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: Лэ Н.З.

ФИТ 2 курс 4 группа

Преподаватель: Берников О.В.

Минск 2020

1. **Описание приложения**

Приложение написано на языке программирования C# и позволяет провести расчет НОД двух либо трех чисел, а также найти все простые числа в заданном промежутке и рассчитать обратное по модулю число, применив расширенный алгоритм Евклида.

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), разложение числа на простые множители, малаятеорема Ферма́ , теорема Эйлера, элементы теории вычетов.

1. **Методика выполнения расчетов**

Сначала мною был рассчитан наибольший общий делитель для двух, а затем и для трех чисел. На рисунке 2.1 отображена функция, отвечающая за вычисления НОД по алгоритму Евклида. Для вычисления НОД трех чисел достаточно лишь вызвать данную функцию второй раз, поместив в качестве первого параметра НОД, вычисленный для двух чисел. Расчеты в данной функции производятся с введенными пользователем с консоли значениями.

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

|  |
| --- |
| static int Nod(int a, int b)  {  while (x != 0 && y != 0)  {  if (x > y)  {  x -= y;  }  else  {  y -= x;  }  }  return Math.Max(x, y);  } |

Таблица 2.1 – Функция вычисления НОД

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.

Для определения, является ли число простым была разработана функция IsSimple(int N), представленная в таблице 2.2.

|  |
| --- |
| private static bool IsSimple(int x)  {  for (int i = 2; Math.Pow(i, 2) <= x; i++)  {  if (x % i == 0)  {  return false;  }  }  return true;  } |

Таблица 2.2 – Функция определения, является ли число простым

Затем для вычисления всех простых числе в заданном диапазоне, выбранном в соответствии с моим 3 вариантом, данная функция была вызвана в цикле, показанном на рисунке 2.3.

|  |
| --- |
| int counter = 0;  if (n < m)  {  Console.WriteLine("Неверный промежуток");  }  Console.Write($"Простые числа интервала [{m},{n}]: ");  for (int i = m; i <= n; i++)  {  if (IsSimple(i))  {  Console.Write(i.ToString() + " ");  counter++;  }  }  Console.WriteLine();  Console.WriteLine($"Число простых чисел {counter}"); |

Таблица 2.3 – Вычисление простых чисел в диапазоне от 367 до 401

Затем по заданию требовалось разработать функцию вычисления обратного по модулю числа, т. е. разработать реализацию расширенного алгоритма Евклида. Расчеты в данной функции производятся с введенными пользователем с консоли значениями. Описание и внешний вид функции представлен на рисунке 2.4.

|  |
| --- |
| private static void InverseMod(int a, int m)  {  a = a % m;  for (int x = 1; x < m; x++)  {  if ((a \* x) % m == 1)  {  Console.WriteLine($"Обратный элемент {a} по модулю {m} это {x}");  }  }  } |

Таблица 2.4 – Функция расчета обратного по модулю числа

1. **Результаты работы приложения**

Для выполнения расчетов достаточно лишь запустить консольное приложение, которое рассчитает вышеописанные функции и отобразит в консольном окне результаты.

Рисунок 3.1 показывает необходимые расчеты и вызовы методов, требуемые в данной лабораторной работе.

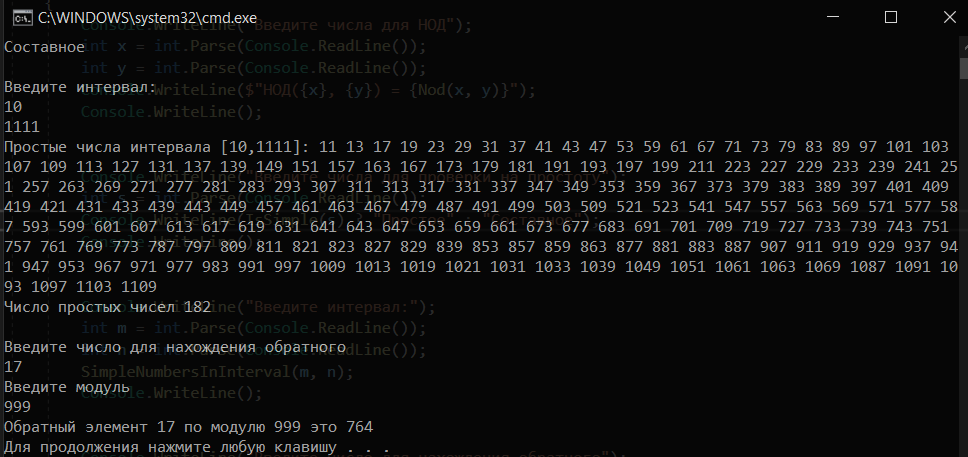


Рисунок 3.1 – Результат работы.

**Вывод**

В ходе этой работы мною были приобретены практические навыки выполнения операций с числами для решения задач в области криптографии и разработки приложений для автоматизации этих задач. Также мною были закреплены теоретические знания по высшей арифметике, я научилась практически решать задачи с использованием простых и взаимно простых чисел, вычислений по правилам модулярной арифметики и  
нахождению обратных чисел по модулю, а также ознакомилась с работой base-64 кодировщиков. А также мною было разработано приложение для реализации указанных преподавателем в соответствии с моим вариантом  
операций с числами.