Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**

**(ФГБОУ ВО «КубГУ»)**

**Факультет компьютерных технологий и прикладной математики**

**Кафедра вычислительных технологий**

**ЛАБОРАТОРНАЯ РАБОТА №1**

**Дисциплина: Криптографические протоколы**

Работу выполнил: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Д.Е.Гиренко

Направление подготовки: 02.03.02 Фундаментальная информатика и информационные технологии

Преподаватель: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_А.А. Крамаренко

**Цель работы:** реализовать программный продукт нахождения функции Эйлера от числа двумя способами (по определению и с помощью формулы). Сравнить эффективность алгоритмов для набора из 100 чисел, каждое из которых больше 10’000’000.

**Ход работы:**

1) Напишем реализацию функции Эйлера по определению: количество чисел, меньших самого числа, которые взаимно простые с ним:

ll firstEuler(ll num) {

ll sum = 1;

for (int i = 2; i < num; i++) {

if (gcd(num, i) == 1) {

sum += 1;

}

}

return sum;

}

2) Напишем реализацию функции Эйлера по формулам:

ll secondEuler(ll num) {

ll ret = 1, i, pow;

for (i = 2; i <= sqrt(num) && num != 1; i++)

{

pow = 1;

while (!(num % i))

{

num /= i;

pow \*= i;

}

ret \*= (pow - (pow / i));

}

if (num != 1)

ret \*= (num - 1);

return ret;

}

3) Проведём испытания при помощи библиотеки chrono согласно цели:

void testEulersByNumbersCount(ll num) {

auto begin = chrono::system\_clock::now();

auto end = chrono::system\_clock::now();

auto diff = end - begin;

vector<ll> randNumbers = vector<ll>(num);

for (int i = 0; i < num; i++) {

ll randNum = rand();

if (randNum <= 10000000)

randNum += 10000000;

randNumbers[i] = randNum;

}

begin = chrono::system\_clock::now();

for (int i = 0; i < num; i++)

firstEuler(randNumbers[i]);

end = chrono::system\_clock::now();

diff = end - begin;

cout << chrono::duration <double, milli>(diff).count() / 1000 << "s of fisrt" << endl;

begin = chrono::system\_clock::now();

for (int i = 0; i < num; i++)

secondEuler(randNumbers[i]);

end = chrono::system\_clock::now();

diff = end - begin;

cout << chrono::duration <double, milli>(diff).count() / 1000 << "s of second" << endl;

}

В результате получилось для одного и того же набора данных, что функция Эйлера по определению считается примерно за 80 секунд, когда функция Эйлера по формулам считается за 0.000838 секунд, что в разы быстрее.

**Листинг программы**

#include <iostream>

#include <chrono>

#include <random>

#include <vector>

using namespace std;

typedef long long ll;

ll gcd(ll a, ll b) {

while (a && b)

if (a > b) a %= b;

else b %= a;

return a + b;

}

ll firstEuler(ll num) {

ll sum = 1;

for (int i = 2; i < num; i++) {

if (gcd(num, i) == 1) {

sum += 1;

}

}

return sum;

}

ll secondEuler(ll num) {

ll ret = 1, i, pow;

for (i = 2; i <= sqrt(num) && num != 1; i++)

{

pow = 1;

while (!(num % i))

{

num /= i;

pow \*= i;

}

ret \*= (pow - (pow / i));

}

if (num != 1)

ret \*= (num - 1);

return ret;

}

void testEulersByNumbersCount(ll num) {

auto begin = chrono::system\_clock::now();

auto end = chrono::system\_clock::now();

auto diff = end - begin;

vector<ll> randNumbers = vector<ll>(num);

for (int i = 0; i < num; i++) {

ll randNum = rand();

if (randNum <= 10000000)

randNum += 10000000;

randNumbers[i] = randNum;

}

begin = chrono::system\_clock::now();

for (int i = 0; i < num; i++)

firstEuler(randNumbers[i]);

end = chrono::system\_clock::now();

diff = end - begin;

cout << chrono::duration <double, milli>(diff).count() / 1000 << "s of fisrt" << endl;

begin = chrono::system\_clock::now();

for (int i = 0; i < num; i++)

secondEuler(randNumbers[i]);

end = chrono::system\_clock::now();

diff = end - begin;

cout << chrono::duration <double, milli>(diff).count() / 1000 << "s of second" << endl;

}

int main()

{

testEulersByNumbersCount(100);

}

// 1 - 80 секунд.

// 2 - 0.000838 секунд