Задания по курсу Python
Задание 2

Д.В. Иртегов

5 марта 2018 г.

Задачи необходимо сдать до 24 марта. Решения необходимо сдавать путем отправки pull request в каталог problems-2 репозитория
https://github.com/dmitry-irtegov/NSUPython2018.
Датой сдачи задания считается дата отправки первого pull request. Если запрос не принят из-за моих замечаний, у вас есть неделя на их исправление.

Если запрос принят, задание считается засчитанным. Если запрос не принят, в комментарии вы можете узнать, почему.

В одном запросе следует отправлять не более одного решения. Если решение состоит из нескольких файлов, в запрос должны быть включены они все. Все запросы одного студента должны отправляться в каталог с именем, соответствующим его учетной записи. Например, для задачи 3 из группы задач 2, сдаваемой студентом v-pupkin, рекомендуемое имя файла `problems-2/v-pupkin/task3.py`.

Задача 1. Напишите скрипт, который требует ввода числа из stdin (стандартного потока ввода). Если введенная строка не является числом, скрипт должен требовать повторить ввод. Скрипт завершается, когда введено число, или когда введен конец файла (CTRL-Z в Windows, CTRL-D в начале строки в Unix).

Задача 2. Реализуйте класс Vector, соответствующий N-мерному вектору линейной алгебры. У этого класса должны быть определены все естественные для вектора операции — сложение, вычитание, умножение на константу, скалярное произведение и сравнение на равенство, — а также операции вычисления длины, получение элемента по индексу и строковое представление. Во всех операциях можно считать, что все передаваемые аргументы корректны.

Задача 3. Создайте документацию для класса Vector и его методов из задания 2. В Python нет закрепленного общего стиля для документации, но можно выбрать какой-нибудь популярный и использовать его. Например, можно взять стиль для SciPy https://github.com/numpy/numpy/blob/master/doc/HOWTO_DOCUMENT.rst.txt
(для начала его можно не читать полностью, а только посмотреть примеры для классов и функций).

Задача 4. Опхряб1: Напишите программу, которая определяет частоту встречаемости байтов в тексте. Программе через аргумент командной строки подается имя файла. Файл содержит текст в неизвестной кодировке. Постройте диаграммы частот встречаемости всех не-ASCII символов (байтов, значение которых превосходит 127) в файле. Программа должна вывести частоты встречаемости каждого байта, в порядке убывания встречаемости (самый часто встречающийся байт идет первым). Встречаемость определяется по формуле $p[b] * 100/N$, где $p[b]$ – количество раз, которые байт встретился в тексте, а N – количество байт с кодами >127 в тексте. Для чтения байтов из файла его нужно открывать с параметрами `open(filename, 'rb')`
Замечание 1: Элемент байтового массива имеет тип int.
Замечание 2: Для проверки правдоподобия значений, вы можете попытаться сконвертировать одиночный байт в символ русского языка, используя код:
`char=bytes([b]).decode('koi8_r')`
где `b` – значение байта, а `'koi8_r'` – название кодировки.
Разумеется, это возможно только если вы знаете кодировку файла.
Таблица встречаемости символов русского языка доступна по адресу
https://www.dpva.ru/Guide/GuideUnitsAlphabets/Alphabets/FrequencyRuLetters/
Замечание 3: Неточное совпадение значений и даже нарушения порядка символов по встречаемости *не* свидетельствуют об ошибке в вашей программе, а только о том, что вам попался неудачный текст. Также, обратите внимание, что подсчет частоты байтов считает встречаемости для строчных и заглввных символов раздельно, а корректно перевести строчные символы в заглавные невозможно без знания кодировки. В рамках подсчета частот байтов эта проблема неразрешима.
Замечание 4: Для генерации тестовых данных, в Linux можно использовать утилиту iconv, а в Windows — сохранение из редактора Far Manager с переключением кодировки.

Задача 5. Опхряб2: На основе решения задачи 4 и частот встречаемости символов русского языка
https://www.dpva.ru/Guide/GuideUnitsAlphabets/Alphabets/FrequencyRuLetters/
напишите программу, которая подбирает наилучшую возможную кодировку для просмотра содержимого файла, среди всех однобайтовых кодировок русского языка, поддерживаемых Python3. Обратите внимание, что частоты символов в реальных текстах небольшой длины не совпадают с частотами в таблице, поэтому нужно реализовать нечеткое сравнение.

Для вывода файла в подобранной кодировке рекомендуется его открыть заново как текстовый файл с использованием пакета codecs.