

a.miklin [Углубленный Python]: Сдать решение

Настройки	Инфо	Итог	Посылки	Положение участников	Отправить вопрос	Сообщения
Выйти из систем	лы [a miklin]					

02:17:55 / **RUNNING**

A B C D E F G H I

Сдать решение задачи А

Полный балл:				
Бонусные баллы:				
Ограничение времени:	1 c			
Ограничение реального времени:	5 c			
Ограничение памяти:				

Выражения-генераторы

Вам предлагается набор из задачек, для каждой из которых решение — одно выражение-генератор.

1. Трансформация выражения (переведите выражение слева в выражение справа):

```
'python' -> ['pppp', 'yyyy', 'tttt', 'hhhh', 'oooo', 'nnnn']
```

2. Трансформация выражения:

```
'python' -> ['p', 'yy', 'ttt', 'hhhh', 'ooooo', 'nnnnnn']
```

3. Трансформация выражения:

```
range(16) -> [0, 3, 5, 6, 9, 10, 12, 15]
```

4. Трансформация выражения:

```
[[1, 2, 3], [4, 5, 6, 7], [8, 9], [0]] -> [1, 2, 3, 4, 5, 6, 7, 8, 9, 0]
```

Замечание: запрещается использовать функции chain и chain.from_iterable из модуля itertools.

5. Пифагоровы тройки.

Выведите все уникальные <u>пифагоровы тройки</u>, где каждое из чисел не превосходит n. На вход подается число n (n ≤ 30).

```
15 -> [(3, 4, 5), (5, 12, 13), (6, 8, 10), (9, 12, 15)]
```

▼ Подсказка

Обратите внимание, что для любой пифагоровой тройки (a, b, c) из примера выполнено условие $a \le b \le c$.

6. Трансформация выражения:

```
([0, 1, 2], [0, 1, 2, 3, 4]) -> [[0, 1, 2, 3, 4], [1, 2, 3, 4, 5], [2, 3, 4, 5, 6]]
```

7. Транспонирование матрицы.

```
[[1, 2], [3, 4], [5, 6]] -> [[1, 3, 5], [2, 4, 6]]
[[1, 3, 5], [2, 4, 6]] -> [[1, 2], [3, 4], [5, 6]]
```

Замечание: запрещается использовать сторонние библиотеки, например, scipy и numpy.

8. Трансформация выражения:

```
["0", "1 2 3", "4 5 6 7", "8 9"] -> [[0], [1, 2, 3], [4, 5, 6, 7], [8, 9]]
```

▼ Подсказка

Используйте функцию map для конвертации строк в числа. В этом задании вам должен помочь метод split() для строк.

9. Трансформация выражения:

```
range(0, 7) -> {'a': 0, 'b': 1, 'c': 4, 'd': 9, 'e': 16, 'f': 25, 'g': 36}
```

10. Трансформация выражения:

```
['Alice', 'vova', 'ANTON', 'Bob', 'kAMILA', 'CJ', 'ALICE', 'Nastya'] -> {'Alice', 'Anton', 'Kamila', 'Nastya', 'Vova'}
```

Подсказка

Оставьте все уникальные имена длины больше 3. Вам может пригодиться set.

Формат результата

Оформите свое решение в виде функций solution{n} (arg), где {n} — номер задания. На вход этим функциям в качестве аргумента будут подаваться выражения из левой части. Поместите ваши функции в словарь.

```
solutions = {
  'solution1': solution1,
```

```
'solution2': solution2,
'solution3': solution3,
'solution4': solution4,
'solution5': solution5,
'solution6': solution6,
'solution7': solution7, 'solution8': solution8,
'solution9': solution9,
'solution10': solution10,
```

Примечания

Напомним, что выражение [i ** 2 for i in range(10)] является примером выражения-генератора.

Каждой задаче соответствует один тест. Задачи и тесты пронумерованы в одном порядке. Например, если у вас не проходит 4-й тест, значит вы решили 4-ю задачу неправильно.

Если вы не знаете, как решить задачу с номером k, то оформите ее решение следующим образом:

```
def solution{k}(arg):
    pass
```

В данном задании не предполагается использование каких-либо собственных функций, помимо функций solution(n). Если же у вас СОВСЕМ не получается решить задачу без них, то написание и использование дополнительных функций не запрещено.

Сдать решение

Язык: python3 - Python3 3.6.5 Файл Choose File No file chosen

Отправить! Отправить!

Предыдущие решения этой задачи

Номер решения	Время	Размер	Задача	Язык	Результат	Пройдено тестов	Баллы	Посмотреть исходный текст	Просмотреть протокол
541	99:31:31	766	Α	python3	ок	10	10	<u>Просмотр</u>	<u>Просмотр</u>
536	98:55:38	808	Α	python3	ок	10	10	<u>Просмотр</u>	<u>Просмотр</u>
535	98:53:47	805	А	python3	Неполное решение	9	9	<u>Просмотр</u>	<u>Просмотр</u>



ejudge 3.7.6 #1 (2018-08-14 14:12:28).

Copyright © 2000-2018 Alexander Chernov.