Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное учреждение высшего образования

«Пермский национальный исследовательский политехнический университет»

ПНИПУ

**Лабораторная работа “Рекурсия”**

**Выполнил: Студент группы РИС-23-3б**

**Асташин Дмитрий Алексеевич**

**Проверила: Доцент кафедры ИТАС**

**О. А. Полякова**

2023

**Задача 1**

1. 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.

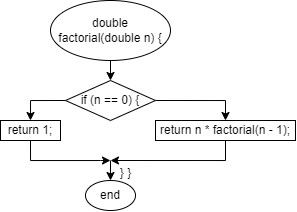
2. Через прямое вычисление значения функции. Аргумент функции - переменная x.

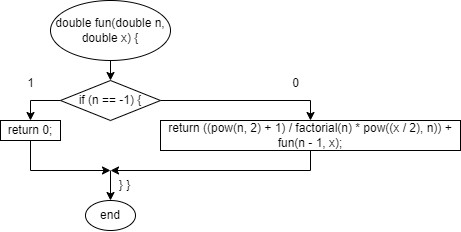
3. Сравнить полученные результаты

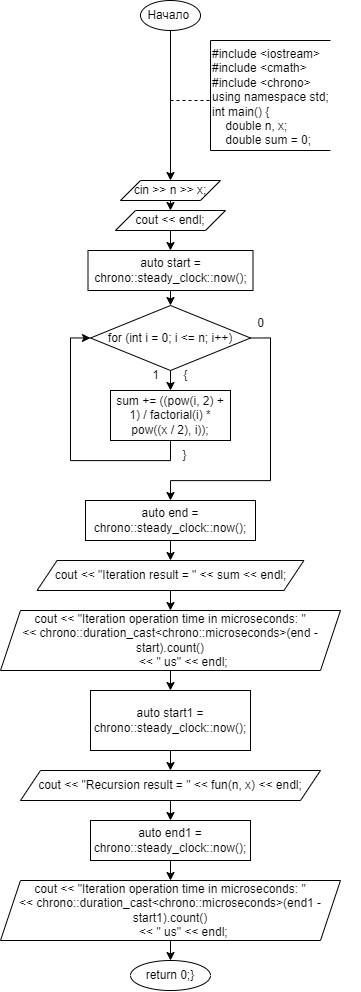


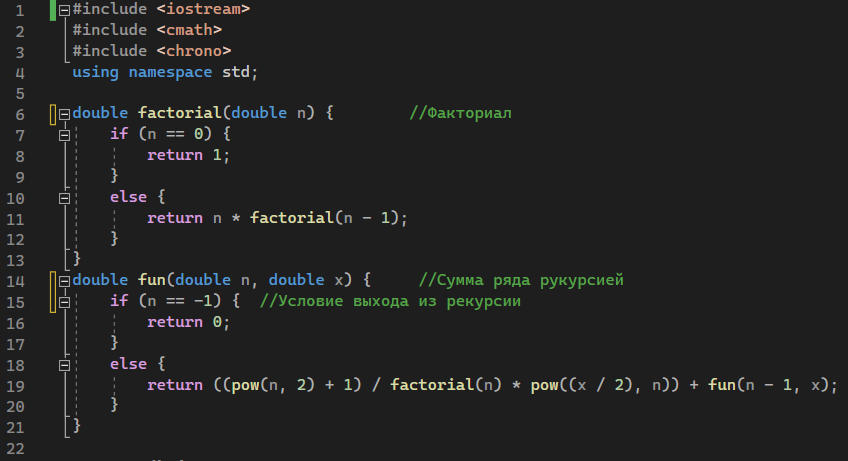
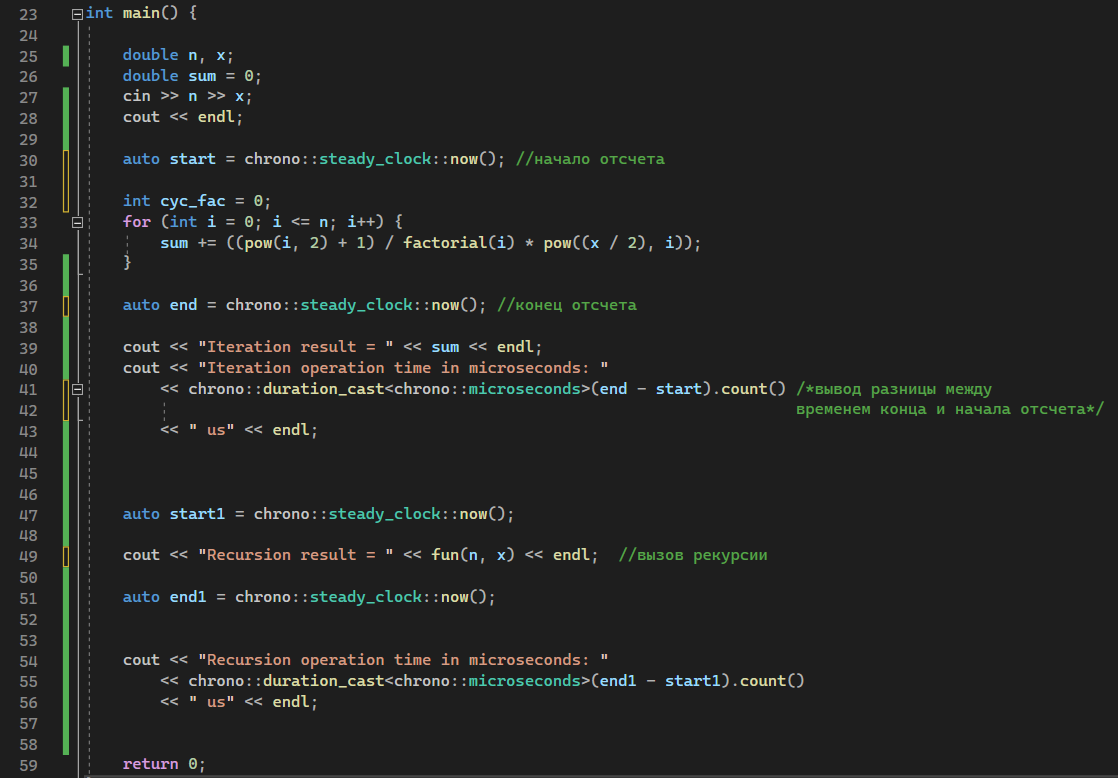
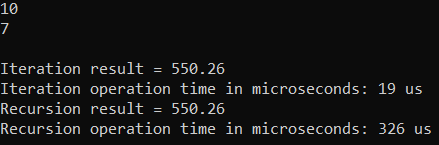
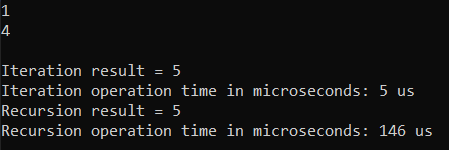
1. Анализ:
2. Для нахождения факториала создадим рекурсивную функцию:

Если входной n параметр равен нулю, то функция возвращает единицу, иначе функция возвращает произведение параметра n на вызов функции с параметром меньшим на 1 (n – 1).

1. Рекурсивная функция нахождения суммы ряда возвращает сумму результата слагаемого от параметра n и результата вызова данной функции от (n – 1). Условие выхода из рекурсии – (n = -1).
2. Итерационный цикл на каждом шаге находит одно слагаемого ряда и складывает его с переменной суммы всего ряда.
3. Для сравнения времени работы используем библиотеку <chrono>.
4. 





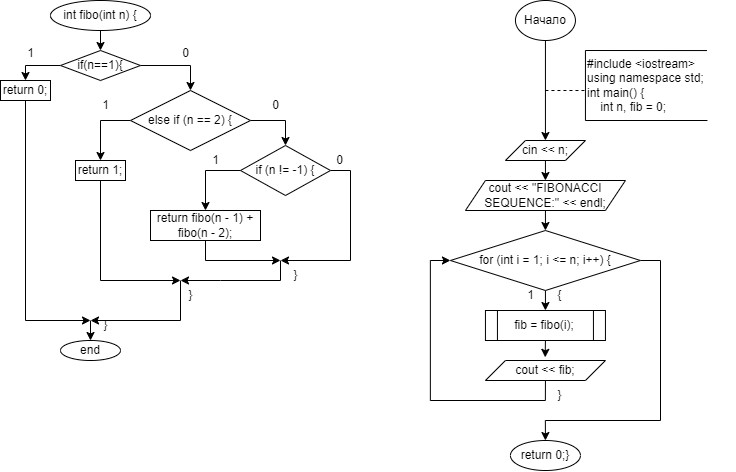
1. Код программы
2. Результаты
3. Исходя из результатов, можно сделать вывод, что итерационный цикл работает быстрее рекурсии.

**Задача 2**

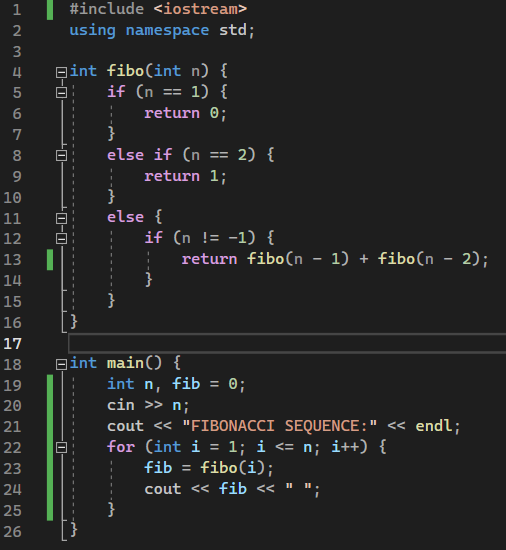
1. Постановка задачи: вывести числа Фибоначчи в количестве, равном введенному пользователем.
2. Анализ: рекурсивная функция будет принимать параметр n (номер элемента последовательности Фибоначчи) и возвращать сумму вызовов самой себя от (n – 1) и

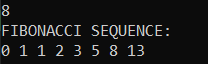
(n – 2). Условие для остановки рекурсии (n!= -1). Итерационный цикл в функции main() выводит всю последовательность чисел.

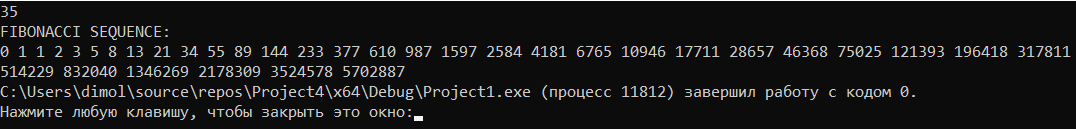
1. Блок-схема:



1. Код программы:



1. Результаты:



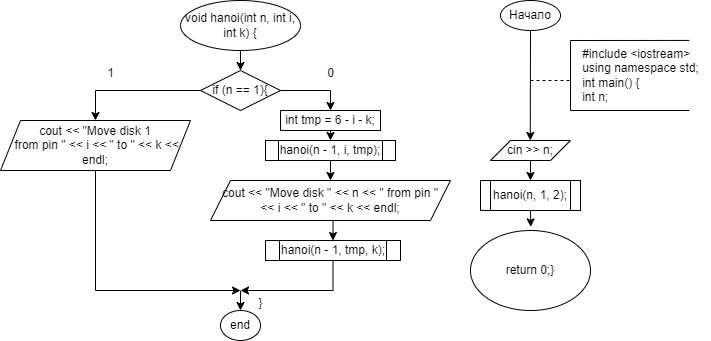
**Задача 3**

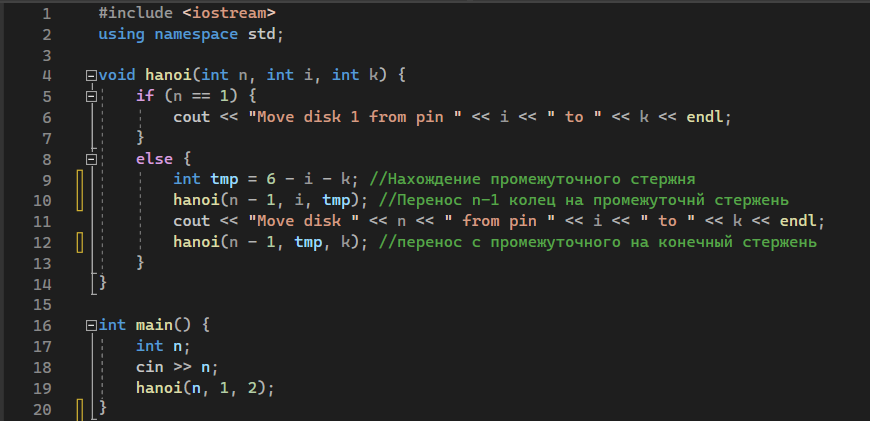
1. Постановка задачи:

Даны три стержня, на один из которых нанизаны n колец, причём кольца отличаются размером и лежат меньшее на большем. Задача состоит в том, чтобы перенести пирамиду из восьми колец за наименьшее число ходов на другой стержень. За один раз разрешается переносить только одно кольцо, причём нельзя класть большее кольцо на меньшее.

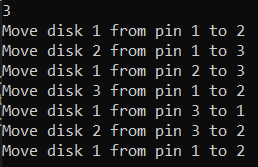
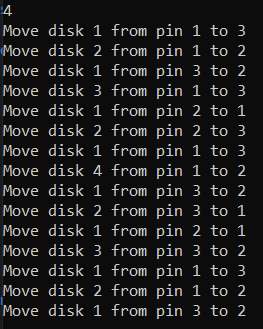
1. Анализ:
2. Рекурсия принимает 3 параметра:

* n - количество перекладываемых дисков;
* i – первый стержень, с которого снимаем кольцо;
* k – второй стержень, на который перекладываем кольцо.

1. Если количество колец (n = 1), то сразу перекладываем c i на k;
2. Иначе находим свободный стержень: tmp = 6 - i - k;
3. Перекладываем n – 1 диск на свободный стержень, вызвав функцию;
4. Перекладываем n – 1 диск на конечный стержень;
5. Блок-схема:
6. Код программы



1. Результаты:



https://github.com/DimetriusAsti/RECURSION/tree/main