

Python. Семинар 3

Преподаватели: Дмитрий Косинин, Светлана Боярович и Анастасия Мицкевич

Задание 1. (0.4 балла). Напишите функцию, которая для выборки, заданной списком, и заданного натурального числа k посчитает и выведет гистограмму распределения шириной k . Другими словами, найдет максимум и минимум, поделит интервал на k частей и посчитает, сколько элементов выборки попало в каждый интервал.

Верните список длины k , содержащий количество элементов в каждой ячейке. Левую границу интервалов считайте включая, правую – исключая (кроме последнего интервала).

Дополнительно (на бонус 0.2 балла). Постарайтесь также наглядно вывести результаты, используя псевдографику: расположите гистограмму горизонтально и используйте паддинги при форматировании строк.

Замечание. Предполагаемая сложность – $O(n^2)$, но желательно $O(n)$ (задача подобная уже была), где n – длина входного списка, а $1 \leq k \leq n$.

Функцию назовите `distribute`. Программу сохраните в файле `hist.py`.

Задание 2. (0.7 балла). Напишите функцию, которая первым аргументом на вход принимает строку с цифрами некоторого целого числа. Второй аргумент функции – числа, вхождения которых нужно найти в переданной строке цифр – является либо целым числом (`int`), либо кортежем целых чисел (если их несколько).

Функция должна возвращать количество всех позиций, на которых обнаружены вхождения, а также отсортированный список обнаруженных позиций. Если передан кортеж, то возвращать нужно суммарное количество вхождений, а также объединенный отсортированный список позиций найденных чисел. Если длина списка больше чем k , то возвращайте только первые k элементов. Параметр k передается третьим аргументом – натуральное число, по умолчанию 5.

Замечание. Индексацию цифр ведите с единицы, т.е. первая цифра некоторого числа имеет индекс *один*, а не *ноль*.

Пример. Для `index('123', 1)` ответом будет `(1, [1])`.

Пример. Для случая `index('121212222', (1, 2, 12), 3)` ответом будет `(13, [1, 1, 2])`.

Функцию назовите `index`. Программу сохраните в файле `big_number.py`.

Задание 3. (0.9 балла). Существует исследование, говорящее о том, что в словах текста можно произвольно переставить буквы (не затрагивая первую и последнюю), и от этого читабельность текста практически не ухудшится. Напишите программу для проверки этого факта.

Ваша программа должна получать на вход какой-нибудь текст и переставлять буквы в его словах случайным образом (см. модуль **random**, постарайтесь выбрать самый подходящий метод).

Также у программы должен быть режим, в котором буквы (кроме первой и последней) переставляются не случайным образом, а сортируются по алфавиту. Сравните результаты (для себя).

Функция должна принимать на вход текст (одна строка, возможно с переносами строк внутри) и параметр, обозначающий, нужно ли использовать случайную перестановку или нет.

Используйте `ArgParser` для передачи строки и указания режима. Предусмотрите возможность передачи пути к файлу. Название параметров и их вид выберите произвольно (будет проверяться вручную).

Утилиту постарайтесь реализовать одновременно как скрипт и как модуль, то есть чтобы была возможность запускать ее как скрипт с удобным интерфейсом командной строки, а также возможность подключать (импортировать) ее как внешний модуль.

Замечание. Постарайтесь организовать работу с кириллицей (`unicode`; в Python 2.7 используйте `encode/decode` методы у строк).

Замечание. Словами считайте последовательности букв, цифр и `dash '-'`. За использование регулярных выражений (изучите самостоятельно, <https://docs.python.org/2.7/library/re.html>) будет бонус (предварительно, 0.3 балла). Если не будете использовать их, то разделителями считайте только переносы строк и пробелы.

Замечание. Предполагаемая сложность решения – $O(n^2)$, но лучше $O(n \ln n)$, где n – длина входных данных.

Саму функцию назовите `shuffle`. Программу сохраните в файле `british.py`.

Задание 4. (0.2 балла). Дополнительно (на бонус).

Напишите функцию, которая при каждом вызове возвращает количество раз, которое она была вызвана. Использовать глобальные переменные не допускается, иначе говоря, в файле должно быть только определение функции (с ключевым словом *def*).

Функцию назовите `smart_function`. Программу сохраните в файле `smart.py`.