Белорусский государственный технологический университет

Кафедра «Информационных систем и технологий»

Лабораторная работа №1

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

Выполнил: Нестер Владислав Александрович

**Минск 2020**

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

|  |  |  |
| --- | --- | --- |
| **Вариант** | **m** | **n** |
| 9 | 587 | 621 |

**Задание**

1. *Используя L\_PROST, найти все простые числа в интервале [2, n]. Подсчитать количество простых чисел в указанном интервале. Сравнить это число с n/ln(n).*

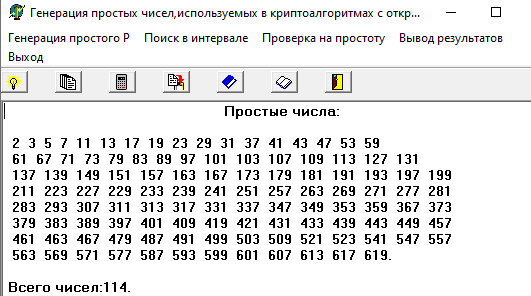


Рисунок 1 – Поиск простых чисел в интервале [2, 621]

На рисунке 1 представлен результат поиска простых чисел в интервале [2, 621] приложением L\_PROST. В данном интервале количество простых чисел равно 114. Сравним число 114 с *n/ln(n).*

Я получила 66. Это число говорит о том, что в нашем интервале существует примерно 66 простых чисел меньших чем 397.

*2. Повторить п.1 для интервала [m, n]. Сравнить полученные результаты с «ручными» вычислениями, используя «решето Эратосфена»*

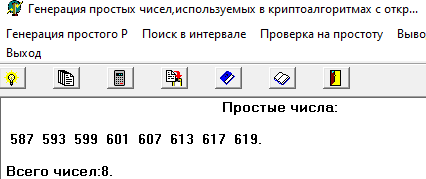


Рисунок 2 – Поиск простых чисел в интервале [587, 621]

На рисунке 1 представлен результат поиска простых чисел в интервале [587, 621] приложением L\_PROST. В данном интервале количество простых чисел равно 7.

Теперь используем “решето Эратосфена” для нахождения простых чисел в интервале [587, 621].

Воспользуемся свойством простых чисел (наименьший простой делитель составного числа n не превышает √*n*) и вычислим √397 ≈ 24,9, т. е. меньше 25. Запишем числа из заданного диапазона и удалим последовательно все числа, делящиеся на простые числа от 2 до 24. Такими простыми числами являются: 2, 3, 5, 7, 11, 13, 17,19, 23.

Шаг 1. Выпишем числа от 587 до 621: 587, 588, 589, 590, 591, 592, 593, 594, 595, 596, 597, 598, 599, 600, 601, 602, 603, 604, 605, 606, 607, 608, 609, 610, 611, 612, 613, 614, 615, 616, 617, 618, 619, 620, 621.

Шаг 2. Удалим из списка числа с учетом s=2: 587, 589, 591, 593, 595, 597, 599, 601, 603, 605, 607, 609, 611, 613, 615, 617, 619, 621.

Текущему s присваивается новое значение: s =3.

Шаг 3. Удалим из списка числа с учетом s=3: 587, 589, 593, 595, 599, 601, 605, 607, 611, 613, 617, 619.

Текущему s присваивается новое значение: s =5.

Шаг 4. Удалим из списка числа с учетом s=5: 587, 589, 593, 599, 601, 607, 611, 613, 617, 619. В этом списке первое число, большее, чем s=3, это 5. Текущему s присваивается новое значение: s =7.

Шаг 5. Удалим из списка числа с учетом s=7: 587, 589, 593, 599, 601, 607, 611, 613, 617, 619. В этом списке первое число, большее, чем s=3, это 5. Текущему s присваивается новое значение: s =11.

Шаг 6. Удалим из списка числа с учетом s=11: 587, 589, 593, 599, 601, 607, 611, 613, 617, 619. В этом списке первое число, большее, чем s=3, это 5. Текущему s присваивается новое значение: s =13.

Шаг 7. Удалим из списка числа с учетом s=13: 587, 589, 593, 599, 601, 607, 613, 617, 619. В этом списке первое число, большее, чем s=3, это 5. Текущему s присваивается новое значение: s =17.

Шаг 8. Удалим из списка числа с учетом s=17: 587, 589, 593, 599, 601, 607, 613, 617, 619. В этом списке первое число, большее, чем s=3, это 5. Текущему s присваивается новое значение: s =19.

Шаг 9. Удалим из списка числа с учетом s=19: 587, 593, 599, 601, 607, 613, 617, 619. В этом списке первое число, большее, чем s=3, это 5.

Шаг 10. Удалим из списка числа с учетом s=23: 587, 593, 599, 601, 607, 613, 617, 619. В этом списке первое число, большее, чем s=3, это 5.

Таким образом мы получили следующие простые числа: 587, 593, 599, 601, 607, 613, 617, 619.

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

Разложим число 621 на простые множители.

|  |  |
| --- | --- |
| 621 | 3 |
| 207 | 3 |
| 69 | 3 |
| 23 | 23 |
| 1 |  |

В результате мы получили следующее разложение 621 = 3 \* 3 \* 3 \* 23.

Так как число 587 простое, его имеет вид: 587 = 1 \* 587.

*4. Проверить, является ли число, состоящее из конкатенации цифр mǀǀn простым.*

В данном случае мне нужно проверить число 587621. Данное число является простым (проверено в приложении L\_PROST).

1. *Поиск НОД для двух и трех чисел*
2. *НОД для двух чисел*

Поиск НОД для двух чисел осуществляется по *алгоритму Евклида.* Суть этого алгоритма состоит из цепочки вычислений, по формуле

*a = b* · *q + r;*

где *a* и *b* числа для которых мы ищем НОД, где *a> b*; *q* – неполное частное деления *a* на *b*; *r* – остаток от деления *a* на *b*.

**Пример**

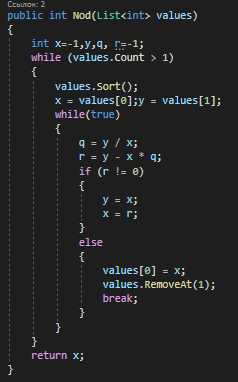
Пусть *a* = 15, *b* = 65. Найти НОД

65 = 15 · 4 + 5;

15 = 5 · 3 + 0;

Последний ненулевой остаток равен 6, поэтому НОД (65, 15) = 5

**Программа:**

****

1. *НОД для трех чисел*

НОД для трех чисел (*a, b, c*), считается так же по Алгоритму Евклида. Сначала мы считаем НОД для чисел (*a, b*) = *d*, далее считаем НОД (*c, d*). Если мы ищем НОД для четырех и более чисел, то продолжаем последнюю процедуру. Ищем НОД следующего числа при помощи самого числа и НОДа предыдущего.

**Пример**

Пусть *a* = 66, *b* = 24, c = 10. Найти НОД

66 = 24 · 2 + 18;

24 = 18 · 1 + 6;

18 = 6 · 3 + 0.

Последний ненулевой остаток равен 6, поэтому НОД (66, 24) = 6.

Используем НОД (66, 24) для поиска НОД с числом 10;

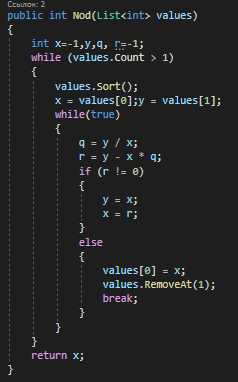
10 = 6 · 1 + 4;

6 = 4 · 1 + 2;

4 = 2 · 2 + 0;

Последний ненулевой остаток равен 2, поэтому НОД (10, 6) = 2. А это значит, что НОД (66, 24, 10) = 2;

**Программная реализация:**

****

1. *Поиск всех простых чисел на промежутке [2, n]*

Поиск простых чисел на промежутке [2, n] осуществляется про помощи *решета Эратосфена.*

Суть алгоритма состоит в том, что мы выписываем все числа из промежутка [2, n]. Далее, проходя по выписанным числам, берем самое меньшее число и делим его на все остальные числа. Если остаток от деления равен нулю, мы удаляем число из промежутка. После обхода всех чисел, мы выбираем следующее и продолжаем процедуру деления, до тех пор, пока не останутся простые числа.

**Пример**

Найти все простые числа до *n =* 20*;*

Выписываем все числа

2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20

Выбираем первое наименьшее число, оно всегда равно 2, и делим на него все остальные числа большие его самого. В итоге получаем следующий промежуток:

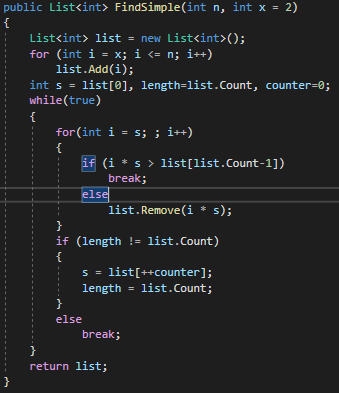
2, 3, 5, 7, 9, 11, 13, 15, 17, 19

Выбираем следующее число, оно равно 3. Делим на него все остальные числа большие его самого. В итоге получаем следующий промежуток:

2, 3, 5, 7, 11, 13, 17, 19

Продолжаем процедуру, пока на промежутке не останутся простые числа. В нашем случае, они остались сразу после деления всех оставшихся чисел на промежутке, на 3.

**Программная реализация:**

****

1. **Поиск простых чисел на заданном промежутке**

Поиск простых чисел на промежутке осуществляет, так же при помощи алгоритма *решето Эратосфена.*

**Пример**

Пусть надо найти простые числа на промежутке [30, 47]. Выпишем все числа на этом промежутке:

30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47

Далее нам нужно поделить все числа на простые числа. Чтобы их получить проведем следующие действия.

Найдем корень числа *n =* 47 и округлим его в меньшую сторону.

√50 = 6.85

Данное число – конец промежуток простых чисел, на которые мы будем делить числа из промежутка [29, 50].

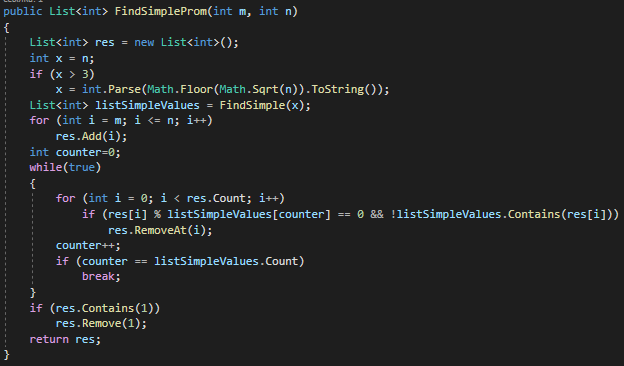
Выпишем числа этого промежутка и оставим в нем только простые числа при помощи того же алгоритма *решето Эратосфена*.

2, 3, 5,

Теперь будем делить каждое число из промежутка [29, 50], на данные простые числа. В итоге мы получим

31, 37, 41, 43, 47

**Программная реализация:**

****

**Вывод:** в результате выполнения лабораторной работы освоил алгоритмы поиска НОД для двух и трех чисел, а также поиск простых чисел до числа n и на заданном промежутке.