

Тестовые задания: программист Python/Django

Решите как можно больше задач.

Задачи можно решить частично или пропустить.

Python

Предпочтительней использовать 3 версию интерпретатора.

1. Есть строка, которая содержит латинские буквы, цифры, пробелы и символы скобок.

Удалите из строки текст, заключённый в скобки.

Напишите вариант с использованием регулярных выражений и без них.

Например:

```
"asdfhj (kla (inner) asd) port (another))(unclosed" -> "asdfhj port )(unclosed"
```

2. Есть несколько кортежей с целыми числами.

Напишите функцию [1], которая принимает на вход неограниченное число кортежей разной длины.

Результатом функции должен быть список кортежей, в которых:

значение 1 - значения кортежей

значение 2 - значение 1, возведенное в степень значения 1

Например: (1, 1), (2, 3), (2,), (2,) -> [(1, 1), (1, 1), (2, 4), (3, 27), (2, 4), (2, 4)]

Напишите функцию [2], которая принимает на вход результат функции [1].

Результатом функции должен быть список значений, возведенных в степень.

Повторяющиеся значения, следующие друг за другом, нужно удалить:

Например: ... -> [1, 4, 27, 4]

Если это возможно, приведите примеры оптимизации написанных функции для уменьшения накладных расходов.

3. Напишите юнит-тесты для проверки написанного кода.

Django и SQL

Даны модели django (таблицы), описывающую базу закупок:

models.py

```
class Store(models.Model):
    name = models.CharField(" ", max_length=64)

class Purchase(models.Model):
    store = models.ForeignKey(Store, verbose_name="")
    name = models.CharField(" ", max_length=128, db_column="product")
    price = models.DecimalField(" .", max_digits=10, decimal_places=2)
```

tables.sql (mysql)

```
CREATE TABLE `store` (  
  `id` int(11) NOT NULL AUTO_INCREMENT,  
  `name` varchar(64) NOT NULL,  
  PRIMARY KEY (`id`)  
);  
  
CREATE TABLE `purchase` (  
  `id` int(11) NOT NULL AUTO_INCREMENT,  
  `product` varchar(128) NOT NULL,  
  `price` decimal(10,2) NOT NULL,  
  `store_id` int(11) NOT NULL,  
  PRIMARY KEY (`id`),  
  KEY `purchase_store_id_fk_store_id` (`store_id`),  
  CONSTRAINT `purchase_store_id_fk_store_id` FOREIGN KEY (`store_id`)  
  REFERENCES `store` (`id`)  
);
```

Решите задачи с помощью Django ORM и SQL:

1. Получить общую стоимость покупок для каждого из магазинов
2. Вывести перечень всех покупок. Каждая строка должна содержать следующие данные:
 - название магазина
 - название товара
 - стоимость
3. Выбрать покупки, цена которых больше или равна 100 руб., сгруппировать по магазинам и посчитать количество таких покупок в каждом магазине.
Минимизировать количество обращений к базе данных и количество передаваемых данных

GIT

Напишите команды:

1. Создать и перейти на новую ветку feature_1
2. Сделать два коммита
3. Удалить второй коммит вместе с изменениями
4. Смержить ветку в master

Дополнительная задача

Решите задачу на любом языке программирования.

Дорожки в аэропорту

Вы находитесь в аэропорту в точке 0. Коридор длиной X ведёт к выходу на посадку в самолёт. В коридоре есть движущиеся дорожки, каждая из которых двигается с некоторой скоростью W_i . Вы идёте по ним со скоростью $W_i + \text{ваша скорость}$. Дорожки не пересекаются: в любом месте коридора есть только одна дорожка; но они могут следовать друг за другом.

Ваша нормальная скорость - S . Но, так как до вылета осталось уже не много, вы можете немного пробежаться со скоростью R , максимум t секунд. При этом, можно как пробежаться 1 раз, так и разбить время t на интервалы.

За какое минимальное время Вы сможете добраться до выхода на посадку?

Входные параметры

Текстовый файл.

Первая строка содержит пять натуральных чисел:

X - длина коридора в метрах (м)

S - скорость ходьбы, метры в секунду (м/с)

R - скорость бега (м/с)

t - максимальное время пробежки, секунды (с)

N - число дорожек

Следующие N строк содержат три натуральных числа:

B_i - начало дорожки (м)

E_i - конец дорожки (м)

W_i - скорость движения дорожки (м/с)

Ограничения:

- $1 \leq S < R \leq 100$
- $1 \leq w_i \leq 100$
- $0 \leq B_i < E_i \leq X$
- $E_i \leq B_{i+1}$
- $1 \leq t \leq 100$
- $1 \leq X \leq 100$
- $1 \leq N \leq 20$

Ответ

Минимальное число секунд необходимое чтобы добраться до выхода на посадку.

Допустимая погрешность - 10^{-6} .

Пример

input.txt

10 1 4 1 2

4 6 1

6 9 2

console 4.000000

input.txt

12 1 2 4 1

6 12 1

console 5.500000

input.txt

20 1 3 20 5

0 4 5

4 8 4

8 12 3

12 16 2

16 20 1

console 3.538095238