Ковкель Никита, ФИТ 3-4

Информационная безопасность

Отчёт по лабораторной работе № 3.

Основы теории чисел и их использование в криптографии

Вариант 6

В этой лабораторной работе задачей было создание функций для определения простых чисел и вычисления наибольшего общего делителя (НОД) для двух или трёх чисел. Простое число - это натуральное число больше 1, которое имеет только два делителя: 1 и само число. Для определения простоты числа мы использовали базовый алгоритм проверки. Единица не считается простым числом.

Реализация функции нахождения простых чисел представлена в листинге 1.

Листинг 1. Функция нахождения простых чисел.

|  |
| --- |
| def is\_prime(N: int) -> bool:      if N < 2: return False      for i in range(2, int(sqrt(N)) + 1):          if N % i == 0: return False      return True |

Для вывода простых чисел в консоль в диапазоне [a; b] была реализована соответствующая функция. Исходный код приведён в листинге 2.

Листинг 1. Функция нахождения простых чисел.

|  |
| --- |
| def print\_primes(a: int, b: int) -> None:      primes = [x for x in range(a, b + 1) if is\_prime(x)]      print(', '.join(map(str, primes)))      print(f'\nКоличество простых чисел в интервале {a, b} равно {len(primes)}') |

Результат работы функций приведен на рисунке 1.

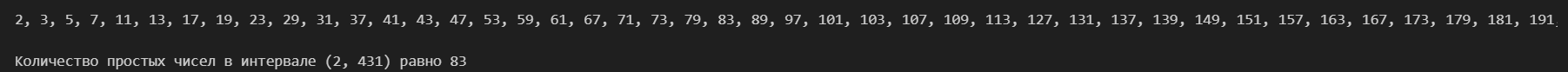


Рисунок 1. Результат работы функций для нахождения простых чисел в диапазоне [2; 431].

Алгоритм Евклида используется для нахождения наибольшего общего делителя (НОД) двух чисел. Если одно число делится на другое без остатка, то второе число и есть НОД. В противном случае, следуем следующим шагам:

Делим большее число на меньшее.

Делим меньшее число на полученный остаток.

Продолжаем деление остатков до тех пор, пока не получим нуль. Последний ненулевой остаток и будет НОД.

Рассмотрим другой пример. Пусть нам нужно найти НОД чисел 60 и 48.

60 : 48 = 1 (остаток 12)

48 : 12 = 4 (остаток 0)

Таким образом, НОД (60, 48) = 12. Это и есть основная суть алгоритма Евклида для нахождения НОД.

. Исходный функции для нахождения НОД двух чисел приведен в листинге 2.

Листинг 2. Исходный функции для нахождения НОД двух чисел.

|  |
| --- |
| def Euclid(a: int, b: int) -> int:      while b != 0:          a, b = b, a % b      return a |

Результат работы функции приведен на рисунке 2.



Рисунок 2. Результат работы функций для нахождения НОД на примере чисел 1178 и 31.

В результате этой лабораторной работы были созданы функции для определения простых чисел и вычисления наибольшего общего делителя. Кроме того, были изучены основные принципы теории чисел и их применение в области криптографии.