

Задания по курсу

Замечание: Если в условии задания присутствует какая-либо неточность, трактуйте ее в свою пользу. Например, если не сказано, корректен ввод данных или нет, можете считать, что он всегда корректен.

Номер: 1 балл: 2 дедлайн: 05.10.2016

Напишите функцию, которая получает на вход последовательность чисел и возвращает последовательность ее кумулятивных сумм:

$$s_i = \sum_0^i a_i$$

Например, для последовательности [1, 2, 3] она должна вернуть [0, 1, 3, 6]

Номер: 2 балл: 2 дедлайн: 05.10.2016

Напишите функцию, которая получает на вход последовательность чисел, а также нижнюю границу a и верхнюю границу b и “обрезает” все числа в соответствии с этим диапазоном. То есть все числа, меньше a , должны быть заменены на a , а все числа, большие b , - на b .

Номер: 3 балл: 2 дедлайн: 05.10.2016

Напишите программу, которая получает из консоли число и проверяет, что для этого числа верна гипотеза Коллатца. Эта гипотеза состоит в следующем. Возьмем произвольное число n . Если оно четно, заменим его на $n/2$, если нет, то заменим его на $3 * n + 1$. К полученному числу применим те же действия, и так далее, пока не получим число 1. Например, для числа 3 получим такую цепочку преобразований:

$$3 \rightarrow 10 \rightarrow 5 \rightarrow 16 \rightarrow 8 \rightarrow 4 \rightarrow 2 \rightarrow 1$$

Программа так же должна выводить цепочку преобразований.

Номер: 4 балл: 4 дедлайн: 12.10.2016

Необходимо реализовать функцию, которая принимает на вход строку состоящую из нижних подчеркиваний и букв, и инвертирует порядок следования слов (слово = подряд идущая последовательность букв), не изменяя при этом количество нижний подчеркиваний.

“ ____The____Day_That_Never_Comes____”

“ ____Comes____Never_That_Day_The____”

Номер: 5 балл: 3 дедлайн: 12.10.2016

Напишите программу, которая выводит на экран текст песни “Ten Green Bottles”

Ten green bottles hanging on the wall,
Ten green bottles hanging on the wall,
And if one green bottle should accidentally fall,
There'll be nine green bottles hanging on the wall.
Nine green bottles hanging on the wall,

Nine green bottles hanging on the wall,
And if one green bottle should accidentally fall,
There'll be eight green bottles hanging on the wall.

...

One green bottle hanging on the wall,
One green bottle hanging on the wall,
If that one green bottle should accidentally fall
There'll be no green bottles hanging on the wall.

Постарайтесь написать программу так, чтобы в ней было как можно меньше “сырого текста”.

Номер: 6 балл: 3 дедлайн: 12.10.2016

Напишите функцию, разлагающую данное число на простые множители. Результатом работы программы должен быть список, в котором каждому простому сомножителю p и его степени k соответствует пара (p, k) , являющаяся вложенным списком. Например, число 12 должно быть разложено так: $[[2, 2], [3, 1]]$

Номер: 7 балл: 3 дедлайн: 26.10.2016

Напишите списковое выражение, которое для данного n генерирует все пифагоровские тройки из чисел не больше чем n . (x, y, z называется пифагоровской тройкой, если $x^2 + y^2 = z^2$)

Номер: 8 балл: 3 дедлайн: 26.10.2016

Напишите списковое выражение, генерирующее все простые числа не больше заданного числа.

Номер: 9 балл: 5 дедлайн: 26.10.2016

Дано некоторое множество объектов X и натуральное число n . Необходимо перебрать все элементы множества X^n (декартово произведение). Например, если $X = \{1, a\}$, то X^2 содержит такие элементы: $(1, 1), (1, a), (a, 1), (a, a)$.

Для этого напишите класс, экземпляры которого соответствуют элементам этого множества, и у которых есть методы для получения текущего элемента, и замены текущего элемента на «следующий». Следующий элемент выбирайте таким образом, что бы он был лексикографически минимальным среди больших. Если элемент уже является лексикографически максимальным, то возвращаем минимальный элемент, тем самым производя заикливание.

Для упрощения считайте, что множество X задается списком, все элементы которого уникальны. Элементы множества – цифры и буквы. Считать, что любое число меньше любой буквы.

Номер: 10 балл: 3 дедлайн: 26.10.2016

Напишите функцию, которая на вход принимает некоторый объект, который состоит из, возможно вложенных, списков, словарей, чисел и строк, и печатает их на экран в «удобном» виде. Это значит, что если объект A вложен в объект B , то при печати A нужно использовать больше отступов, чем при печати B .

```
format_print([1, {'a': [1, 2], 'b': 'hi'}])
```

```
[
  1,
  {
    a: [
      1,
      2,
    ],
  },
]
```

```
    b : hi,  
    },  
]
```

Указание. Можно сделать, что бы функция не сразу выводила результат на экран, а возвращала строку для печати. Тогда генерировать строки для вложенных контейнеров можно будет рекурсивно.

Номер: 11 балл: 5 дедлайн: 26.10.2016

Реализуйте класс *N* — мерный вектор. У этого класса должны быть определены все естественные для вектора операции — сложение, вычитание, умножение на константу и скалярное произведение, сравнение на равенство. Так же операции вычисления длины, получения элемента по индексу, преобразование в строку. Во всех операциях можно считать, что все передаваемые аргументы корректны.

Номер: 12 балл: 5 дедлайн: 26.10.2016

Тщательно реализуйте программу как с точки зрения модуля (выделить весь содержательный код в удобные функции), так и с точки зрения скрипта (сделать удобный интерфейс, написать описание, убедиться, что программа устойчива к некорректному вводу). Так же нужно самостоятельно разобраться, как использовать параметр `required` у функции `add_argument` и, при необходимости, использовать его в программе.

Напишите аналог Unix-утилиты `cut`. Эта утилита работает следующим образом.

Будем считать, что входной файл представляет из себя таблицу, в которой строки отделены переносом строки `\n`, а столбцы — специальным разделительным символом или последовательностью символов (например, `\t` или `;`). Тогда можно сделать срезы этой таблицы по столбцам. Утилита `cut` получает на вход список столбцов (или полей, `fields`) и делает из них новую таблицу из заданных столбцов. При этом у столбцов может поменяться порядок, и некоторые столбцы могут повториться.

Например, вызов `$ cut -f 3,1,5` возьмет из таблицы третий, первый и пятый столбцы и поставит их в этом порядке.

Сделайте так, что бы в вашей программе была возможность изменять разделитель столбцов, но по умолчанию бы использовался символ табуляции `\t`.

Список полей должен задаваться с помощью ключа `-f (--fields)`, разделитель с помощью ключа `-s (--separator)`, входной файл как аргумент без ключа.

Если число столбцов в какой-то строке меньше требуемого, то это считать ошибкой во входных данных, и программа должна сообщать об этом и завершать работу.

Замечание. Можете выбрать любой способ нумерации столбцов (с нуля или с единицы)

Номер: 13 балл: 20 дедлайн: xx.xx.2016

Todo

Номер: 14 балл: 5 дедлайн: 30.11.2016

Известно, что объект `None` является синглтоном, то есть он всегда существует в единственном экземпляре (и все переменные с этим значением ссылаются на один и тот же объект). Но технически любой неизменяемый объект тоже может быть синглтоном, и это не будет заметно сточки зрения кода (объясните, почему?). Проверьте (без обращения к документации), какие объекты в вашей версии Python реализованы таким образом. Предложите объяснение, зачем это нужно.

Номер: 15 балл: 5 дедлайн: 30.11.2016

Изучите, как реализуется менеджер контекста.

https://lancelote.gitbooks.io/intermediate-python/content/book/context_managers.html

Напишите менеджер контекста, который позволит засекать время выполнения блока кода с помощью конструкции **with** и выводить это время на экран по выходу из блока.

Пример использования:

with Timer():

```
do_something()
```

Номер: 16 балл: 4 дедлайн: 21.12.2016

Реализуйте функцию `xrange`. Функция должна полностью соответствовать по интерфейсу встроенной функции `xrange`, но при этом допускается, что бы ее результат был итерируемым.

Номер: 17 балл: 5 дедлайн: 21.12.2016

Предположим, дана некоторая последовательность объектов, например:

```
["hi", "qwerty", "qwerty", [1, 2], 5, 5, 5, 1, 5, 5]
```

Эту последовательность можно сжать:

```
[1, "hi", 2, "qwerty", 1, [1, 2], 3, 5, 1, 1, 2, 5]
```

Для этого каждую последовательность из одинаковых элементов заменяем на два элемента: длину последовательности и ее представитель.

Задача состоит в том, что бы написать две функции-генератора, одна из которых может сжимать любую последовательность по предложенному алгоритму, а вторая разжимать. Корректно учесть, что 2-я функция на вход не может принимать любую последовательность.

Номер: 18 балл: 5 дедлайн: 21.12.2016

Реализуйте класс «последовательность с фильтрацией», который соответствует некоторой последовательности объектов и имеет следующие методы:

- Создать объект на основе произвольного итерируемого объекта.
- Проитерироваться по элементам (`__iter__`).
- Отфильтровать последовательность с помощью некоторой функции и вернуть новую последовательность, в которой присутствуют только элементы, для которых эта функция вернула `True`.

Будем считать, что в данной задаче основным приоритетом является экономия памяти. Поэтому количество копирований данных нужно свести к минимуму (возможно, пожертвовав скоростью работы).

Номер: 19 балл: 8 дедлайн: 21.12.2016

Напишите функцию, которая принимает на вход текст `html` страницы, и возвращает все ссылки, которые присутствуют на ней.

При написании кода используйте библиотеку

Номер: 20 балл: 15 дедлайн: 21.12.2016

Существует шуточный «закон Философии» для статей Википедии: если переходить по первой нормальной ссылке в статье, то рано или поздно мы придем на статью о философии. Ваша задача заключается в том, что бы проверить этот закон.

Для этого реализуйте программу, которая получает на вход ссылку на статью Википедии, а потом циклически переходит по первой нормальной ссылке и повторяет эту операцию до тех пор, пока не достигнет статьи про философию, либо не заикнется по ссылкам. Нормальной ссылкой будем называть ссылку, которая находится в основном содержании статьи, не в инфобоксах, не в служебных блоках, написана синим цветом (красный соответствует несуществующей статье), не курсивом, не является сноской и не находится в скобках. Обратите внимание, что для проверки нормальности не обязательно разбирать таблицы стилей и проверять цвет и т.п., достаточно сделать, что бы программа работала для текущей версии Википедии (например, использовать атрибут class у тегов).

Для удобства проверки сделайте, что бы последовательность переходов выводилась на экран (выводим номер в цепочке и название статей)

Замечание: Чтобы не создавать большую нагрузку на сервер Википедии, сделайте так, что бы ваша программа делала не более 2-х запросов в секунду. Для этого воспользуйтесь функцией sleep из модуля time.

Примечание: Для демонстрации работы вашей программы к моменту сдачи подготовьте лог работы программы, начиная с 5 статей, одна из которых <https://ru.wikipedia.org/wiki/Python>. Для нее первый переход будет на ссылку “высокоуровневый язык программирования”.

При написании кода рекомендуется использовать:

<http://docs.python-requests.org/en/master/>

<https://www.crummy.com/software/BeautifulSoup/bs4/doc/>

Номер: 21 балл: 5 дедлайн: 21.12.2016