МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждение образования «БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ

ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №3 НА ТЕМУ:**

**Основы теории чисел и их использование в криптографии**

Выполнила студентка 3 курса 5 группы

Авсюкевич Полина Вадимовна

Минск 2024

**Задание 1:** Разработать авторское приложение в соответствии с целью лабораторной работы. Разработать авторское приложение в соответствии с целью лабораторной работы. Приложение должно реализовывать следующие операции: вычислять НОД двух и трех чисел, выполнять поиск простых чисел.

Реализация данной функциональности представлена в листинге 1:

|  |
| --- |
| static int CalculateGCD(int a, int b)  {  while (a != 0 && b != 0)  {  if (a > b)  a = a % b;  else  b = b % a;  }  return a + b;  }  static int CalculateGCDOfThree(int a, int b, int c)  {  return CalculateGCD(CalculateGCD(a, b), c);  }  static List<int> FindPrimesInRange(int min, int max)  {  List<int> primes = new List<int>();  int[] sieve = new int[max + 1];  for (int i = 0; i <= max; i++)  sieve[i] = 1;  for (int i = 2; i <= Math.Sqrt(max); i++)  {  if (sieve[i] == 1)  {  int j = 2;  while (i \* j <= max)  {  sieve[i \* j] = 0;  j++;  }  }  }  for (int i = 2; i <= max; i++)  {  if (sieve[i] == 1)  primes.Add(i);  }  primes.RemoveAll(num => num < min);  return primes;} |

Листинг 1 – Реализация задания 1

**Задание 2:** С помощью приложения выполнить задания по условиям п. 1 и 2.

Для выполнения данного задания были использованы функции из листинга 1, пример вывода представлен на рисунке 1.

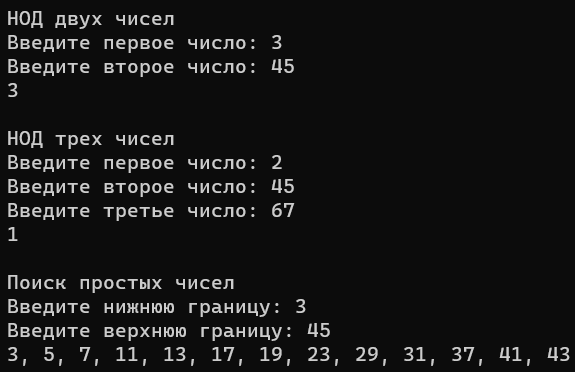


Рисунок 1– Пример вывода для задания 2

Количество подсчитанных простых чисел в интервале примерно совпадает с *n/*ln*(n)*.

**Задание 3:** Записать числа *m* и *n* в виде простых множителей (использовать каноническую форму записи).

Для выполнения этого задания был использован следующий код, представленный в листинге 2:

|  |
| --- |
| static string GetPrimeFactors(int n)  {  List<int> factors = new List<int>();  int divisor = 2;  while (divisor \* divisor <= n){  if (n % divisor == 0){  factors.Add(divisor);  n /= divisor;}  else  divisor++;}  if (n > 1)  factors.Add(n);  string canonicalForm = $"{n} = {factors[0]}";  for (int i = 1; i < factors.Count; i++){  canonicalForm += $" \* {factors[i]}"; }  return canonicalForm;  } |

Листинг 2 – Функция для реализации задания 3

**Задание 4:** Проверить, является ли число, состоящее из конкатенации цифр *m* || *n* простым. Для выполнения этого задания код, была разработана функция, ее реализация представлена в листинге 3.

|  |
| --- |
| static bool IsNumberPrime(int n)  {  int divisorCount = 0;  for (int i = 2; i <= n / 2; i++)  {  if (n % i == 0)  divisorCount++;  }  return divisorCount <= 0;  } |

Листинг 3– Задание 4

Результат выполнения представлен на рисунке 2.



Рисунок 2– Результат выполнения задания 4

**Задание 5:** Найдите НОД (*m*, *n*). Для выполнения данного задания был использован код из задания 1, с подстановкой заданных значений. Результат выполнения представлен на рисунке 3.



Рисунок 3– Результат выполнения задания 5

Решето Эратосфена представлено на рисунке 4.

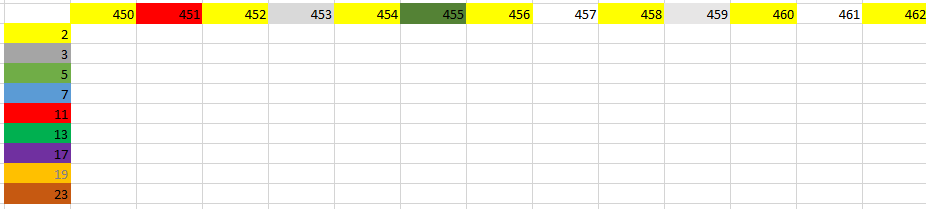


Рисунок 4– Решето Эратосфена

**Вывод:** В результате выполнения лабораторной работы я приобрела практические навыки в области числовых операций, необходимых для решения задач, связанных с криптографией. Эти умения включают в себя выполнение различных операций с числами, таких как вычисление наибольшего общего делителя (НОД), поиск простых чисел, а также работу с простыми множителями чисел. В рамках лабораторной работы я также приобрела навыки разработки приложений, направленных на автоматизацию этих числовых операций.