

## Large numbers

### Что случится после выполнения?

Поймёте, почему математика хорошо работает для использования обеспечения базовых свойств безопасности и зачем использовать ключи и секреты большого размера.

### Задача:

1. Используя одну из существующих библиотек (можете сделать самостоятельно, но лучше всё-таки использовать существующую для экономии времени), вывести количество вариантов ключей, которые можно задать 8-, 16-, 32-, 64-, 128-, 256-, 512-, 1024-, 2048-, 4096-битной последовательностью.
  - a. Пример: Если длина ключа равна **16** битам, то пространство ключей равно **65536**.
    - i. Пространство ключей – количество уникальных ключей, которые находятся в заданном диапазоне.
2. Для каждого из вариантов необходимо сгенерировать случайное значение ключа, которое принадлежит диапазону от 0x00...0 до 0xFF...F в зависимости от выбранной длины ключа.
3. Написать функцию для брутфорса значений из диапазона для нахождения ключа. Цель функции – перебирать значения ключа от 0x00...0 до тех пор, пока не будет найдено значение, равное предварительно сгенерированному ключевому. Функция должна выводить количество времени в миллисекундах, которое было затрачено на нахождение ключа.

Библиотеки, которыми есть смысл пользоваться:

<https://docs.oracle.com/javase/7/docs/api/java/math/BigInteger.html> – Java

<https://pkg.go.dev/math/big> – Go

<https://peps.python.org/pep-0237/> – Python (bignum type)

**Язык программирования – на ваш выбор. Желательно в "ReadMe.md" описать особенности реализации, если они есть.**

**Результат в виде ссылки на публичный репозиторий GitHub отправить в ответ на эту задачу в Classroom.**