

# Software Engineering

Moscow Institute of Physics and Technology

# Table of Contents

|   |    |
|---|----|
| 01. Introduction and Brief Overview .....   | 4  |
| 02. Basics of Programming.....              | 5  |
| 03. Object - Oriented Programming .....     | 6  |
| 04. Generic Programming .....               | 7  |
| 05. Software Architecture Patterns.....     | 8  |
| 06. Projects and Libraries .....            | 9  |
| 07. Handling Errors and Debugging.....      | 10 |
| 08. Instruments of Calculus .....           | 11 |
| 09. Detailed Memory Management.....         | 12 |
| 10. Collections and Containers .....        | 13 |
| 11. Iterators and Algorithm Libraries ..... | 14 |
| 12. Text Data Processing .....              | 15 |
| 13. Streams and Data Serialization .....    | 16 |
| 14. Concurrent Programming.....             | 17 |
| 15. Network Technologies and Tools .....    | 18 |

# 00. Mandatory Requirements

## Требования: часть 1

- Ваши решения должны собираться без предупреждений и ошибок с флагами `-Wall` и `-Wextra`.
- Ваши решения должны сопровождаться по крайней мере пятью тестами.
- Ваши решения должны выполняться до конца и проходить тесты без ошибок и неопределенного поведения.
- Ваши решения должны быть адекватно отформатированы в рамках единого стиля.

## Требования: часть 2

- Ваши решения не должны содержать дублирующегося кода, непонятных названий и магических литералов.
- Ваши решения не должны использовать стандартные символьные потоки ввода и вывода.

В случае конфликтов требований в приоритете является выполнение требований, указанных в условиях задач.

# 01. Introduction and Brief Overview

## 01.01 [06.07]

Напишите программу, которая выводит в стандартный символьный поток вывода `std::cout` любую строку и при этом обладает функцией `main` с единственной инструкцией `return 0`. Предложите по крайней мере четыре разных решения. Впервые я столкнулся с этой задачей на техническом собеседовании в крупную российскую компанию. Для ее решения Вам потребуются технологии, которые будут рассматриваться во втором, третьем и шестом модулях данного курса, поэтому Вы можете пропустить эту задачу и вернуться к ней позже. Возможно, Вы немного удивились тому, что первая же задача данного курса обладает настолько неадекватным уровнем сложности. Это своеобразная дань памяти моему детству. Я начал серьезно изучать компьютерные науки и языки программирования в 12 лет, когда проводил летние школьные каникулы на даче у бабушки с дедушкой. Родители подарили мне две книги: Программирование – принципы и практика с использованием C++ Бьёрна Страуструпа и Язык программирования С Брайана Кернигана и Денниса Ритчи. Также у меня имелся простой ноутбук со средой разработки Code::Blocks, однако не было ни интернета, ни даже мобильной связи, потому что дача находится в низине, а сеть можно поймать только на определенном тайном холмике в лесу. Я решил начать изучение с визуально небольшой книги по языку С, быстро проработать ее, а потом приступить к монографии Страуструпа. Опрометчивое решение! В одном из первых заданий просили написать программу, которая удалила бы все комментарии из исходного кода другой программы на языке С. Предположу, что это весьма сложная задача для третьего дня изучения программирования, но я справился, потому что из-за отсутствия связи с внешним миром я просто не понял, что это сложно. Возможно, именно этот случай помог мне определиться с основным направлением всей дальнейшей деятельности. Любопытно, что случилось, если бы мне тогда подарили монографию Искусство программирования Дональда Кнута? Возможно, я стал бы лучше относиться к математике. Пожалуй, стоит провести небольшой эксперимент над собственными детьми.

## 02. Basics of Programming

### 02.01 [02.12]

Реализуйте алгоритм вычисления  $N$ -ого числа ряда Фибоначчи на основе формулы Бине. Используйте тип `double`. Используйте константы. Используйте стандартные функции `std::sqrt` и `std::pow`. Используйте стандартный символьный поток ввода `std::cin` для ввода числа  $N$ . Используйте стандартный символьный поток вывода `std::cout` для вывода числа ряда Фибоначчи. Не сопровождайте Ваше решение данной задачи тестами.

### 02.02 [02.17]

Реализуйте алгоритм вычисления корней алгебраического уравнения второй степени с вещественными коэффициентами  $a$ ,  $b$  и  $c$ . Используйте тип `double`. Используйте ветвления типа `if` для проверки значения коэффициента  $a$  и значения дискриминанта. Используйте константу `epsilon` для корректного сравнения чисел с плавающей точкой с заданной точностью. Используйте стандартную функцию `std::abs`. Используйте стандартный символьный поток ввода `std::cin` для ввода чисел  $a$ ,  $b$  и  $c$ . Используйте стандартный символьный поток вывода `std::cout` для вывода корней уравнения. Не сопровождайте Ваше решение данной задачи тестами.

### 02.03 [02.18]

Реализуйте алгоритм классификации символов таблицы ASCII с десятичными кодами от 32 до 127 включительно на 5 следующих классов: заглавные буквы, строчные буквы, десятичные цифры, знаки препинания, прочие символы. Используйте литералы типа `char`. Используйте ветвление `switch` с проваливанием. Используйте стандартный символьный поток ввода `std::cin` для ввода символов. Используйте стандартный символьный поток вывода `std::cout` для вывода названий классов. Не сопровождайте Ваше решение данной задачи тестами.

### 02.04 [02.20]

Реализуйте алгоритм вычисления всех трехзначных чисел Армстронга. Используйте тройной вложенный цикл `for` для перебора. Не используйте стандартную функцию `std::pow`. Используйте стандартный символьный поток вывода `std::cout` для вывода чисел Армстронга. Не сопровождайте Ваше решение данной задачи тестами.

### 02.05 [02.24]

Реализуйте алгоритм вычисления числа  $e$  на основе суммы членов ряда Маклорена при  $x = 1$  с точностью, заданной числом `epsilon`. Используйте тип `double`. Не вычисляйте факториалы, чтобы не столкнуться с проблемой переполнения. Используйте известное соотношение между членами ряда Маклорена для оптимизации вычисления каждого нового члена ряда на основе предыдущего члена ряда. Используйте цикл `while` для вычисления членов ряда Маклорена до тех пор, пока очередной член ряда не станет меньше числа `epsilon`. Используйте стандартный символьный поток ввода `std::cin` для ввода числа `epsilon`. Используйте стандартный символьный поток вывода `std::cout` для вывода числа  $e$ . Не сопровождайте Ваше решение данной задачи тестами.

### 02.06 [02.31]

Реализуйте алгоритм вычисления наибольшей длины последовательности Коллатца среди всех последовательностей Коллатца, начинающихся со значений от 1 до 100. Используйте тип `unsigned long long int`. Используйте кэширование длин последовательностей Коллатца в контейнере `std::vector` для оптимизации вычисления длины каждой новой последовательности на основе предыдущих последовательностей. Используйте стандартный символьный поток вывода `std::cout` для вывода наибольшей длины последовательности Коллатца и соответствующего ей начального значения. Не сопровождайте Ваше решение данной задачи тестами.

### 02.07 [02.41]

Реализуйте алгоритм вычисления наибольшего общего делителя двух натуральных чисел на основе рекурсивного подхода. Реализуйте алгоритм вычисления наименьшего общего кратного двух натуральных чисел. Используйте тип `int`. Используйте стандартные функции `std::gcd` и `std::lcm` для сравнения результатов тестов.

### 02.08 [02.42]

Доработайте пример 02.42 таким образом, чтобы вместо алгоритма сортировки слиянием использовался алгоритм быстрой сортировки. Обоснуйте временную сложность полученного гибридного алгоритма сортировки.

# 03. Object - Oriented Programming

## 04. Generic Programming

## 05. Software Architecture Patterns



## 06. Projects and Libraries

## 07. Handling Errors and Debugging

## 08. Instruments of Calculus

## 09. Detailed Memory Management

# 10. Collections and Containers

# 11. Iterators and Algorithm Libraries

## 12. Text Data Processing

## 13. Streams and Data Serialization



# 14. Concurrent Programming

# 15. Network Technologies and Tools