(*^*) Дока/Python/

homework-1

Авторы: <u>BernarBerdikul</u>

Редактировать на 🖸 GitHub

```
Время чтения: 5 мин
Кратко
Задача №1.
```

Задача №2. Задача №3. Задача №4. Задача №5.

Кратко

Задачи после первой лекций. Требования к задачам:

• Должны проходить все тесты; • Написаны без использования сторонних библиотек (можно внутренние, к примеру itertools); • Код соответствует РЕР8.

Задача №1. &

Haписать метод domain_name, который вернет домен из url адреса: SHELL

1 url = "http://github.com/carbonfive/raygun" -> domain name = "github" 2 url = "http://www.zombie-bites.com" -> domain name = "zombie-b: -> domain name = "cnet" 3 url = "https://www.cnet.com" Скопировать

PYTHON

Основа: 1 def domain_name(url):

Для проверки:

Скопировать

return

PYTHON 1 assert domain_name("http://google.com") == "google" 2 assert domain_name("http://google.co.jp") == "google" 3 assert domain_name("www.xakep.ru") == "xakep" 4 assert domain_name("https://youtube.com") == "youtube" Скопировать

Задача №2.

Написать метод int32_to_ip, который принимает на вход 32-битное целое число (integer) и возвращает строковое представление его в виде IPv4-адреса:

SHELL 1 2149583361 -> "128.32.10.1" 2 32 -> "0.0.0.32"

Основа: PYTHON 1 def int32_to_ip(int32): 2 return

Скопировать

Скопировать

SHELL

Скопировать

Для проверки: PYTHON

2 assert int32_to_ip(0) == "0.0.0.0"

Задача №3. Haписать метод zeros, который принимает на вход целое число (integer) и

.. * N) заданного числа:

Будьте осторожны 1000! имеет 2568 цифр. Доп. инфо: http://mathworld.wolfram.com/Factorial.html

возвращает количество конечных нулей в факториале (N! = 1 * 2 * 3 *.

1 zeros(6) = 12 # 6! = 1 * 2 * 3 * 4 * 5 * 6 = 720 --> 1 trailing zero

1 assert int32_to_ip(2154959208) == "128.114.17.104"

3 assert int32_to_ip(2149583361) == "128.32.10.1"

4 zeros(12) = 25 # 12! = 479001600 --> 2 trailing zeros Скопировать Основа: PYTHON

Для проверки: PYTHON

возвращает количество слов «banana» в строке.

Подсказка: вы не должны вычислять факториал.

Найдите другой способ найти количество нулей.

2 assert zeros(6) == 1 3 assert zeros(30) == 7Скопировать

1 assert zeros(0) == 0

1 def zeros(n):

Скопировать

return 0

(Используйте - для обозначения зачеркнутой буквы) Input: bbananana

1 b-anana--

Задача №4.

Output: SHELL

Hаписать метод bananas, который принимает на вход строку и

2 b-anan--a 3 b-ana--na 4 b-an--ana 5 b-a--nana 6 b---anana 7 -banana--8 -banan--a

10 -ban--ana -ba--nana ₁₂ -b--anana Скопировать

Основа:

PYTHON

9 -bana--na

1 def bananas(s) -> set: result = set() # Your code here! return result Скопировать

1 assert bananas("banann") == set() 2 assert bananas("banana") == {"banana"} 3 assert bananas("bbananana") == {"b-an--ana", "-banana--", "-b--anana",

PYTHON

Для проверки:

"b-ana--na", "b---anana", "-bana--na", "-ba--nana "-ban--ana", "b-anana--"} 6 assert bananas("bananaaa") == {"banan-a-", "banana--", "banan--a"} 7 assert bananas("bananana") == {"ban--ana", "ba--nana", "bana--na", "b--Скопировать Задача №5.

PYTHON 1 primesL = [2, 5, 7]2 limit = 500 3 List of Numbers Under 500 Prime Factorization 70 [2, 5, 7][2, 2, 5, 7] 140 [2, 2, 2, 5, 7] 280 [2, 5, 5, 7]350 [2, 5, 7, 7] 490

Haпиcaть метод count_find_num, который принимает на вход список

предел (limit), после чего попробуйте сгенерировать по порядку все числа.

Меньшие значения предела, которые имеют все и только простые множители

простых множителей (primesL) и целое число,

простых чисел primesL.

SHELL 1 primesL = [2, 5, 7]2 limit = 500

5 из этих чисел меньше 500, а самое большое из них 490.

3 count_find_num(primesL, val) == [5, 490] Скопировать Основа: PYTHON

1 def count_find_num(primesL, limit):

your code here

return []

PYTHON

Тема: Светлая Авто Тёмная

Для проверки:

Скопировать

Скопировать

1 primesL = [2, 3]2 limit = 200 3 assert count_find_num(primesL, limit) == [13, 192] 5 primesL = [2, 5]6 limit = 200 7 assert count_find_num(primesL, limit) == [8, 200] 9 primesL = [2, 3, 5]10 limit = 500 assert count_find_num(primesL, limit) == [12, 480] primesL = [2, 3, 5]limit = 1000assert count_find_num(primesL, limit) == [19, 960] 16 primesL = [2, 3, 47]limit = 200assert count_find_num(primesL, limit) == [] Скопировать

Если вы нашли ошибку, отправьте нам пул-реквест! def...

 \wedge

Статья была полезной?

list, dict, set comprehensions

<u>GitHub O проекте Лицензии При поддержке Яндекс Практикума</u>