Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет информационных технологий

Кафедра информационных систем и технологий

**Лабораторная работа № 3**

«Основы теории чисел и их использование в криптографии»

Выполнил:

Студент: Белицкий В.Д.

ФИТ 2 курс 5 группа

Преподаватель: Савельева М.Г.

Минск 2023

1. **Описание приложения**

Приложение написано на языке программирования C# и позволяет:

* Найти НОД двух чисел
* Найти НОД трёх чисел
* Найти простые числа в заданном диапазоне
* Привести число в каноническую форму
* Проверить является ли число простым
* Поиск простых чисел используя «решето Эратосфена»

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 наибольшегообщего делителя (НОД).

1. **Методика выполнения расчетов**

В данной лабораторной работе была поставлена цель создания приложения, позволяющее произвести расчёт НОД двух либо трёх чисел, а также выполнить поиск простых чисел из диапазона. На следующих листингах представлены методы, реализующий данную функциональность.

public static int Compute(int x, int y)

{

while (x != 0 && y != 0)

{

if (x > y)

{

x -= y;

}

else

{

y -= x;

}

}

return Math.Max(x, y);

}

**Листинг 2.1 – метод, реализующий нахождение НОД**

Метод Compute используется для получения НОД двух чисел путём исполнения алгоритма Евклида.

public static void FindSimple(int m, int n)

{

int counter = 0;

if (n < m)

{

Console.WriteLine("Неверный промежуток");

}

Console.Write($"Простые числа интервала [{m},{n}]: ");

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

{

if (IsSimple(i))

{

Console.Write(i.ToString() + " ");

counter++;

}

}

Console.WriteLine();

Console.WriteLine($"Количество простых чисел: {counter}");

Console.WriteLine("n/ln(n) = " + n / Math.Log(n));

}

**Листинг 2.2 – Нахождение простых чисел в заданном диапазоне**

public static void Canon(int num)

{

int numCopy = num;

Dictionary<int,int> CanonList = new ();

for (int i = 0; num % 2 == 0; num /= 2)

{

if(CanonList.ContainsKey(2))

{

CanonList[2] += 1;

}

else

{

CanonList.Add(2, 1);

}

}

for (int i = 3; i <= num;)

{

if (num % i == 0)

{

if (CanonList.ContainsKey(i))

{

CanonList[i] += 1;

}

else

{

CanonList.Add(i, 1);

}

num /= i;

}

else

{

i += 2;

}

}

Console.Write(numCopy + " = 1");

foreach (int key in CanonList.Keys)

{

Console.Write($" \* {key}");

if(CanonList[key] != 1)

{

Console.Write($"^{CanonList[key]}");

}

}

}

Листинг 2.3 – Разложение на множители

public static bool IsSimple(int x)

{

for (int i = 2; Math.Pow(i, 2) <= x; i++)

{

if (x % i == 0)

{

return false;

}

}

return true;

}

**Листинг 2.4 – Определение является ли число простым**

public static List<uint> SieveEratosthenes(uint m, uint n)

{

var numbers = new List<uint>();

//заполнение списка числами от 2 до n-1

for (var i = 2u; i < n; i++)

{

numbers.Add(i);

}

for (var i = 0; i < numbers.Count; i++)

{

for (var j = 2u; j < n; j++)

{

//удаляем кратные числа из списка

numbers.Remove(numbers[i] \* j);

}

}

for (int i = 0; i < numbers.Count(); i++)

{

if (numbers[i] < m)

{

numbers.RemoveAt(i);

i--;

}

}

numbers.Add(n);

Console.WriteLine("Колличество чисел: "+numbers.Count);

return numbers;

}

**Листинг 2.5 – «Решето Эратосфена»**

**Таблица 2.1**

**«Ручное решето Эратосфена»**

|  |  |  |  |
| --- | --- | --- | --- |
| Ручное решето Эратосфена (2-18) | | | |
|  | S2 | S3 | S4 |
| 2 | 2 | 2 | 2 |
| 3 | 3 | 3 | 3 |
| 4 | 5 | 5 | 5 |
| 5 | 7 | 7 | 7 |
| 6 | 9 | 11 | 11 |
| 7 | 11 | 13 | 13 |
| 8 | 13 | 17 | 17 |
| 9 | 15 |  |  |
| 10 | 17 |  |  |
| 11 |  |  |  |
| 12 |  |  |  |
| 13 |  |  |  |
| 14 |  |  |  |
| 15 |  |  |  |
| 16 |  |  |  |
| 17 |  |  |  |
| 18 |  |  |  |

**3. Результаты работы приложения**

Для выполнения расчетов достаточно необходимо запустить приложение, выбрать необходимый пункт задания и ввести числа. Рисунки 3.1, 3.2, 3.3 показывают необходимые расчеты и вызовы методов, требуемые в данной лабораторной работе.

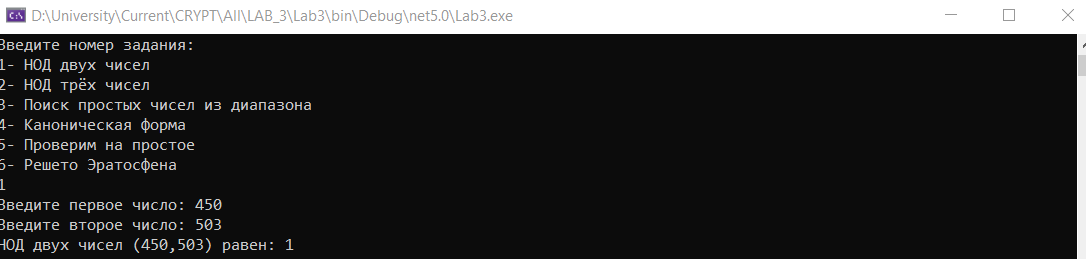


Рисунок 3.1 – Результат работы метода Compute с двумя числами

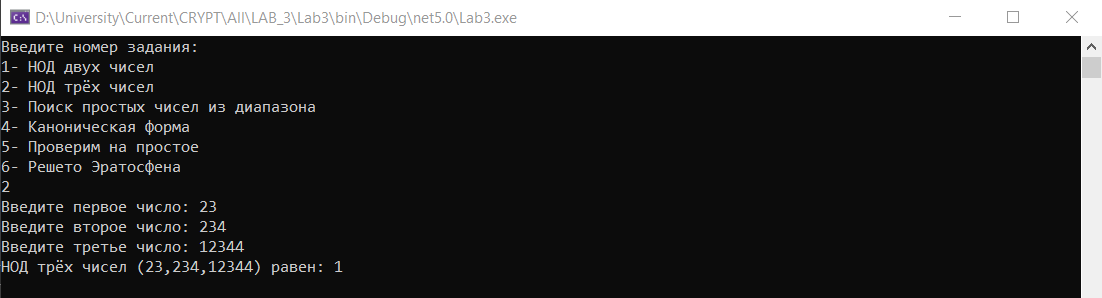


Рисунок 3.2 – Результат работы метода Compute с тремя числами

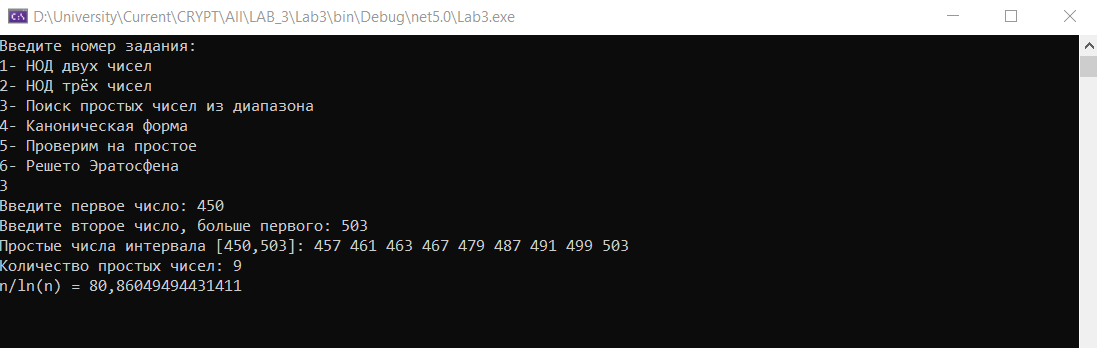


Рисунок 3.3 – Результат работы метода FindSimple

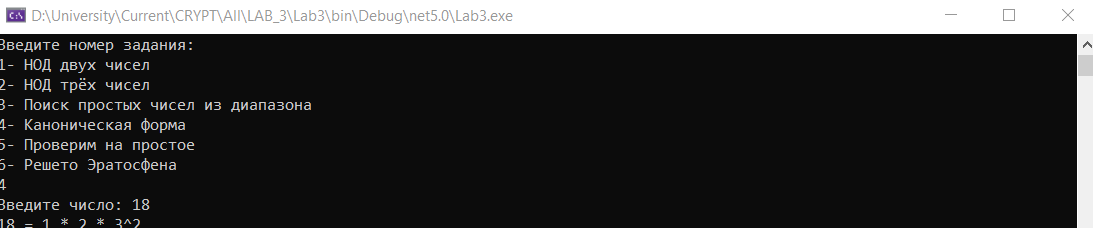


Рисунок 3.4 – Результат работы метода Canon

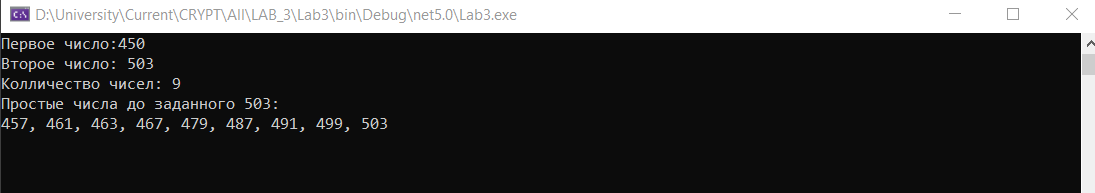


Рисунок 3.5 – Результат работы метода SieveEratosthenes

**Вывод**

В ходе изучения теоретических материалов лабораторной работы и выполнения её практической части были изучены основы теории чисел, используемые в криптографии. Изучены основные свойства простых и составных чисел, взаимно простых чисел, а также признаки делимости чисел. На практике полученные знания были закреплены через создание собственного приложения, реализующего функционал вычисления НОД двух и трёх чисел и получения списка простых чисел в заданном диапазоне.