Министерство образования Республики Беларусь

Учреждение образования

«Брестский Государственный технический университет»

Кафедра ИИТ

**Лабораторная работа №4**

По дисциплине «Криптографические методы защиты информации»

Тема: «Тестирование чисел на простоту и построение больших простых чисел»

**Выполнил:**

Студент 2 курса

Группы ИИ-23

Романюк А.П.

**Проверил:**

Хацкевич А. С.

Брест 2024

**Цель:** Освоить основные программные методы тестирования чисел на простоту

**Ход работы:**

**Вариант 9**



#include <iostream>

#include <cstdlib>

#include <ctime>

#include <cmath>

#include <vector>

#include "ap/ap.hpp"

ap\_uint<32768> long\_pow(ap\_uint<32768>a, ap\_uint<32768>b) {

ap\_uint<32768> result = 1;

for (int i = 0; i < b-1; i++) {

result \*= a;

}

return result;

}

ap\_uint<32768> modulo(ap\_uint<32768>a, ap\_uint<32768>b) {

if (a < b) {

return a;

}

else {

return a - (a / b) \* b;

}

}

int true\_rand\_num(int p = 8) {

unsigned int random\_number = rand() % (1 << p); // Генерируем случайное число от 0 до 2^p - 1

random\_number |= (1u << (p - 1)); // Устанавливаем старший бит в 1

random\_number |= 1u; // Устанавливаем младший бит в 1

return random\_number;

}

int NOD(int a,int b) {

if (a % b == 0) {

return b;

}

if (b % a == 0) {

return a;

}

if (a > b) {

return NOD(a % b, b);

}

return NOD(a, a % b);

}

std::vector<int> sieve\_of\_eratosthen(int n) {

std::vector<bool> is\_prime(n + 1, true);

std::vector<int> primes;

is\_prime[0] = is\_prime[1] = false; // 0 и 1 не являются простыми числами

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

if (is\_prime[i]) {

for (int j = i \* i; j <= n; j += i) {

is\_prime[j] = false;

}

}

}

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

if (is\_prime[i]) {

primes.push\_back(i);

}

}

return primes;

}

bool bruteforce\_check(int n, std::vector<int> primes) {

for (int prime : primes) {

if (prime >= n) break;

if (n % prime == 0) return 0;

}

return 1;

}

bool rubber\_miller\_test(int n) {

int t = n - 1;

int s = 0;

while (t % 2 == 0) {

t /= 2;

s += 1;

}

for (int k = 0; k < 5; k++) {

int a = rand() % (n - 3) + 2;

ap\_uint<32768> x = modulo(long\_pow(a, t),n);

if (x == 1 || x == n - 1) continue;

for (int b = 0; b < s - 1; b++) {

x = x \* x % n;

if (x == 1) return 0;

if (x == n - 1) break;

}

if (x != n - 1) return 0;

}

return 1;

}

int main() {

srand(time(nullptr));

int p = 16; //требуемая длина простого числа

int n;

std::vector<int> primes = sieve\_of\_eratosthen(2000);

while (true) {

n = true\_rand\_num(p);

if (!bruteforce\_check(n, primes)) continue;

if (rubber\_miller\_test(n)) break;

}

std::cout << "простое число: " << n;

return 0;

}





**Вывод:** освоил основные методы тестирования чисел на простоту.