**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:

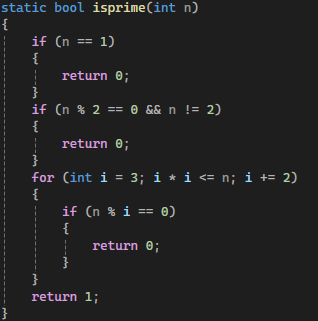
****

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:

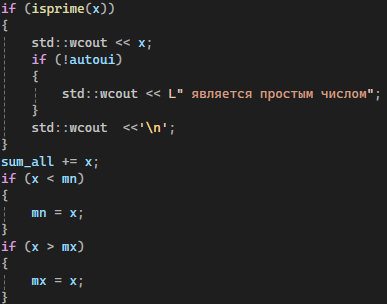


Рисунок 3

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

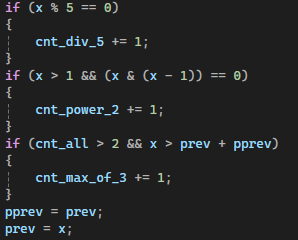


Рисунок 4

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

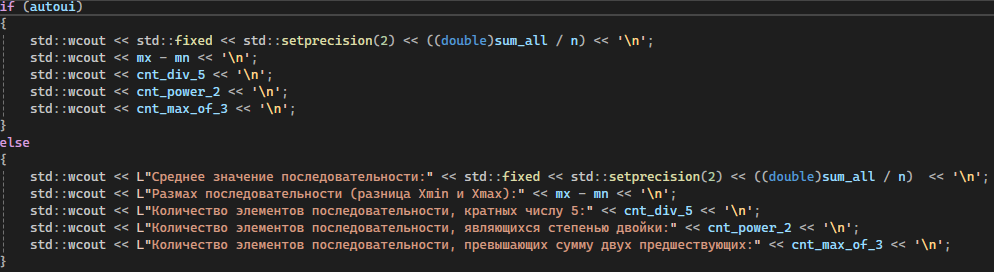


Рисунок 5

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

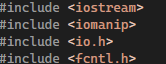


Рисунок 6

**Проверка работоспособности программы:**

Для теста были использованы входные данные из методических материалов:

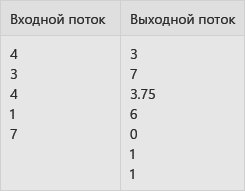


Рисунок 7

Полученные результаты:

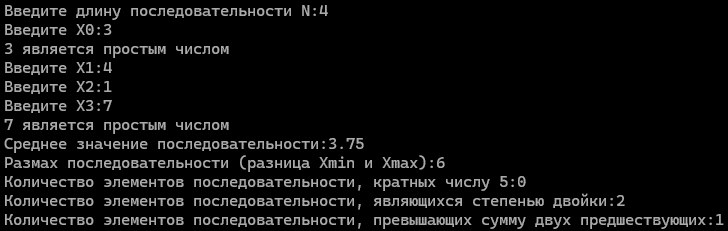


Рисунок 8

**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 <iomanip>

#include <io.h>

#include <fcntl.h>

static bool isprime(int n)

{

if (n == 1)

{

return 0;

}

if (n % 2 == 0 && n != 2)

{

return 0;

}

for (int i = 3; i \* i <= n; i += 2)

{

if (n % i == 0)

{

return 0;

}

}

return 1;

}

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

{

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

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

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

bool autoui = true;

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

{

autoui = false;

}

int i = 0;

int x = -1;

int sum\_all = 0;

int mn = INT32\_MAX, mx = INT32\_MIN;

int cnt\_div\_5 = 0;

int cnt\_power\_2 = 0;

int cnt\_max\_of\_3 = 0;

int prev = 0, pprev = 0;

int n;

if (!autoui)

{

std::wcout << L"Введите длину последовательности N:";

}

std::wcin >> n;

for (i = 0; i < n; i++)

{

if (!autoui)

{

std::wcout << L"Введите X" << i << L":";

}

std::wcin >> x;

if (isprime(x))

{

std::wcout << x;

if (!autoui)

{

std::wcout << L" является простым числом";

}

std::wcout <<'\n';

}

sum\_all += x;

if (x < mn)

{

mn = x;

}

if (x > mx)

{

mx = x;

}

if (x % 5 == 0)

{

cnt\_div\_5 += 1;

}

if (x > 0 && (x & (x - 1)) == 0)

{

cnt\_power\_2 += 1;

}

if (i > 1 && x > (prev + pprev))

{

cnt\_max\_of\_3 += 1;

}

pprev = prev;

prev = x;

}

if (autoui)

{

std::wcout << std::fixed << std::setprecision(2) << ((double)sum\_all / n) << '\n';

std::wcout << mx - mn << '\n';

std::wcout << cnt\_div\_5 << '\n';

std::wcout << cnt\_power\_2 << '\n';

std::wcout << cnt\_max\_of\_3 << '\n';

}

else

{

std::wcout << L"Среднее значение последовательности:" << std::fixed << std::setprecision(2) << ((double)sum\_all / n) << '\n';

std::wcout << L"Размах последовательности (разница Xmin и Xmax):" << mx - mn << '\n';

std::wcout << L"Количество элементов последовательности, кратных числу 5:" << cnt\_div\_5 << '\n';

std::wcout << L"Количество элементов последовательности, являющихся степенью двойки:" << cnt\_power\_2 << '\n';

std::wcout << L"Количество элементов последовательности, превышающих сумму двух предшествующих:" << cnt\_max\_of\_3 << '\n';

}

}