

Задание:

Есть функция `random2()` — которая выдает случайную величину 0 или 1 равновероятно. Напиши функцию `random3()` — которая будет выдавать равновероятно 0/1/2 используя функцию `random2()`

Использовать библиотеки или другие функции `random` нельзя. Используйте только функцию `random2()` и классический инструментарий `python`.

Решение:

Первое, что приходит в голову – сложить два результата функции `random2()`. Да, это переведёт множество $\{0; 1\}$ в множество $\{0; 1; 2\}$, но с точки зрения вероятностей ответ будет неверным:

	0	1
0	0	1
1	1	2

В сводной таблице указаны получаемые суммы. Легко увидеть, что вероятность получить единицу при таком подходе – 0.5, а нуля или двойки – по 0.25. Нас же интересуют вероятности по одной третьей для каждого числа.

Двумя вызовами `random2()` можно получить четыре различные упорядоченные комбинации нулей и единиц. Но можно не просто складывать их, а составить из полученной комбинации двоичное число

1-й разряд	2-й разряд	Десятичное число
0	0	0
0	1	1
1	0	2
1	1	3

Получение такого числа можно записать как $2 * \text{random2}() + \text{random2}()$ (мы переводим число из двоичной формы записи в десятичную)

Таким образом мы с равной вероятностью получаем 0, 1, 2 или 3. Но тройка нас не интересует. В случае, если мы получаем 3, можем начать сначала.

Посчитаем для такого алгоритма вероятность получить, например, ноль:

$$\begin{aligned} P(0) &= \frac{1}{4} + \frac{1}{4} \cdot \left(\frac{1}{4} + \frac{1}{4} \cdot \left(\frac{1}{4} + \dots \right) \right) = \\ &= \frac{1}{4} + \frac{1}{16} + \frac{1}{64} + \dots = \sum_{n=1}^{\infty} \left(\frac{1}{4} \right)^n = \\ &= \sum_{n=0}^{\infty} \left(\frac{1}{4} \right)^n - 1 = \frac{1}{1 - \frac{1}{4}} - 1 = \frac{4}{3} - 1 = \frac{1}{3} \end{aligned}$$

Для единицы и двойки вычисления абсолютно такие же. Минусом такого алгоритма является теоретическая возможность работать бесконечно долго (при должном «везении» на тройки).

Ответ – в файле `problem_solution.py`, проверка – в файле `problem_check.py`