Министерство образования Республики Беларусь

Учреждение образования

«Брестский Государственный технический университет»

Кафедра ИИТ

**Лабораторная работа №4**

По дисциплине «Криптографические методы защиты информации»

Тема: «Факторизация составного числа»

**Выполнил:**

Студент 2 курса

Группы ИИ-21

Кирилович

**Проверил:**

Хацкевич М. В.

Брест 2023

**Цель:** освоить алгоритмы факторизации составного числа.

**Ход работы:**

Метод ро-Поларда.

#include "stdio.h"

#include "stdlib.h"

#include "string.h"

#include "time.h"

#include <vector>

std::vector<int> factors;

std::vector<int> divisors = {

2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37,

41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89,

97, 101, 103, 107, 109, 113, 127, 131, 137, 139,

149, 151, 157, 163, 167, 173, 179, 181, 191, 193,

197, 199, 211, 223, 227, 229, 233, 239, 241, 251

};

int isprime(int num)

{

if ((num \* num) % 24 == 1) return true;

if (num == 2 || num == 3) return true;

return false;

}

int gcd(int a, int b)

{

return b == 0 ? a : gcd(b, a % b);

}

int pollard\_rho(int n)

{

if(n % 2 == 0) return 2;

int x = rand() % n + 1;

int c = rand() % n + 1;

int y = x;

int g = 1;

while(g == 1)

{

x = ((x \* x) % n + c) % n;

y = ((y \* y) % n + c) % n;

y = ((y \* y) % n + c) % n;

g = gcd(abs(y - x), n);

}

return g;

}

void factorize\_r\_polard(int n)

{

if(n == 1) return;

if(isprime(n))

{

factors.push\_back(n);

return;

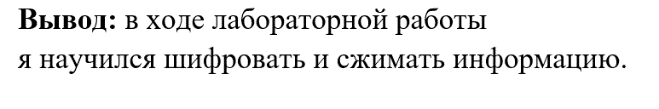
}

int divisor = pollard\_rho(n);

factorize\_r\_polard(divisor);

factorize\_r\_polard(n / divisor);

}



void factorize\_prime(int n)

{

for (int i = 0; i < divisors.size(); i++)

{

int divisor = divisors[i];

while (n % divisor == 0)

{

factors.push\_back(divisor);

n /= divisor;

}

}

if(n > 1) factors.push\_back(n);

}

void factorize\_combine(int n)

{

if(isprime(n))

{

factors.push\_back(n);

return;

}

for (int i = 0; i < divisors.size(); i++)

{

int divisor = divisors[i];

while (n % divisor == 0)

{

factors.push\_back(divisor);

n /= divisor;

}

}

if (n > 1)

{

srand(time(NULL));

int divisor = pollard\_rho(n);

factorize\_r\_polard(divisor);

factorize\_r\_polard(n / divisor);

}

}

int main(int argc, char\*\* argv)

{

int choice = std::atoi(argv[1]);

int n = 1234567890;

srand(time(NULL));

if(choice == 1) factorize\_prime(n);

if(choice == 2) factorize\_r\_polard(n);

if(choice == 3) factorize\_combine(n);

for (int i = 0; i < factors.size(); ++i)

{

printf("%d ", factors[i] );

}

printf("\n");

return 0;