

# Реквизиты

Жолобов Даниил Валерьевич

3 курс

z33434

2024

## ЛР #3: [C++ & UNIX]: C++ CLI / FUNCTION / LOOP / RECURSION

### Цель

Познакомить студента с основными алгоритмическими конструкциями, которые будут использоваться для создания CLI программы. Далее продемонстрировать эффективность использования механизма рекурсии.

C++ алгоритмы: CLI Калькулятор вещественных чисел  $+$ ,  $-$ ,  $^$ ,  $.$ . Реализация с использованием только функций, условий, циклов,  $+$  и  $-$ . Вид команд в консоли: `calc plus / minus / power;`  
Ханойская башня, результат корректной последовательности

### Задача

1. [C++ CLI CALC] Создать программу CALC с интерфейсом CLI

Создать программу под названием CALC, которая будет принимать на вход 3 аргумента (2 операнда и 1 оператор). Оператор может быть:  $+$ ,  $-$ ,  $^$ . Реализация операторов только с использованием функций, условий, циклов,  $+$ ,  $-$  и  $*$ .

2. [C++ RECURSION] Решить задачу ханойской башни с использованием рекурсии

Описание: Ханойская башня является одной из популярных головоломок XIX века. Даны три стержня, на один из которых нанизаны восемь 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.

Результат обнаруженной последовательности шагов записать в виде двусвязного списка. В конце программы сделать вывод этого списка на экран. Освободить память списка перед завершением программы.

3. [SAVE] Результат всех вышеперечисленных шагов сохранить в репозиторий (+ отчет по данной ЛР в папку doc)

Фиксацию ревизий производить строго через ветку dev. С помощью скриптов накатить ревизии на stg и на prd. Скрипты разместить в корне репозитория. Также создать скрипты по возврату к виду текущей ревизии (даже если в папке имеются несохраненные изменения + новые файлы).

## Решение

1. [C++ CLI CALC] Создать программу CALC с интерфейсом CLI

Создать программу под названием CALC, которая будет принимать на вход 3 аргумента (2 операнда и 1 оператор). Оператор может быть: +, -, ^. Реализация операторов только с использованием функций, условий, циклов, +, - и \*.

Для простоты реализуем в начале быстрое возведение в степень для целых степеней.

Затем можно реализовать возведение в дробную степень, используя разложения в ряды Тейлора для натурального логарифма и возведения экспоненты в степень, добавив к каждому из них редукцию аргумента из-за малых промежутков сходимости рядов.

$$a^b = e^{b \ln a} = e^x$$

$$x = z \ln 2 + r,$$

$$z \in \mathbb{Z}$$

$$r \in \mathbb{R}$$

$$0 \leq r < 1$$

$\implies$

$$e^x = e^{z \ln 2 + r} = 2^z e^r$$

Тогда

$$z = \left\lfloor \frac{x}{\ln 2} \right\rfloor$$

$$r = x - z \ln 2$$

## 2. [C++ RECURSION] Решить задачу ханойской башни с использованием рекурсии

Описание: Ханойская башня является одной из популярных головоломок XIX века. Даны три стержня, на один из которых нанизаны восемь 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.

Результат обнаруженной последовательности шагов записать в виде двусвязного списка. В конце программы сделать

вывод этого списка на экран. Освободить память списка перед завершением программы.

Для начала воспользуемся `std::deque` и рекурсией. Пусть известно, как перенести  $k$  дисков башни со стержня `from` на стержень `to` (решена задача для предыдущего размера башни). Тогда для выполнения переноса  $k+1$  дисков достаточно перенести  $k$  из `from` на "третий" стержень, перенести 1 из `from` на `to` и снова перенести  $k$  дисков с "третьего" на `to`.

Затем результат можно проверить на собственной реализации `my_deq` с урезанным функционалом.

3. [SAVE] Результат всех вышеперечисленных шагов сохранить в репозиторий (+ отчет по данной ЛР в папку `doc`)

Фиксацию ревизий производить строго через ветку `dev`. С помощью скриптов накатить ревизии на `stg` и на `prd`. Скрипты разместить в корне репозитория. Также создать скрипты по возврату к виду текущей ревизии (даже если в папке имеются несохраненные изменения + новые файлы).

## Заключение

Реализовал простой калькулятор и алгоритм решения Ханойской башни.

Ознакомился с исходниками функции возведения в дробную степень (<https://git.musl-libc.org/cgit/musl/tree/src/math/pow.c>) и пришёл к выводу, что там используется редукция аргумента и переход к возведению экспоненты в степень. Перевёл реализацию логарифма из численных методов за прошлый семестр на C++. Вспомнил, как работать с `constexpr`. Узнал, как работать с `assert` и профилировать код через VS Code.

Вспомнил, как работать с `std::deque` и `std::pair`. Получилось самостоятельно додуматься до алгоритма решения задачи Ханойской башни, которую раньше никогда не решал. Вспомнил, как писать классы, что делает конструктор и деструктор, чем структура отличается от класса. Узнал, как работать с перечислениями типов в шаблонах. Ознакомился с правосторонней ссылкой (rvalue reference).