

# Python. Семинар 1

Преподаватели: Дмитрий Косицин, Светлана Боярович и Анастасия Мицкевич

**Задание 0.** Установите и настройте Python.

**Общие замечания к заданиям.** Оценивается количество сделанных задач и их корректность. Эффективность, краткость и стиль будут поощряться. В свою очередь, полное отсутствие стиля и неэффективность – штрафовать. Использование функций, которые выполняют значительную часть задачи или реализуют ее полностью, будет также штрафовать, если это не оговорено в задаче отдельно.

**Задание 1.** Напишите программу, которая будет выводить все четные числа от 1 до 50, попутно заменяя числа, которые делятся на 3, на **Fizz**, делящиеся на 5 – на **Buzz**, а делящиеся и на 3, и на 5 – на **FizzBuzz**.

*Замечание.* Программу сохраните в файле `fizzbuzz.py`.

**Задание 2.** Реализуйте три простые функции: `set_bit` (устанавливает  $k$ -тый бит в переданном числе), `clear_bit` (сбрасывает  $k$ -тый бит) и `test_bit` (проверяет, установлен ли  $k$ -тый бит).

Первые две функции должны возвращать измененные числа, а функция `test_bit` – **True** или **False**. Значения всех аргументов предполагайте допустимыми.

*Замечания.* Функции сохраните в файле `bits.py` и назовите их именно так, как указано в задании. Все функции принимают непосредственно число для проверки/изменения и номер бита  $k$ .

**Задание 3.** Напишите функцию, возвращающую число, битовое представление которого является  $k$ -тым членом последовательности Морса–Тью (последовательность OEIS номер A010060, [https://en.wikipedia.org/wiki/Thue-Morse\\_sequence](https://en.wikipedia.org/wiki/Thue-Morse_sequence)). Значение  $k$  полагайте допустимым целым числом, большим либо равным нулю.

*Уточнение.* Ожидается, что для аргументов 0, 1, 2, 3 и т.д. функция будет возвращать `0b0`, `0b1`, `0b110`, `0b1101001` и т.д.

*Подсказка.* Возможно, полезной будет эта ссылка: <https://stackoverflow.com/a/12790495>.

*Замечания.* Программу сохраните в файле `thue_morse.py`, функцию назовите `get_sequence_item`. О быстрой скорости роста числа не беспокойтесь. Постарайтесь в решении обойтись без использования строк.

**Задание 4.** Билетик называется счастливым, если сумма первых трех цифр его номера равна сумме последних цифр. Напишите функцию, принимающую номер билета и возвращающую номер ближайшего счастливого билета (если их два – то любой из них). Номер переданного билета полагайте допустимым целым шестизначным числом.

*Замечания.* Программу сохраните в файле `ticket.py`, функцию назовите `get_nearest_lucky_ticket`. Постарайтесь в решении обойтись без использования строк.

**Задание 5.** Напишите функцию из не более чем двух строк, которая реализует стандартный алгоритм Евклида поиска НОД ([https://en.wikipedia.org/wiki/Greatest\\_common\\_divisor](https://en.wikipedia.org/wiki/Greatest_common_divisor)). Значения аргументов функции полагайте допустимыми целыми числами.

*Замечания.* Программу сохраните в файле `gcd.py`, функцию назовите `calculate_gcd`. Использование функции `gcd` из стандартной библиотеки не предполагается, что, впрочем, напрямую следует из условия задачи.