Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Основы теории чисел и их использование в криптографии

Студент: Стрелковская В.А.

ФИТ 4 курс 2 группа

Преподаватель: Хартанович А.А.

Минск 2025

**1 Теоретические сведения**

Теория чисел, также известная как **высшая арифметика,** – раздел математики, посвящённый изучению натуральных чисел и других близких по свойствам величин. В зависимости от применяемых методов, теория чисел делится на несколько направлений.

**Множество целых чисел** (обозначается *Z*) включает все **действительные числа** без дробной части: {..., –3, –2, –1, 0, 1, 2, 3, ...}.

**Натуральные числа** представляют собой подмножество целых чисел и образуют множество *N*: {1, 2, 3, ...}.

Одним из ключевых понятий теории чисел является **делимость**. Если для некоторого целого числа a и натурального числа *b* существует такое целое число *q*, что выполняется равенство *a* = *b* × *q*, то говорят, что a делится на *b*. В этом случае *b* называют делителем числа *a*, а само число *a* — кратным числу *b.*

**Обозначение**: a ⋮ b – a делится на b, или b | a – b делит a.

Из этого следует, что:

* любое натуральное число является делителем нуля;
* единица является делителем любого целого числа;
* любое натуральное число является делителем самого себя

Делитель *a* считается **собственным делителем** числа *b*, если выполняется неравенство 1 < |a| < |b|. Если это условие не соблюдается, то такой делитель называется **несобственным**.

**Свойство 1 собственного делителя:** положительный наименьший собственный делитель составного числа n не превосходит √n.

**Свойство 2 собственного делителя**. Положительный наименьший собственный делитель составного числа n есть простое число.

Всякое целое число а можно представить с помощью положительного целого числа b равенством вида а = bq + r, 0 ≤ r ≤ b. Число q называется **неполным частным**, а число r – **остатком от деления** а на b.

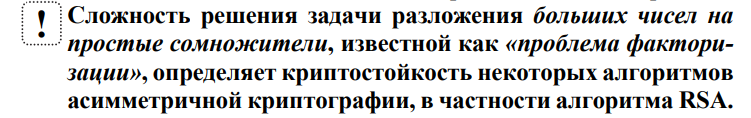
Натуральное число *n* > 1 называется **простым**, если оно делится только на 1 и на само себя. Если число имеет дополнительные делители, оно называется **составным**. Единица не считается ни простым, ни составным числом.

**Свойство 1 простых чисел**. Любое составное число представляется уникальным образом в виде произведения простых чисел; иначе еще говорят, что разложение числа на простые множители однозначно.

**Основная теорема арифметики. Всякое натуральное число n, кроме 1, можно представить как произведение простых множителей.**

**Свойство 2 простых чисел**. Простых чисел бесконечно много, причем существует примерно n / ln(n) простых чисел, меньших числа n.

**Свойство 3 простых чисел**. Наименьший простой делитель составного числа n не превышает √n, поэтому для проверки простоты числа достаточно проверить его делимость на 2 и на все нечетные (а еще лучше простые) числа, не превосходящие √n; как видим, данное свойство коррелирует со свойством 1 собственного делителя.



**Свойство 4 простых чисел**. Любое четное число, большее 2, представимо в виде суммы двух простых чисел, а любое нечетное, большее 5, представимо в виде суммы трех простых чисел.

**Свойство 5 простых чисел**. Для любого натурального n, большего 1, существует хотя бы одно простое число на интервале от n до 2n.

Существуют правила, способные заметно сократить время вычислений:

**Правило 1**. Воспользоваться свойством 3 простых чисел.

**Правило 2**. Если последняя цифра анализируемого числа является четной, то это число заведомо составное.

**Правило 3**. Числа, делящиеся на 5, всегда оканчиваются пятеркой или нулем. Если младшим разрядом анализируемого числа являются 5 или 0, то такое число не является простым.

**Правило 4.** Если анализируемое число делится на 3, то и сумма его цифр тоже обязательно делится на 3.

**Правило 5**. Основано на свойстве делимости на 11. Нужно из суммы всех нечетных цифр числа вычесть сумму всех четных его цифр. Четность и нечетность определяется счетом от младшего разряда.

**Правило 6**. Основано на свойстве делимости на 7 и 13. Нужно разбить анализируемое число на тройки цифр, начиная с младших разрядов. Просуммировать числа, стоящие на нечетных позициях, и вычесть из них сумму чисел на четных.

Если два простых числа отличаются на 2, то их называют **числами-близнецами.**

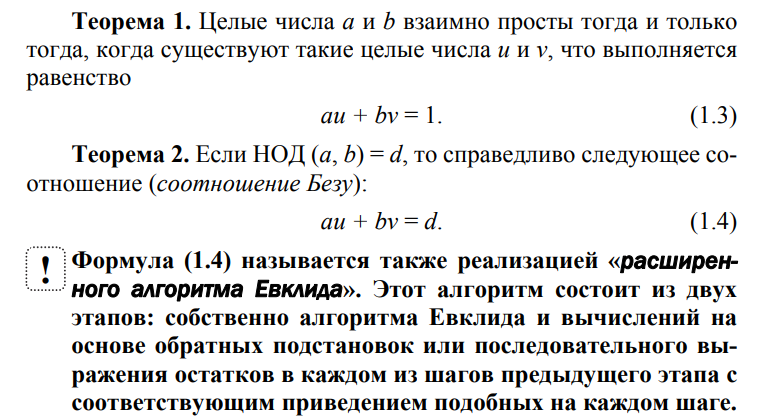
**«Решето Эратосфена».** Его суть в последовательном исключении из списка целых чисел от 1 до n (или из сокращенного диапазона, например от m до n, 1 < m ≤ n) чисел, кратных 2, 3, 5 и другим простым числам, уже найденным «решетом». Шаги:

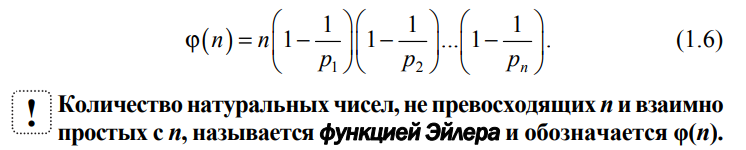
* выписать подряд все целые числа от двух (либо от m) до n (2, 3, 4, …, n). Пусть некоторая переменная (напрмер, s) изначально равна 2 – первому простому числу;
* удалить из списка числа от 2s до n, считая шагами по s (это будут числа, кратные s: 2s, 3s, 4s, …);
* найти первое из оставшихся чисел в списке, большее чем s, и присвоить значению переменной s это число;
* повторять шаги 2 и 3, пока возможно.

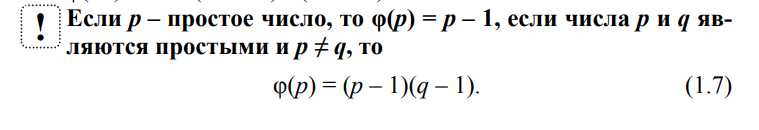
Наибольшее целое число, которое делит без остатка числа a и b, называется **наибольшим общим делителем** этих чисел – НОД (a, b)/

Простым и эффективным средством вычисления НОД (a, b) является **алгоритм Евклида**.

**Взаимно простыми** являются целые числа, наибольший общий делитель которых равен 1.

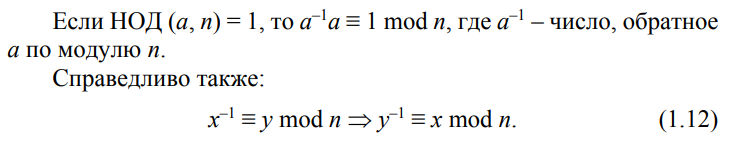


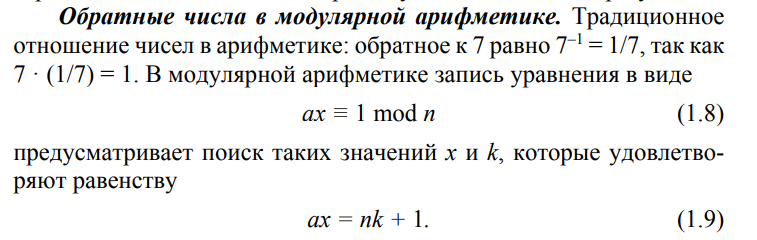


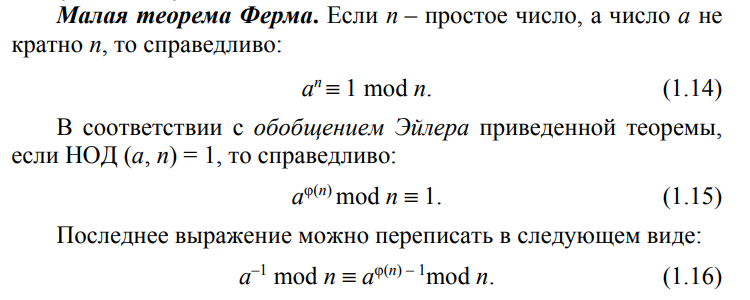


Модулярная арифметика, введённая Карлом Фридрихом Гауссом, изучает остатки от деления чисел на заданный модуль *n*. Если число *a* при делении на *n* даёт остаток *b*, то записывают *a* ≡ *b* (mod *n*), что читается как «*a* сравнимо с *b* по модулю *n*».

Иногда b называют **вычетом** по модулю n, а числа a и b называют **сравнениями** (по модулю n).







**2 Практическое задание**

**Поиск простых чисел в интервале [354, 397]**

Шаг 1. Записываем все числа:

354, 355, 356, 357, 358, 359, 360, 361, 362, 363, 364, 365, 366, 367, 368, 369, 370, 371, 372, 373, 374, 375, 376, 377, 378, 379, 380, 381, 382, 383, 384, 385, 386, 387, 388, 389, 390, 391, 392, 393, 394, 395, 396, 397

Шаг 2. Удаляем числа, делящиеся на 2:

355, 357, 359, 361, 363, 365, 367, 369, 371, 373, 375, 377, 379, 381, 383, 385, 387, 389, 391, 393, 395, 397

Шаг 3. Удаляем числа, делящиеся на 3:

359, 361, 367, 373, 379, 383, 385, 389, 391, 397

Шаг 4. Удаляем числа, делящиеся на 5:

359, 367, 373, 379, 383, 389, 397

Шаг 5. Удаляем числа, делящиеся на 7:

359, 367, 373, 379, 383, 389, 397

Шаг 6. Удаляем числа, делящиеся на 11, 13, 17, 19:

359, 367, 373, 379, 383, 389, 397

Результат. Простые числа: 359, 367, 373, 379, 383, 389, 397

Пункт 3: Каноническое разложение: 354 = 2 \* 3 \* 59; 397 = 397

Пункт 4: Проверка числа 354 397 на простоту. Число 354397 не является простым, так как делится на 397 (354397 = 397 \* 893).

Пункт 5: НОД(354, 397)

397 = 354 \* 1 + 43

354 = 43 \* 8 + 10

43 = 10 \* 4 + 3

10 = 3 \* 3 + 1

3 = 1 \* 3 + 0

НОД(354, 397) = 1

**Основное задание**

Приложение должно реализовывать вычисление НОД двух либо трех чисел, а также выполнять поиск простых чисел.

Для вычисления НОД чисел был разработан метод, представленный на листинге 1. Данный метод считает НОД чисел с помощью алгоритма Евклида.

|  |
| --- |
| static int GCD(int a, int b)  {  while (b != 0)  {  int temp = b;  b = a % b;  a = temp;  }  return Math.Abs(a);  }  // НОД трёх чисел  static int GCD(int a, int b, int c)  {  return GCD(GCD(a, b), c);  } |

Листинг 1 – Метод вычисление НОД для двух и трех чисел

Результат выполнения функции представлен на рисунке 2 для различного количества чисел.

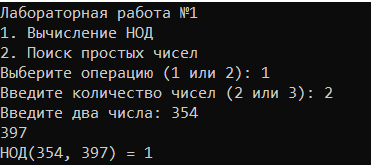
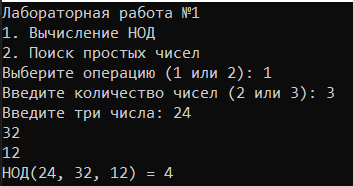
 

Рисунок 2 – Результаты нахождения НОД

Исходя из предоставленного рисунка можно понять, что программа корректно вычисляет НОД для двух и трех чисел.

Также была реализована возможность поиска простых чисел с помощью решето «Эратосфена» до n. Код представлен на листинге 2.

// Решето Эратосфена — поиск простых чисел до n

static List<int> SieveOfEratosthenes(int n)

{

bool[] prime = new bool[n + 1];

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

prime[i] = true;

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

{

if (prime[p])

{

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

prime[i] = false;

}

}

List<int> primes = new List<int>();

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

if (prime[i]) primes.Add(i);

return primes;

}

// Поиск простых чисел в диапазоне [m, n]

static List<int> PrimesInRange(int m, int n)

{

List<int> allPrimes = SieveOfEratosthenes(n);

return allPrimes.FindAll(p => p >= m);

}

Листинг 2 – Поиск простых чисел

Результат выполнения поиска до числа n представлен на рисунке 2.

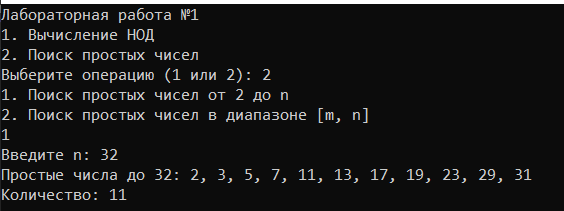


Рисунок 2 – Поиск простых чисел до числа n

Результат выполнения поиска простых чисел в диапазоне от m до n представлен на рисунке 3.

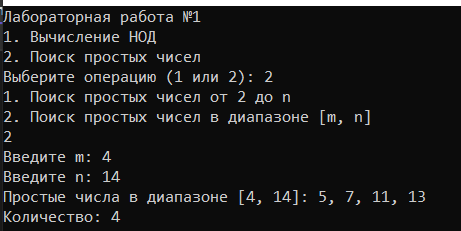


Рисунок 3 – Поиск простых чисел в диапазоне

**Заключение**

В рамках выполнения лабораторной работы были успешно решены поставленные задачи, направленные на закрепление теоретических знаний в области теории чисел и их практического применения в криптографии. Разработанное консольное приложение на языке C# позволило реализовать ключевые операции: вычисление наибольшего общего делителя (НОД) для двух и трех чисел с использованием алгоритма Евклида, поиск простых чисел в заданных интервалах методом Решета Эратосфена, разложение чисел на простые множители в канонической форме, а также проверку числа, образованного конкатенацией, на простоту.