Учреждение образования

«Белорусский государственный технологический университет»

**Кафедра информационных систем и технологий**

**«Отчёт по лабораторной работе №3»**

“Основы теории чисел и их использование в криптографии”

**Выполнила:** студентка 3 курса

4 группы специальности ПОИТ

Кравцова Диана Вячеславовна

**Проверил:** преподаватель

Блинова Евгения Александровна

Минск 2021

**Теоретические сведение**

В основе современной криптографии лежит теория чисел. Теория чисел или высшая арифметика – раздел математики, изучающий натуральные числа и иные похожие величины. В зависимости от используемых методов в теории чисел рассматривают несколько направлений. Нас будут интересовать вопросы делимости целых чисел, вычисления наибольшего общего делителя (НОД), разложение числа на простые множители, малая теорема Ферма́, теорема Эйлера, элементы теории вычетов.

Множество всех целых чисел (обозначим буквой Z) есть набор всех действительных чисел без дробной части: {..., -3, -2, -1, 0, 1, 2, 3, ...}.

Натуральные числа являются подмножеством целых чисел и образуют множество N: {1, 2, 3, ...}.

Делимость – одно из основных понятий теории чисел. Если для некоторого целого числа a и натурального числа b существует целое число q, такое, что bq=a, то говорят, что число a делится на b. В этом случае b называется делителем числа a, а a называется кратным числа b. При этом используются следующие обозначения: a⋮b – a делится на b или b|a – b делит a. Из последнего определения следует, что:

• любое натуральное число является делителем нуля;

• единица является делителем любого целого числа;

• любое натуральное число является делителем самого себя.

Делитель a называется собственным делителем числа b, если 1<|a|

Каждое натуральное число, большее единицы, делится, по крайней мере, на два числа: на 1 и на само себя. Если число не имеет делителей, кроме самого себя и единицы, то оно называется простым, а если у числа есть еще делители, то составным.

Натуральное число n называется простым, если n > 1 и не имеет положительных делителей, отличных от 1 и n.

Простое число не делится без остатка ни на одно другое число.

Перечислим несколько важных свойств простых чисел.

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

Это свойство вытекает из основной теоремы арифметики.

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

n = p1 · p2 · p3 · ... · pz, z > 1. (1.1)

Для того, чтобы представить относительно небольшое число в виде простых сомножителей, достаточно уметь делить числа столбиком. Однако при этом следует придерживаться некоторых простых правил. Для первого деления нужно выбрать наименьшее простое число большее 1, которое делит исходное число без остатка. Частное от первого деления также нужно разделить с учетом указанных ограничений. Процесс деления продолжаем до тех пор, пока частным не будет 1.

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

Свойство 3. 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.

Из соотношения n=qp натуральных чисел, больших единицы, следует, что либо p, либо q принадлежит отрезку от 2 до √n.

Поиск сомножителей числа n может вестись, например, перебором всех простых чисел до √n. Однако, если множители – большие простые числа, то на их поиск может потребоваться много времени

Сложность решения задачи разложения больших чисел на простые сомножители, известной как проблема факторизации, определяет криптостойкость некоторых алгоритмов асимметричной криптографии, в частности алгоритма RSA.

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

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

Натуральное число n называется составным, если n > 1 и имеет, по крайней мере, один положительный делитель, отличный от 1 и n.

Единица не считается ни простым числом, ни составным.

Вернемся к собственному делителю.

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

Так как простое число не делится ни на какое другое, кроме себя самого, очевидный способ проверки числа n на простоту – разделить n на все числа n1 и проанализировать наличие остатка от деления. Этот способ «в лоб» часто реализуется в компьютерных программах. Однако перебор может оказаться достаточно трудоемким, если на простоту нужно проверить число с количеством цифр в несколько десятков.

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

Таких чисел не очень много. Например, ими являются 5 и 7, 29 и 31, 149 и 151.

Всякое натуральное число n > 1либо является простым числом, либо имеет простой делитель.

Понятно, что в криптографии используются числа, проверка на простоту которых производится гораздо дольше, и для работы с этими числами требуются специальные программные средства. К вопросу проверки чисел на простоту мы еще вернемся. Здесь же отметим, что первый алгоритм нахождения простых чисел, не превышающих n, был придуман Эратосфеном во 2 в. до н. э. и известен сейчас как «решето Эратосфена». Его суть в последова- тельном исключении из списка целых чисел от 1 до n чисел (или из сокращенного диапазона, например, от m до n, 1.

Для нахождения всех простых чисел не больше заданного числа n в соответствии с «решетом Эратосфена» нужно выполнить следующие шаги:

1. Выписать подряд все целые числа от двух (либо от m) до n (2, 3, 4, …, n).

Пусть некоторая переменная (положим s) изначально равна 2 – первому простому числу.

1. Удалить из списка числа от 2s до n, считая шагами по s (это будут числа кратные s: 2s, 3s, 4s, …).
2. Найти первое из оставшихся чисел в списке, большее чем s, и присвоить значению переменной s это число.
3. Повторять шаги 2 и 3, пока возможно.

Понятие делимости чисел является одним из важных в теории чисел. С этим понятием, а также с его производным – общим делителем связаны другие важнейшие (в частности, для криптографии) понятия: наибольшего общего делителя (НОД) и взаимно простых чисел.

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 метод или алгоритм Евклида (примеры его использования приведены в [2]. В основе алгоритма лежит Определение 5. В соответствии с этим определением используется цепочка вычислений двумя исходными (начальными) числами: а и b:

аi= biqi + ri, 0 ≤ ri ≤ bi. (1.2)

При i = 0 в (1.2) аi и bi соответствуют как раз числам а и b. Последний ненулевой остаток (ri, i ≥ 0) соответствует НОД (a, b).

Чтобы найти НОД нескольких чисел (например, a, b, c), достаточно найти НОД двух чисел (например, НОД(a, b) = d) потом НОД полученного (НОД(a, b)) и следующего числа (НОД(c, d) и т. д.

Таким образом, чтобы вычислить НОД k чисел, нужно последовательно вычислить (k–1) НОД. Последнее вычисление дает искомый результат.

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

Теорема 1. Целые числа a и b взаимно просты тогда и только тогда, когда существуют такие целые u и v, что выполняется равенство

аu + bv = 1. (1.3)

Теорема 2. Если НОД (a, b) = d , то справедливо следующее соотношение (соотношение Безу):

аu + bv = d. (1.4)

Формула (1.4) называется также реализацией «расширенного алгоритма Евклида». Этот алгоритм состоит из двух этапов: собственно алгоритма Евклида и вычислений на основе обратных подстановок или последовательного выражения остатков в каждом из шагов предыдущего этапа с соответствующим приведением подобных на каждом шаге.

Исследованием целых чисел занимался швейцарский математик Леонард Эйлер (Leonard Euler). Один из важных вопросов его исследования: сколько существует натуральных чисел, не превосходящих некоторое число n и взаимно простых с n? Ответ на этот вопрос связан с каноническим разложением числа n на простые множители.

Число натуральных чисел, не превосходящих n и, взаимно простых с n, называется функцией Эйлера и обозначается φ(n).

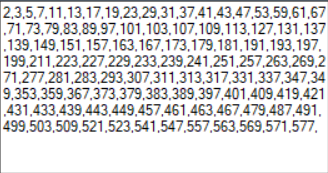
Если p – простое число, то φ(p) = p – 1, если числа p и q являются простыми и p ≠ q, то

φ(p) = (p – 1) (q – 1).

**Практическая часть**

**1) Используя L\_PROST, найти все простые числа в интервале [2, n]. Значение n соответствует варианту из табл. 1.2, указанному преподавателем. Подсчитать количество простых чисел в указанном интервале. Сравнить это число с n/ln(n) (см. выше пример 15).**

n = 577;

****

Количество простых чисел в интервале [0,577]: 106.

577 / ln(577) = 90,7540602

**2) Повторить п. 1 для интервала [m, n]. Сравнить полученные результаты с «ручными» вычислениями, используя «решето Эратосфена» (см. примеры 11 и 12).**

m=540

Количество простых чисел в интервале [521,553]: 7.



**3) Записать числа m и n в виде произведения простых множителей (форма записи – каноническая).**

n = 577

577= 577

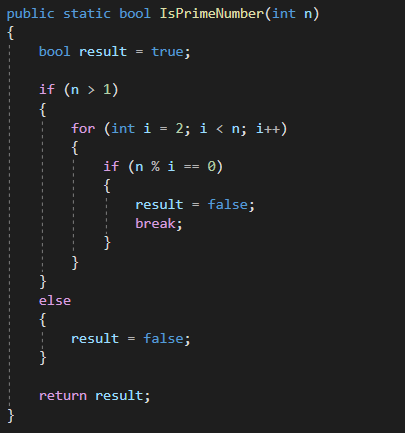
m =540

540 = 2\*2\*5\*3\*3\*3

**4) Проверить, является ли число, состоящее из конкатенации цифр m ǀǀ n (табл. 1.2), простым.**

540577

Для решения этого задания была разработана функция, которая проверяет остаток от деления числа 540577 на каждое число из промежутка [2,540576].



Число является простым.

**5) Найти НОД (540, 577).**

577 = 540\*1 + 37;

540 = 37\*14 + 22;

37 = 22\*1 + 15;

22 = 15\*1 + 7;

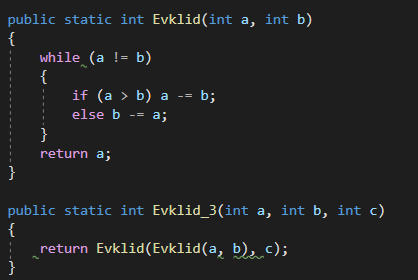
15 = 7\*2 + 1;

7 = 7\*1 + 0;

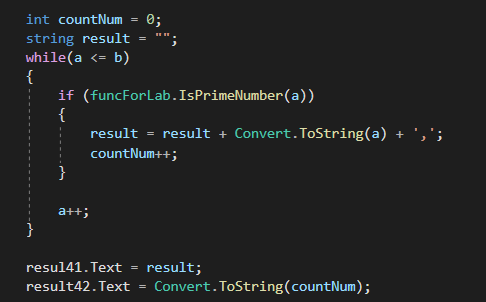
НОД(540;577) = 1;

**6) Разработать авторское приложение в соответствии с целью лабораторной работы. Приложение должно реализовывать следующие операции: • вычислять НОД двух либо трех чисел; • выполнять поиск простых чисел.**

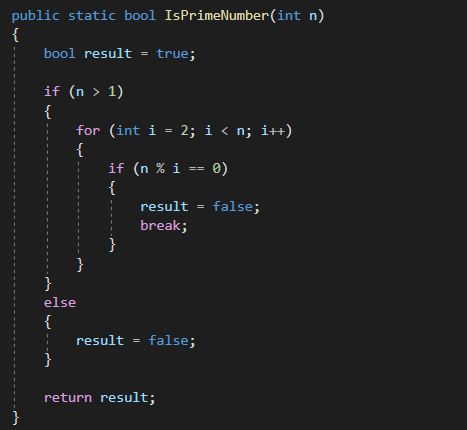
Функции вычисляющие НОД для 2-х и 3-х чисел:

****

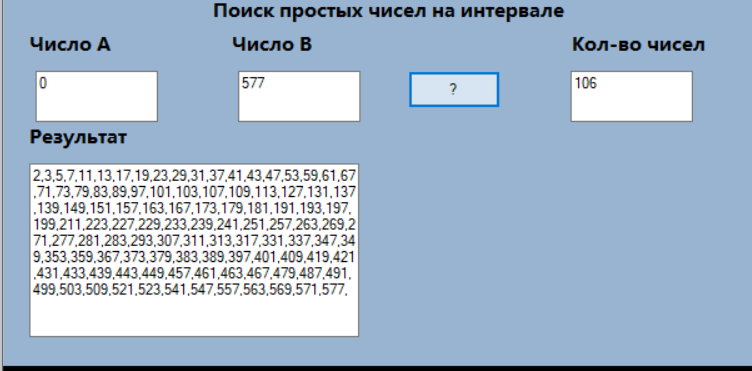
Код выполняющий поиск простых чисел:

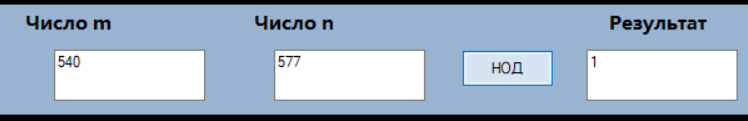


Функция IsPrimeNumber проверяет является ли число простым.

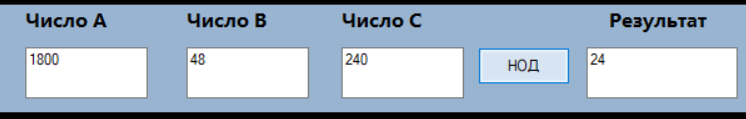


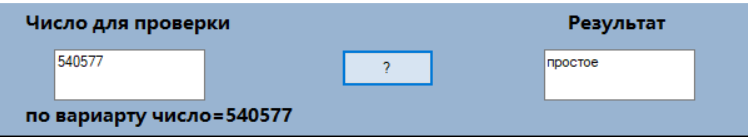
7) Результаты работы приложения

****

****

****

****

****