Київський національний університет імені Тараса Шевченка
Факультет комп'ютерних наук та кібернетики

Лабораторна робота
На тему “**Основні алгоритми теорії чисел та криптографії**”

Виконав студент 3-го курсу
групи МІ-31 Гришечкін Тихон

Київ – 2023

Хід виконання

Мова виконання - **C#**

1. Обчислення функцій Ейлера та Мьобіуса. Знаходження найменшого спільного кратного набору чисел.

```
BigInteger a = 36449955;
BigInteger b = 114643293;
BigInteger result = Algorithms_NumberTheory.lcm(a, b);
Console.WriteLine($"The least common multiple of {a} and {b} is {result}");
```

```
The least common multiple of 36449955 and 114643293 is 906255231165
```

```
Number: 2, Mobius function: 1
Number: 3, Mobius function: 1
Number: 5, Mobius function: 1
Number: 7, Mobius function: 1
Number: 8, Mobius function: 0
Number: 10, Mobius function: -1
Number: 11, Mobius function: 1
Number: 25, Mobius function: 0
Number: 35, Mobius function: -1
Number: 50, Mobius function: 0
Number: 42, Mobius function: 1
Number: 100, Mobius function: 0
```

```
Number: 13, Euler's totient function: 12
Number: 14, Euler's totient function: 6
Number: 20, Euler's totient function: 8
Number: 21, Euler's totient function: 12
Number: 30, Euler's totient function: 8
Number: 48, Euler's totient function: 16
Number: 49, Euler's totient function: 42
Number: 50, Euler's totient function: 20
Number: 53, Euler's totient function: 52
Number: 100, Euler's totient function: 40
```

2. Розв’язання системи лінійних порівнянь за модулем (китайська теорема про лишки).

```
BigInteger k = 5;
BigInteger[] num = { 6, 7, 5, 11, 23 };
BigInteger[] rem = { 3, 1, 0, 5, 3};
Console.WriteLine($"x%{num[0]} = {rem[0]}");
Console.WriteLine($"x%{num[1]} = {rem[1]}");
Console.WriteLine($"x%{num[2]} = {rem[2]}");
Console.WriteLine($"x%{num[3]} = {rem[3]}");
Console.WriteLine($"x%{num[4]} = {rem[4]}");
Console.WriteLine($"x = {ChineseRemainderSolution(num, rem, k)}");
```

```
x%6 = 3
x%7 = 1
x%5 = 0
x%11 = 5
x%23 = 3
x = 23325
```

3. Обчислення символів Лежандра та Якобі.

```
for (int p = 3;p<14; p++)
{
    if (isPrimeMiller(p))
    {
        Console.WriteLine($"Prime {p} legandre symbols:");
        for (int j = 0; j < p; ++j)
        {
            Console.Write(LegangreSymbol(j, p) + " ");
        }
        Console.WriteLine();
    }
}
```

```
Prime 3 legandre symbols:
0 1 -1
Prime 5 legandre symbols:
0 1 -1 -1 1
Prime 7 legandre symbols:
0 1 1 -1 1 -1 -1
Prime 11 legandre symbols:
0 1 -1 1 1 1 -1 -1 -1 1 -1
Prime 13 legandre symbols:
0 1 -1 1 1 -1 -1 -1 -1 1 1 -1 1
```

```
Odd 3 jacobi symbols:
0 1 -1
Odd 5 jacobi symbols:
0 1 -1 -1 1
Odd 7 jacobi symbols:
0 1 1 -1 1 -1 -1
Odd 9 jacobi symbols:
0 1 1 0 1 1 0 1 1
```

4. Один алгоритм факторизації довгих цілих чисел на вибір: ро-алгоритм Полларда

```
BigInteger bigInteger = BigInteger.Parse("258354295128624765762058752000");
List<BigInteger> bigNumbers = new List<BigInteger>();
PollardRhoFactorization pollard = new PollardRhoFactorization();
bigNumbers= pollard.PollardRho(bigInteger);
Console.WriteLine("Factors of " + bigInteger + " are:");
foreach (BigInteger bigNumber in bigNumbers)
{
    Console.Write(bigNumber+", ");
}
```

```
Factors of 258354295128624765762058752000 are:
2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 5, 5, 7, 7, 11, 17, 17, 17, 29, 29, 31, 37, 331, 653, 3761,
```

Число 258354295128624765762058752000

Множители: $2^{11} * 3^5 * 5^2 * 7^2 * 11 * 17^3 * 29^2 * 31 * 37 * 331 * 653 * 3761$

5. Один алгоритм знаходження дискретного логарифма на вибір: ро-алгоритм Полларда або алгоритм «великий крок – малий крок».

```
// test for discrete logarithm
long m = 2669442613;
BigInteger a = 57957;
BigInteger b = 2146571266;
Console.WriteLine("Finding descrete logarithm for a^x = b (mod m)");
Console.WriteLine($"{a}^x={b} (mod {m})");
BigInteger x = DiscreteAlgorithm(a, b, m);
Console.WriteLine($"Solution: {a}^{x}={b} (mod {m})");
Console.WriteLine($"Check: {a}^{x}={st(a, x, m)} (mod {m})");
```

```
Finding descrete logarithm for a^x = b (mod m)
57957^x=2146571266 (mod 2669442613)
Solution: 57957^144=2146571266 (mod 2669442613)
Check: 57957^144=2146571266 (mod 2669442613)
```

6. Алгоритм знаходження дискретного квадратного кореня.

```
// test for discrete root
BigInteger n = 15680206471;
BigInteger k = 2;
BigInteger a = 849384;

List<BigInteger> list = DiscreteRoot.DiscreteRootAlg(n,k,a);
Console.WriteLine("Discrete root of " + a + " in " + k + " degree modulo " + n + " is:");
foreach (BigInteger i in list)
{
    Console.WriteLine(i);
}
```

```
Discrete root of 849384 in 2 degree modulo 15680206471 is:
4267658495
11412547976
```

7. Один алгоритм перевірки чисел на простоту на вибір: алгоритм Соловея-
Штрассена або алгоритм Міллера-Рабіна.

```
// test for prime test Miller-Rabin
List<BigInteger> testNumbers = new List<BigInteger>();
testNumbers.Add(BigInteger.Parse("7631858471"));
testNumbers.Add(BigInteger.Parse("30827361108343263889"));
testNumbers.Add(BigInteger.Parse("2050819583291293484373335228051"));
testNumbers.Add(BigInteger.Parse("2825432293086848558659223343124688905693"));
testNumbers.Add(BigInteger.Parse("61677636203847380433186644045689462130175176083507"));
testNumbers.Add(88463 * 23099);
testNumbers.Add(BigInteger.Parse("25903306907652919531"));
testNumbers.Add(BigInteger.Parse("807146509990936358417611974 83"));
testNumbers.Add(BigInteger.Parse("18446744073709551617"));
testNumbers.Add(BigInteger.Parse("18446744073709551617"));
testNumbers.Add(BigInteger.Parse("1713000920401"));
foreach (var testNumber in testNumbers)
{
    Console.WriteLine($"Test for prime test Miller-Rabin for {testNumber} is {isPrimeMiller(testNumber)}");
}
```

```
Test for prime test Miller-Rabin for 7631858471 is True
Test for prime test Miller-Rabin for 30827361108343263889 is True
Test for prime test Miller-Rabin for 2050819583291293484373335228051 is True
Test for prime test Miller-Rabin for 2825432293086848558659223343124688905693 is True
Test for prime test Miller-Rabin for 61677636203847380433186644045689462130175176083507 is True
Test for prime test Miller-Rabin for 2043406837 is False
Test for prime test Miller-Rabin for 25903306907652919531 is False
Test for prime test Miller-Rabin for 807146509990936358417611974 83 is False
Test for prime test Miller-Rabin for 18446744073709551617 is False
Test for prime test Miller-Rabin for 18446744073709551617 is False
Test for prime test Miller-Rabin for 1713000920401 is False
```

Також протестовано на числах Кармайкла до 10^{12}.

8. Криптосистема RSA або криптосистема Рабіна (на вибір).

```
// test for prime RsaCryptoSystem
BigInteger p = BigInteger.Parse("389156779108590923 89");
BigInteger q = BigInteger.Parse("632706255929676824 11");
BigInteger msg = BigInteger.Parse("3576486846835735 73346246");

var keys = RsaCryptosystem.RsaKeys(p, q, msg);

Console.WriteLine($"Rsa public key {keys.Item1}, {keys.Item2}");
Console.WriteLine($"Rsa private key {keys.Item3}, {keys.Item4}");
```

```
Message data = 35764868468357357346246
Encrypted data = 20028241750291988207025686849733558269 35
Original Message Sent = 35764868468357357346246
Rsa public key 24622192867944884026398353888922592698 79, 7
Rsa private key 24622192867944884026398353888922592698 79, 10552368371976378868018496080236139264 63
```

Github:
https://github.com/DeDTihoN/NumberTheoryAlgos