**Задания для контрольной точки 2**

**Задание 1**. Обработка текста.

На входе – текстовый файл с некоторым текстом на любом языке. Текст либо поместить в файл заранее, в этом случае его нужно будет прислать вместе с исходными кодами, либо при запуске программы получить откуда угодно любым методом – например, "спарсить" веб-страницу или сгенерировать случайным образом ("Hjekjl sdy6uyeru gash, heruityui!" – тоже как бы "текст"). Во втором случае необходимый инструментарий изучить самостоятельно.

1) Выполнить количественный анализ текста:

* подсчитать общее количество слов (знаки препинания словами не считаются);
* подсчитать количество предложений;
* подсчитать количество слов в каждом предложении;
* подсчитать количество букв в каждом слове;
* подсчитать количество каждого знака препинания.

Результат представить в виде следующей структуры:

словарь, состоящий из элементов:

строка "Всего слов": <количество слов>,

строка "Всего предложений": <количество предложений>,

строка "Предложения": список из кортежей из 2-х элементов вида (само предложение, <количество слов>),

строка "Слова": словарь из элементов {слово: <количество букв>},

строка "Знаки препинания": словарь из элементов {знак препинания: <количество таких знаков в тексте>}.

2) Сохранить полученный объект в двоичный файл с помощью модуля pickle, затем загрузить из файла.

3) Вывести рассчитанную статистику на экран в удобоваримом виде (не просто print(<словарь>), а с использованием заголовков, перечислением элементов, и т.д., например:

Всего слов: столько-то

Всего предложений: столько-то

Предложения:

Предложение 1. Столько-то слов

Предложение 2. Столько-то слов

).

4) Разбить текст на абзацы по *n* предложений (*n* вводится с клавиатуры, должно быть натуральным числом). Если *n* = 1, каждый абзац должен состоять из 1 предложения. Если *n* заведомо очень большое и превышает количество предложений в тексте (например, 99999), либо просить ввести другое *n*, либо поместить все предложения в один абзац.

5) Абзацы отсортировать по возрастанию количества слов в них.

6) Сохранить полученный текст в текстовый файл.

**Обязательно** использование функций *map*(), *sorted*() (или метода *sort*()), лямбда-функций.

*Пример*: текст

Захотелось мне как-то сделать более надёжной передачу информации через радиоканал. Это не какой-то промышленный проект, или что-то другое серьёзное. Это скорее для хобби и саморазвития.

Результирующая структура:

{"Всего слов": 25,

"Всего предложений": 3,

"Предложения": [("Захотелось мне как-то сделать более надёжной передачу информации через радиоканал.", 10),

("Это не какой-то промышленный проект, или что-то другое серьёзное.", 9),

("Это скорее для хобби и саморазвития.", 6)],

"Слова": {"Захотелось": 10,

"мне": 3,

"как-то": 5,

"сделать": 7,

…},

"Знаки препинания": {".": 3,

",": 1,

"-": 3}}

Текст после разбиения на абзацы по 2 предложения и сортировки их по числу слов:

Это скорее для хобби и саморазвития.

Захотелось мне как-то сделать более надёжной передачу информации через радиоканал. Это не какой-то промышленный проект, или что-то другое серьёзное.

Программа будет оцениваться на примере предоставленного текста, однако возможность возникновения исключительных ситуаций необходимо предусмотреть (например, отсутствие входного файла, пустой файл или двоичный файл вместо текстового на входе).

**Задание 2**. Расчет финансовых показателей инвестиционного проекта.

Придумать инвестиционный проект. Прямо в комментариях коротко (в двух-трех предложениях) описать его идею, а также указать основные финансовые параметры (первоначальные затраты, период и срок реализации проекта – например, год и 5 лет, сумма поступлений за период, сумма расходов за период, ставка дисконтирования).

С помощью функций из модулей *itertools*, *functools* вычислить основные показатели проекта: NPV, дисконтированный срок окупаемости, IRR (в процентах). Сделать вывод об эффективности данного инвестпроекта (он совершенно не обязательно должен быть эффективен, что уж получится), вывести результаты на экран.

*Пояснение к заданию 2*:

У нас есть входные данные:

- первоначальные затраты (ПЗ)

- регулярный доход за период (РД)

- регулярные затраты за период (РЗ)

- срок реализации проекта (количество периодов) (n)

- ставка дисконтирования, например, 15% (в долях 0,15) (r)

Надо вычислить 3 показателя:

1)

Желательно подобрать такие входные данные, чтобы NPV оказался больше нуля.

2) Срок окупаемости проекта DPP.

Это такое количество периодов, за которое проект полностью окупает свои затраты, т.е. NPV за этот срок равен нулю.



Чтобы его вычислить:

формируем 2 массива: NPV проекта за все периоды и NPV нарастающим итогом.

Первый массив: NPV0, NPV1, NPV2, …, NPVn-1, NPVn

NPV0 = –ПЗ, NPVi =

Второй массив: NPV0, NPV0 + NPV1, NPV0 + NPV1 + NPV2, …, итоговый NPV

Чтобы получить второй массив, надо применить функцию itertools.accumulate().

Теперь DPP можно вычислить так:

\* Находим номер периода m, в котором последний отрицательный NPV нарастающим итогом (по массиву 2). Например, в периоде 3 NPV еще с минусом, а в 4-ом уже с плюсом. Значит, m = 3.

\* Находим NPVm+1 по массиву 1.

\* Находим NPV за m периодов по массиву 2, т.е. тот самый последний отрицательный накопленный NPV.

DPP =

В числителе должно быть положительное число, поэтому берем модуль.

3) Внутренняя норма доходности IRR. Это такая ставка дисконтирования r, при котором NPV проекта за весь срок реализации n (то, что считали в пункте 1) равен 0.

Как ее посчитать? Самый простой способ – перебрать разные варианты r и найти такой, при котором NPV будет по модулю наиболее близок к нулю, т.е. найти минимум |NPV| и запомнить, какая при этом будет ставка r.

*Указания к выполнению работы*:

1. Не должно быть необработанных исключений (ошибок выполнения). При их обнаружении во время проверки оценка за контрольную точку сразу снижается на 2 балла.
2. Программа должна иметь понятный, интуитивный пользовательский интерфейс.
3. Если задание непонятно, есть какие-то вопросы, следует либо уточнить у преподавателя, либо описать в комментариях свое затруднение и использованное решение.
4. Текстовые файлы с исходными кодами (модули с расширением .py, ***каждое задание в отдельном модуле***) высылать на почту преподавателя ([rvgarafutdinov@gmail.com](mailto:rvgarafutdinov@gmail.com)) с указанием группы, ФИО и номера контрольной точки. Никакие архивы папок с проектами PyCharm и т.д. отправлять не надо, только текстовый файл с программой.
5. Текст в задании 1 взять свой собственный, не из примера!