



Добрый день!

*Еще раз благодарим тебя за интерес к стажировке в М.Тех. И поздравляем с прохождением на третий этап нашего отбора.*

*Перед тобой задание для вакансии «[Младший инженер по качеству данных \(DQ\)](#)».*

***Подробнее с описанием вакансии ты можешь ознакомиться по ссылке из ее названия (зажав Ctrl и кликнув).***

*Вернемся к заданию. Оно состоит из 2 частей: задачи по SQL и задача по Python. На его выполнение у тебя есть 72 часа с момента отправки письма.*

*Мы просим прислать письменное решение в одном файле в формате .doc, .docx, .pdf. Код может быть представлен ссылкой на GitHub или аналогичный репозиторий. Ссылка на код также должна быть размещена в указанном файле. Файл назови по следующему принципу: Фамилия\_Имя\_Название\_вакансии. **Задания сформулированы исчерпывающе и не требуют дополнительных уточнений. Пожалуйста, отвечай на них так, как понимаешь.***

*Если у тебя возникнут организационные вопросы по стажировке, можешь задать их, написав на почту [Alexey.stolyarov@mvideo.ru](mailto:Alexey.stolyarov@mvideo.ru) (Алексей)*

*Желаем удачи!*

# Тестовое задание SQL

В ресторане «Обыкновенный день» представлены два вида меню. Главное меню в таблице MainMenu, вегетарианское меню в таблице VegMenu.

## MainMenu.

ID_Dish	Name	Group	Price
1	Салат Цезарь	Салаты	400
2	Салат Греческий	Салаты	300
3	Овощной суп	Супы	200
4	Суп куриный	Супы	250
5	Суп Харчо	Супы	350
6	Спагетти карбонара	Основное блюдо	400
7	Мясная лазанья	Основное блюдо	350
8	Пирожочек с капустой	Десерты	50
9	Чизкейк	Десерты	250
10	Капучино	Напитки	150
11	Nescafe 3 in 1	Напитки	50
12	Чай вкусный	Напитки	100

## VegMenu

ID_Dish	Name	Group	Price
13	Винегрет	Салаты	200
2	Салат Греческий	Салаты	300
3	Овощной суп	Супы	200
14	Гречневая лапша с овощами	Основное блюдо	350
8	Пирожочек с капустой	Десерты	50
10	Капучино	Напитки	150
15	Апельсиновый сок (Fresh)	Напитки	200
12	Чай вкусный	Напитки	100

Описание столбцов таблиц: ID\_Dish – Номер блюда, Name – Название блюда, Group – Раздел меню, Price – Цена

## Задание 1

Напишите результат следующих запросов:

- a) Select MainMenu.Name, VegMenu.Name

From MainMenu **Inner join** VegMenu  
on (MainMenu.ID\_Dish = VegMenu.ID\_Dish)

b) Select MainMenu.Name, VegMenu.Name  
From MainMenu **Left join** VegMenu  
on (MainMenu.ID\_Dish = VegMenu.ID\_Dish)

c) Select MainMenu.Name, VegMenu.Name  
From MainMenu **Full outer join** VegMenu  
on (MainMenu.ID\_Dish = VegMenu.ID\_Dish)

## Задание 2

Бедный студент хочет посчитать минимальную стоимость ужина из салата, основного блюда и напитка.

- a) Напишите запрос, который возвращает минимальную стоимость блюда по разделам меню, которые смотрит студент.
- b) Выведите запросом сумму заказа, по списку из пункта а.

## Задание 3

Хитрый официант хочет предложить посетителю по самому дорогому блюду из каждого раздела Главного меню. Каким запросом можно вывести список самых дорогих блюд по разделам на основании таблицы MainMenu? Результат в виде таблицы со столбцами Name, Group, Price.

## Задание 4

Шеф-повар хочет изменить Вегетарианское меню так, чтобы все блюда в Главном и Вегетарианском меню были разными. Напишите запрос, которым можно вывести все блюда из таблицы VegMenu, которых нет в таблице MainMenu.

# Python

Дана строка длины N, где N - целое число, степень 4. Элементами строки являются латинские маленькие буквы от "a" до "z".

Строка может быть записана в виде пирамиды.

Например, строка "a" может быть записана в виде:

```
  ^  
/a\  
*___*
```

строка "abcd" в виде:

```

^
/d\
*__*
/c\b/a\
*__*__*

```

строка "abcdefghijklmnp" в виде:

```

^
/p\
*__*
/o\n/m\
*__*__*
/l\k/j\i/h\
*__*__*__*
/g\f/e\d/c\b/a\
*__*__*__*__*

```

```

^
/p\
*__*
/o\n/m\
*__*__*
/l\k/j\i/h\
*__*__*__*
/g\f/e\d/c\b/a\
*__*__*__*__*

```

и так далее.

Пирамида размером более 4 элементов может быть представлена в виде нескольких пирамид по 4 элемента.

Например, пирамида:

```

^
/p\
*__*
/o\n/m\
*__*__*
/l\k/j\i/h\
*__*__*__*
/g\f/e\d/c\b/a\
*__*__*__*__*

```

```

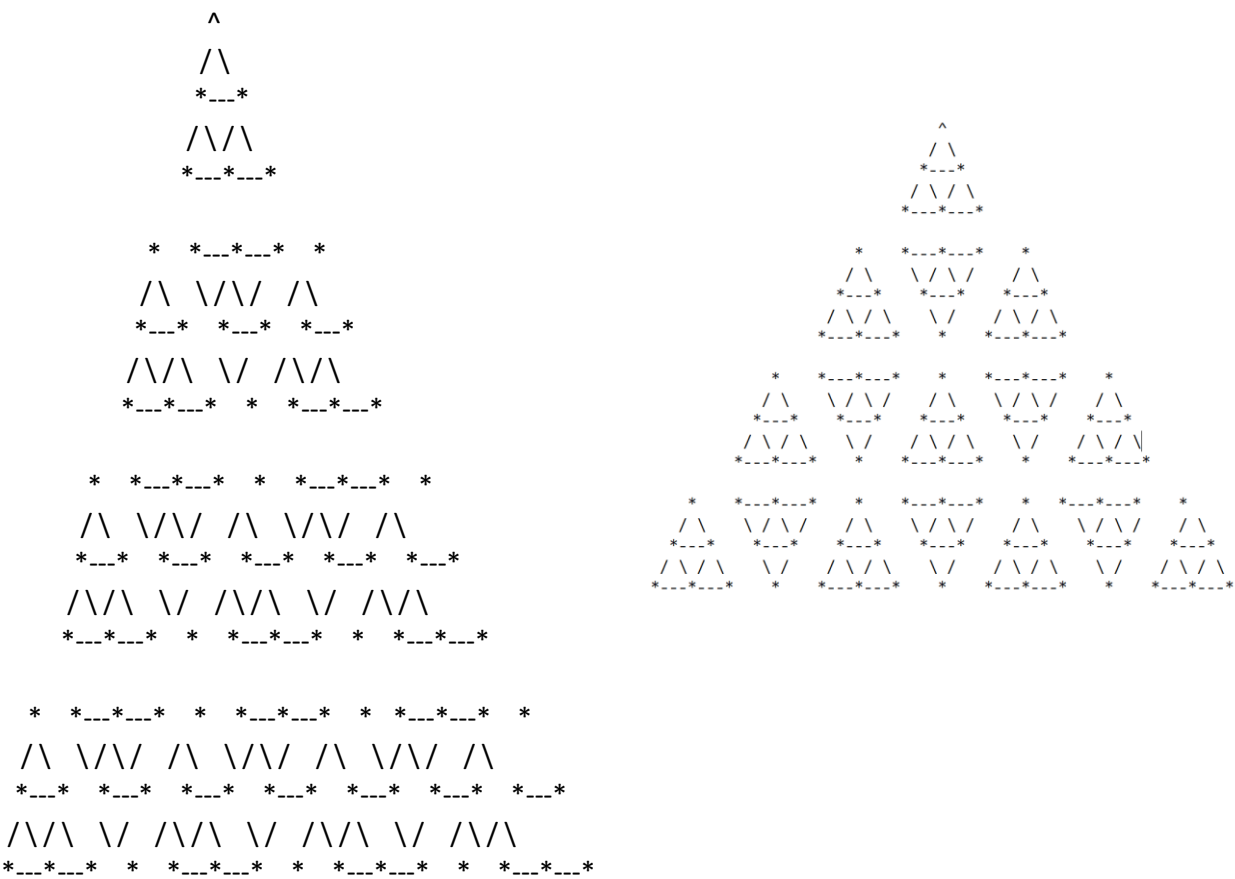
^
/p\
*__*
/o\n/m\
*__*__*
/l\k/j\i/h\
*__*__*__*
/g\f/e\d/c\b/a\
*__*__*__*__*

```

может быть представлена в виде 4х пирамид:

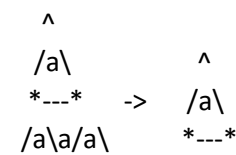


Пирамида из 64 элементов разбивается на



И так далее.

Если все элементы внутри пирамиды из 4-х элементов одинаковые, то пирамида может быть сжата до одного элемента:



\*\_\_\_\*\_\_\_\*

Пирамида размером более 4-х элементов может быть сжата, если сжимаются все 4-х элементные пирамиды, из которых она состоит.

```
      ^
     /a\
    *___*
   /a\a/a\
  *___*___*  ->  *___*
 /b\c/c\c/d\  /b\c/d\
*___*___*___* *___*___*
/b\b/b\c/d/d/d\
*___*___*___*___*

      ^
     /a\
    *___*
   /a\a/a\
  *___*___*___*  ->  *___*
 /b\c/c\c/d\  /b\c/d\
*___*___*___*___* *___*___*
/b\b/b\c/d/d/d\
*___*___*___*___*___*
```

К полученной пирамиде можно снова попробовать применить операцию сжатия.

Необходимо написать программу на языке программирования Python, сжимающую пирамиду до самого маленького размера и вывести ее строковое представление.

Исходная строка находится в файле input.txt

Результирующую строку необходимо записать в файл output.txt

Критерии оценки:

1. Программа должна работать правильно.
2. Код должен легко читаться и обслуживаться.
3. Код не должен включать в себе лишнюю функциональность, а делать только то, что требуется в задаче.
4. Код должен быть оформлен по PEP8.
5. В реализации должно быть использовано ООП.

## Пример 1:

input.txt

a

output.txt

a

### Пример 2:

input.txt

abcd

output.txt

abcd

### Пример 3:

input.txt

aaabcccabbbcd

output.txt

abcd

### Пример 4:

input.txt

aaaaaaaaabccccccaaaaabbbcccccaabbbbbbccccabbbbbbbcd

output.txt

abcd

Задание в формате txt:

