

№1 Базовые элементы языка Python

Формат сдачи: результат выполнения каждого задания – отдельный файл с расширением *.py. Каждый файл с выполненным заданием (или все сразу, или часть) необходимо выслать на ящик **a.kozak.edu@gmail.com** в теме письма необходимо указать номер ЛР и ФИО студента, при этом файл с исходным кодом должен иметь следующие название [номер_студента_в_общем_списке_группы]_lab_[номер_ЛР]_[номер_задания].py (например, 12_lab_1_3.py) (невыполнение этих простых правил влечет аннулирование баллов за задание). Во время защиты необходимо продемонстрировать работу вашей программы, а также ответить на вопросы преподавателя. Максимальный балл за задание ставится в случае если студент правильно выполнил задание, его код соответствует PEP-8, стиль кода соответствует pythonic стилю, задание выполнено наиболее оптимальным образом и студент прошел успешно защиту на паре.

Обращаю внимание на то, что организация ввода-вывода программы должна быть интуитивно понятна пользователю!

Что такое JSON? – читает -> <https://ru.wikipedia.org/wiki/JSON>

Срок сдачи: до 01.04.2019 (может меняться)

Задание 1. (15 баллов). Числа Фибоначчи

Напишите функцию, которая будет получать на вход число n и возвращать n -ое число Фибоначчи. Считаем, что нулевое и первое числа Фибоначчи равны 0 и 1 соответственно.

Задание 2. (20 баллов). Сочетания

Задана строка, состоящая только из слов латинского алфавита, разделенных пробелами. Нужно привести все слова к нижнему регистру и из всех таких уникальных слов сгенерировать всевозможные сочетания без повторов четной длины. (В комбинаторике сочетанием из n по k называется набор k элементов, выбранных из данного множества, содержащего n различных элементов. Наборы, отличающиеся только порядком следования элементов (но не составом), считаются одинаковыми.)

Задание 3. (20 баллов). Разложения на простые множества

Каждое составное число можно единственным способом представить в виде произведения простых чисел. Например: $20 = 2 \cdot 2 \cdot 5 = 2^2 \cdot 5$. Реализуйте алгоритм разложения числа на простые множители (если число простое, то выводится оно само). (по возможности организовать ввод как с клавиатуры, так и с помощью аргументов командной строки)

Задание 4. (30 баллов). Object to JSON

Реализовать функцию `to_json(obj)`, которая на вход получает python-объект, а на выходе у неё строка в формате JSON. Если при вызове `to_json` был передан объект (или объект, содержащий) тип, которого не может быть преобразован в JSON, функция выбрасывает исключение `ValueError`.

Задание 5. (40 баллов). Merge it

Вам нужно написать функцию-генератор `merge_sort`, которая принимает на вход любой массив/список и выдает все промежуточные списки, которые получаются в процедуре MergeSort от самого нижнего уровня до отсортированного массива в конце, т.е. сначала нужно выдать список, в котором отсортирована каждая пара, затем список, в котором

отсортирована каждая четверка, и так далее, пока не дойдем до полностью отсортированного массива. Гарантируется, что список содержит количество элементов, являющееся степенью двойки. (по возможности организовать вывод как в консоль, так и в файл)

Задание 6. (25 баллов). Генерируем мусор :)

Реализовать программу генерирующую текстовый файл. На вход программы поступает два (три) числа – N , K , L ; где N – количество строк в файле, K – количество слов в строке (слова разделены пробелами) и L – длина каждого слова, если не задана пользователем, то для каждого нового слова – является случайным значением от 1 до 15. При запуске программы использовать аргументы командной строки.

Задание 7. (25 баллов). Собираем статистику по тексту

На вход поступают текстовые данные (использовать ввод через файл). Необходимо посчитать и вывести:

1. сколько раз повторяется каждое слово в указанном тексте
2. среднее количество слов в предложении
3. медианное количество слов в предложении
4. top- K самых часто повторяющихся буквенных N -грам (K и N имеют значения по умолчанию 10 и 4, но должна быть возможность задавать их с клавиатуры)

При решении использовать контейнер `dict()` или его аналоги и встроенные операции над строками. Предусмотреть обработку знаков препинания. При тестировании использовать генератор текстового файла из Задания 6. (ввод данных организовать как с клавиатуры, так и с помощью аргументов командной строки)

Задание 8. (25 баллов). Хранилище

При запуске программа работает в интерактивном режиме и поддерживает команды:

1. `add <key> [<key> ...]` - добавить один или более элементов в хранилище (если уже содержится, то не добавлять).
2. `remove <key>` - удалить элемент из хранилища.
3. `find <key> [<key> ...]` - проверить наличие одного или более элементов в хранилище, вывести найденные.
4. `list` - вывести все элементы в хранилище.
5. `grep <regex>` - поиск значения по регулярному выражению.
6. `save` и `load` - сохранить хранилище в файл и загрузить хранилище из файла

При решении использовать контейнер `set()`.

Задание 9. (30 баллов). Flatten

Вам нужно написать функцию-генератор `flatten_it`, которая принимает на вход любой итерируемый аргумент и линеаризует его, т.е. нужно рекурсивно проходить по элементам из этого итерируемого объекта, которые также являются итерируемыми объектами, и

так далее до произвольной глубины вложенности, пока не дойдем до неитерируемых "листьев". Ваша функция в итоге должна являться итератором по всем "листьям". Обратите внимание, что возможны циклы по вложенности, вы должны уметь обрабатывать такие ситуации и бросать исключение `ValueError` в этом случае. (Пример – на входе список `[1, 2, [3, 4, 5], [6, [7, 8]]]`, на выходе `[1, 2, 3, 4, 5, 6, 7, 8]`)

Задание 10. (40 баллов). JSON to Object

Реализовать функцию `from_json(text)`, которая возвращает python-объект соответствующий json-строке. Не использовать стандартные инструменты работы с JSON.