Министерство образования Республики Беларусь

Учреждение образования «БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Отчет к лабораторной работе № 1:

«Основы теории чисел

и их использование в криптографии»

Выполнил:

студент 4 курса 1 группы

Халалеенко А. Н.

Преподаватель:

Сазонова Д.В.

Минск 2024

**Цель**: приобретение практических навыков выполнения операций с числами для решения задач в области криптографии и разработка приложений для автоматизации этих операций.

**Задачи:**

1. Закрепить теоретические знания по высшей арифметике.

2. Научиться практически решать задачи с использованием

простых и взаимно простых чисел, вычислений по правилам модулярной арифметики и нахождению обратных чисел по модулю.

3. Ознакомиться с особенностями реализации готового программного средства L\_PROST и особенностями выполнения с его

помощью операций над простыми числами.

4. Разработать приложение для реализации указанных преподавателем операций с числами.

5. Результаты выполнения лабораторной работы оформить в виде описания разработанного приложения, методики выполнения эксперимента с использованием приложения и результатов эксперимента. 1 Теоретические сведения

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.

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, при котором bq = a, то говорят, что число a делится на b. В этом случае b называется делителем числа a, а a называется кратным числу 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

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.

Если число не имеет делителей, кроме самого себя и единицы, то оно называется простым, а если у числа есть еще делители, то составным.

Основная теорема арифметики. Всякое натуральное число n, кроме 1, можно представить как произведение простых множителей: n = p1p2p3...pz, z > 1.

2 Практическая часть

|  |
| --- |
| internal class simpleNumbers  {  public static string getSimpleNumbers(string aText, string bText)  {  string result;  int a = aText == null ? 0 : int.Parse(aText);  int b = bText == null ? 0 : int.Parse(bText);  List<int> numb = new List<int> { };  for (int i = 2; i <= b; i++)  {  numb.Add(i);  }  for (int i = 0; i < numb.Count; i++)  {  for (int j = 2; j < b; j++)  numb.Remove(numb[i] \* j);  }  if (numb.Count != 0)  {  result = "Простые числа от " + a + " до " + b + ": ";  int res=0;  foreach (int w in numb)  {  if (w > a)  {  result += w + ", ";  res++;  }  }  result += "\nКоличество простых чисел: " + res;  }  else  result = "Простых чисел в данном диапозоне нет";    return result;  }  } |

Листинг 1 – класс для подсчёта простых чисел

|  |
| --- |
| internal class nodCreator  {  public static int nodeResult;  public static string getNod(string aText, string bText, string? cText)  {  int a = aText == null ? 0 : int.Parse(aText);  int b = bText == null ? 0 : int.Parse(bText);  int c = cText == null ? 0 : int.Parse(cText);  GetNodResult(a, b);  if (c != 0)  {  GetNodResult(nodeResult, c);  }  return "НОД: " + nodeResult;  }  static void GetNodResult(int a, int b)  {  if (a != 0 && b != 0)  {  if (a > b)  {  a = a % b;  }  else  {  b = b % a;  }  GetNodResult(a, b);  }  else  nodeResult = a + b;  }  } |

Листинг 2 – подсчёт НОД

Вывод

В ходе выполнения лабораторной работы были изучены и приобретены практические навыки разработки и использования приложений для реализации алгоритма вычисления простых чисел и их НОД.