

Министерство образования и науки РФ  
Санкт-Петербургский политехнический университет Петра Великого  
Институт компьютерных наук и кибербезопасности  
Высшая школа «Компьютерных технологий и информационных систем»

ОТЧЕТ  
по дисциплине «Практикум по программированию»  
**Лабораторная работа № 2**

**Выполнил:**  
Студент гр. з5130902/20001

Д.Л. Рязанцев

**Проверил**  
Ст. преподаватель

А.М. Журавская

Санкт-Петербург  
2024 г.

## Задание

В соответствии с вариантом **19**, написать программу, которая предлагает пользователю ввести требуемые данные (параметры A,B,C,D,G,M,N,K,W для различных вариантов), проверяет корректность введенных данных (отрицательный или дробный возраст, нулевой год и т.д.), предлагает исправить некорректно введенные данные и вычисляет требуемые в задании условия.

Исходные данные:

Даны 2 натуральных числа M и N.

- Определить сумму M последних цифр числа N
- Количество четных цифр входящих в число N
- Количество нечетных цифр, входящих в число M
- Определить, является ли число M или N простым

## Код программы/Листинг программы

**CMakeLists.txt:**

```
cmake_minimum_required(VERSION 3.0.0)
project(lab2
  DESCRIPTION "Лабораторная работа 2"
  HOMEPAGE_URL "github.com/whs31/education"
  LANGUAGES CXX
)

set(CMAKE_CXX_STANDARD 20)

add_executable(${PROJECT_NAME})
target_sources(${PROJECT_NAME} PRIVATE main.cc)
target_include_directories(${PROJECT_NAME} PRIVATE ${CMAKE_CURRENT_SOURCE_DIR})
```

**main.cc:**

```

/*
 * Лабораторная работа 2
 * Студент: Рязанцев Дмитрий
 */

#include <cmath>
#include <concepts>
#include <iostream>
#include <format>

namespace
{
    [[nodiscard]] constexpr auto sum_of_digits(int const n) -> int { // NOLINT(*-no-recursion)
        return n == 0 ? 0 : n % 10 + sum_of_digits(n / 10);
    }

    [[nodiscard]] constexpr auto sum_of_m_last_digits(int const n, int const m) -> int { // NOLINT(*-no-recursion)
        return m == 0 ? 0 : sum_of_m_last_digits(n / 10, m - 1) + n % 10;
    }

    [[nodiscard]] constexpr auto digits(int const n) -> int { // NOLINT(*-no-recursion)
        return n == 0 ? 0 : 1 + digits(n / 10);
    }

    [[nodiscard]] constexpr auto odd_digits_count(int const n) -> int { // NOLINT(*-no-recursion)
        return n == 0 ? 0 : n % 2 + odd_digits_count(n / 10);
    }

    [[nodiscard]] constexpr auto is_prime(long const n) -> bool
    {
        if(n <= 1)
            return false;
        for(auto i = 2; i * i <= n; ++i)
            if(n % i == 0)
                return false;
        return true;
    }
}

auto main() -> int
{
    using std::string;
    using std::string_view;
    using std::cin;
    using std::cout;
    using std::cerr;
    using std::endl

```

```

auto read_number_from_terminal = [](string_view const name) -> long
{
    cout << std::format("Enter {}:\n", name);
    auto buf = string();
    cin >> buf;
    try {
        return std::stoi(buf);
    } catch(std::invalid_argument const&) {
        cerr << std::format("Incorrect value of {} ({}):\n", name, buf);
        cin.get();
        return 1;
    }
};

auto const m = read_number_from_terminal("M");
auto const n = read_number_from_terminal("N");
if(m <= 0 or n <= 0) {
    cerr << "Incorrect values of M and N (must be > 0)\n";
    cin.get();
    return 1;
}

cout << std::format("\nSum of digits of {} last digits of {} = {}\n", m, n,
::sum_of_m_last_digits(n, m));
cout << std::format("Odd digits count of {} = {}\n", n, ::odd_digits_count(n));
cout << std::format("Even digits count of {} = {}\n", n, ::digits(n) -
::odd_digits_count(n));
cout << std::format("Is N = {} prime? {}\n", n, ::is_prime(n) ? "Yes" : "No");
cout << std::format("Is M = {} prime? {}\n", m, ::is_prime(m) ? "Yes" : "No");

return 0;
}

```

## Пример работы программы



```
PS C:\Users\whs31\Documents\GitHub\doc\education\programming-workshop-spring-2024\labs\2\cmake-build-debug> .\lab2.exe
Enter M:
3
Enter N:
127

Sum of digits of 3 last digits of 127 = 10
Odd digits count of 127 = 2
Even digits count of 127 = 1
Is N = 127 prime? Yes
Is M = 3 prime? Yes
PS C:\Users\whs31\Documents\GitHub\doc\education\programming-workshop-spring-2024\labs\2\cmake-build-debug> |
```

Рисунок 1 – Правильная работа программы



```
PS C:\Users\whs31\Documents\GitHub\doc\education\programming-workshop-spring-2024\labs\2\cmake-build-debug> .\lab2.exe
Enter M:
13
Enter N:
1234567890

Sum of digits of 13 last digits of 1234567890 = 45
Odd digits count of 1234567890 = 5
Even digits count of 1234567890 = 5
Is N = 1234567890 prime? No
Is M = 13 prime? Yes
PS C:\Users\whs31\Documents\GitHub\doc\education\programming-workshop-spring-2024\labs\2\cmake-build-debug> |
```

Рисунок 2 – Правильная работа программы

```
Windows PowerShell
PS C:\Users\whs31\Documents\GitHub\doc\education\programming-workshop-spring-2024\labs\2\cmake-build-debug> .\lab2.exe
Enter M:
-2
Enter N:
-5555
Incorrect values of M and N (must be > 0)
PS C:\Users\whs31\Documents\GitHub\doc\education\programming-workshop-spring-2024\labs\2\cmake-build-debug> |
```

Рисунок 3 – Сообщение об ошибке при попытке задать ненатуральные числа

```
Windows PowerShell
PS C:\Users\whs31\Documents\GitHub\doc\education\programming-workshop-spring-2024\labs\2\cmake-build-debug> .\lab2.exe
Enter M:
asd
Incorrect value of M (asd)
Enter N:
```

Рисунок 4 – Сообщение об ошибке при попытке ввести буквы в поле для чисел

## **Вывод**

В ходе выполнения лабораторной работы я выполнил поставленную задачу по работе с базовыми алгоритмическими операциями над числом на языке C++. Получены навыки работы с оператором получения остатка от деления (%), тернарным оператором, рекурсивными функциями, лямбда-функциями, навыки форматирования вывода с использованием `std::format`, ввод и вывод в окне терминала с использованием потоковых классов стандартной библиотеки.