**МИНОБРНАУКИ РОССИИ**

**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**

**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**

**Кафедра «АПУ»**

**ОТЧЁТ**

**по лабораторной работе № 2**

**по дисциплине «Программирование»**

**«АНАЛИЗ ЧИСЛОВЫХ ПОСЛЕДОВАТЕЛЬНОСТЕЙ»**

|  |  |  |
| --- | --- | --- |
| Студент гр. 4391 | **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** | Мухачев Д. О. |
| Преподаватель | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Писарев А.С. |

Санкт-Петербург

2024

**Цель работы:**

Задачей была представлена последовательность N целых положительных чисел. Необходимо вывести на экран некоторые данные об этой последовательности:

**Ход работы:**

Первым шагом было написание функции проверки числа на простоту:

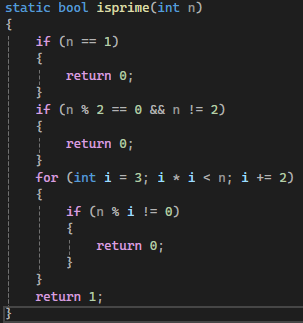
****

Рисунок 1

Далее был написан основной цикл программы, включающий в себя интерфейс, позволяющий вводить последовательность в соответствии с условиями задачи и использующий Unicode для вывода кириллицы:



Рисунок 2

Каждый введенный элемент проверяется на простоту с помощью ранее написанной функции, ведется подсчет суммы и минимума/максимума:

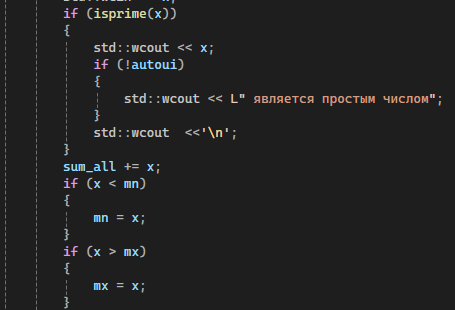


Рисунок 3

Потом происходит проверка делимости элемента на 5, проверка того, является ли элемент степенью числа 2, и сравнение элемента с суммой 2 предыдущих:

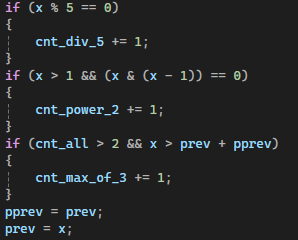


Рисунок 4

После обработки всей последовательности, программа выдает искомые величины в нужном формате:

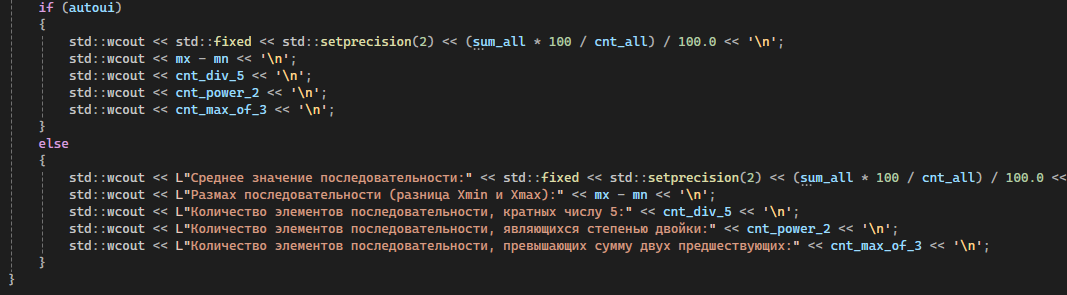


Рисунок 5

В завершении, были подключены все необходимые для работы программы библиотеки:

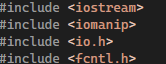


Рисунок 6

**Проверка работоспособности программы:**

Для теста были использованы входные данные из методических материалов:

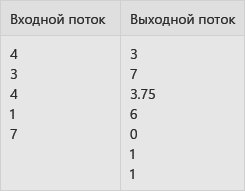


Рисунок 7

Полученные результаты:

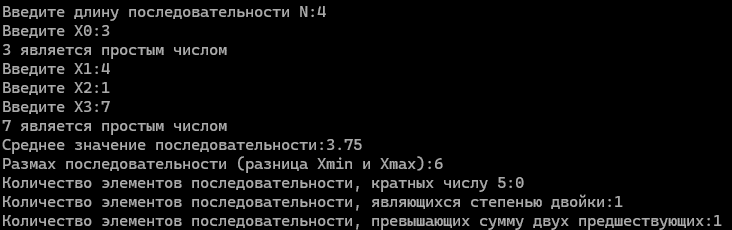


Рисунок 8

**Вывод:** В ходе выполнения лабораторной работы, целью которой была разработка программы для определения некоторых свойств произвольной числовой последовательности, был получен практический опыт в нескольких ключевых областях. Было освоено создание различных типов интерфейсов, использование циклов, условных операторов и разработка функций для решения математических задач. Теоретические знания, полученные на лекциях, помогли в написании алгоритма проверки числа на простоту.

Приложение 1

Код программы

#include <iostream>

#include <io.h>

#include <fcntl.h>

constexpr int M = 3, N = 4;

static inline bool check(double x, double y)

{

if (x < 0 || y < 0)

{

return false;

}

else if (M \* y > M \* N - N \* x)

{

return false;

}

else

{

return true;

}

}

int wmain(int argc, wchar\_t\* argv[])

{

\_setmode(\_fileno(stdout), \_O\_U8TEXT);

\_setmode(\_fileno(stdin), \_O\_U8TEXT);

\_setmode(\_fileno(stderr), \_O\_U8TEXT);

double x = 1, y = 1;

if (argc <= 1 || std::wcsncmp(argv[1], L"false", sizeof(L"false") - 1) != 0)

{

while (1)

{

std::wcout << L"Новая точка?\nВведите значение 0 в координаты X и Y, чтобы выйти из программы.\nВведите значение координаты X:";

std::wcin >> x;

std::wcout << L"Введите значение координаты Y:";

std::wcin >> y;

if (x == 0 && y == 0)

{

break;

}

else

{

std::wcout << L"Точка (" << x << L", " << y << (check(x, y) ? L") " : L") не ") << L"входит в треугольник с вершинами (0, 0); (" << M << L", 0); (0, " << N << L")\n";

}

} return 0;

}

while (1)

{

std::wcin >> x >> y;

if (x == 0 && y == 0)

{

break;

}

else

{

std::wcout << (check(x, y) ? "YES" : "NO") << '\n';

}

}

return 0;

}