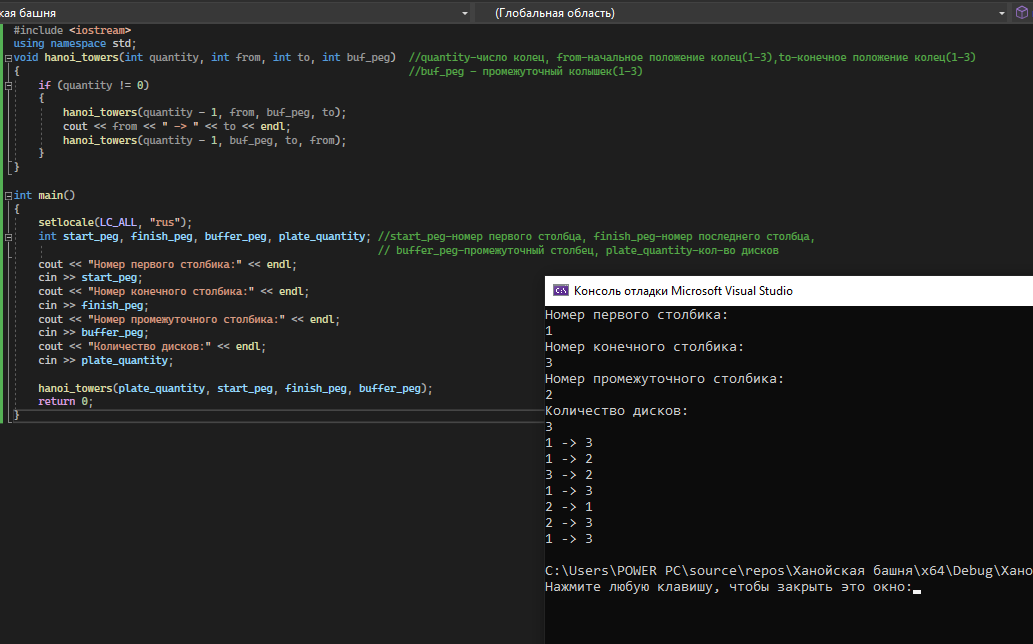
****

**Ханойская башня**

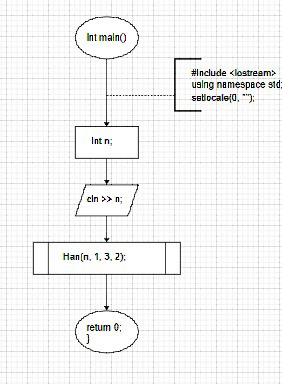
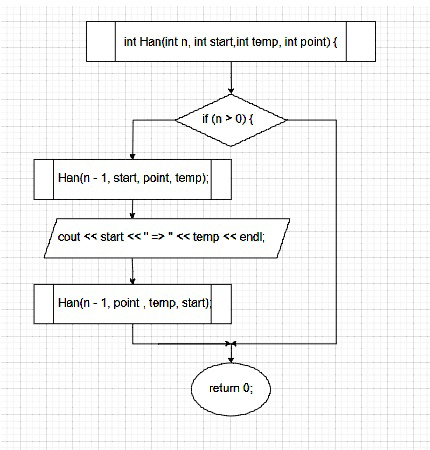
**Задача:** имеется 3 стержня, на первом стержне расположены диске в порядке убывания диаметра, остальные два стержня пустые. Необходимо перенести диски с первого стержня на третий, выполняя условие, что больший диск не может ставиться на меньший.

**Анализ задачи:**

1. 3 стержня, 2 пустых
2. Диски расположены по убыванию диаметра начиная сверху
3. Перенести все диски с 1го стержня на 3й используя все 3 стержня
4. Один шаг-перенос одного диска

**Код программы:**

**Блок-схема:**

****

hanoi\_towers (3, 1, 3, 2)

{

Большее кольцо

Находится под средним и меньшей

If (quantity ! = 0);

{

hanoi\_towers (2, 1, 2, 3);

cout<<from<<”🡪”<<to<<endl; \\ 1🡪3

hanoi\_tower (2, 2, 3, 1);

На 2 стержне ещё нет колец

}

hanoi\_towers (2, 1, 2, 3);

{

If (quantity ! = 0);

{

hanoi\_towers (1, 1, 3, 2);

cout<<from<<”🡪”<<to<<endl; \\ 1🡪2

hanoi\_tower (1, 3, 2, 1);

Среднее кольцо находится пол меньшим

}

hanoi\_towers (2, 2, 3, 1);

{

If (quantity ! = 0);

{

hanoi\_towers (1, 2, 1, 3);

cout<<from<<”🡪”<<to<<endl; \\ 2🡪3

hanoi\_tower (1, 1, 3, 2);

}

hanoi\_towers (1, 1, 3, 2);

{

If (quantity ! = 0);

{

hanoi\_towers (0, 1, 2, 3);

cout<<from<<”🡪”<<to<<endl; \\ 1🡪3

hanoi\_tower (0, 2, 3, 1);

}

}

hanoi\_towers (1, 3, 2, 1);

На 3 стержне находится меньшее кольцо

{

If (quantity ! = 0);

{

hanoi\_towers (0, 3, 1, 2);

cout<<from<<”🡪”<<to<<endl; \\ 1🡪3

hanoi\_tower (0, 1, 2, 3);

}

}

На 1 стержне находится меньшее кольцо, поэтому мы не можем туда ничего перекинуть

hanoi\_towers (1, 2, 1, 3);

{

If (quantity ! = 0);

{

hanoi\_towers (0, 2, 3, 1);

cout<<from<<”🡪”<<to<<endl; \\ 2🡪1

hanoi\_tower (0, 3, 1, 2);

}

Перекидываем среднее кольцо

}

hanoi\_towers (1, 1, 3, 2);

{

If (quantity ! = 0);

{

hanoi\_towers (0, 1, 2, 3);

cout<<from<<”🡪”<<to<<endl; \\ 1🡪3

hanoi\_tower (0, 3, 1, 2);

}