

WORKBOOK

**База тестовых заданий
с реальных собеседований
на позицию программиста**

@vlad.kolotvin

Задача 1.

Напишите функцию, которая переворачивает строку.

Задача 2.

Даны два массива:

[1, 2, 3, 2, 0] и [5, 1, 2, 7, 3, 2]

Надо вернуть их пересечение

[1, 2, 2, 3] (порядок неважен)

Задача 3.

Sample Input

["eat", "tea", "tan", "ate", "nat", "bat"]

Sample Output

[["ate", "eat", "tea"], ["nat", "tan"], ["bat"]]

Т.е. сгруппировать слова по "общим буквам".

Задача 4.

Дан массив из нулей и единиц. Нужно определить, какой максимальный по длине подинтервал единиц можно получить, удалив ровно один элемент массива.

[1, 1, 0]

Задача 5.

Дан список интов, повторяющихся элементов в списке нет.

Нужно преобразовать это множество в строку, сворачивая соседние по числовому ряду числа в диапазоны.

Примеры:

[1,4,5,2,3,9,8,11,0] => "0-5,8-9,11"

[1,4,3,2] => "1-4"

[1,4] => "1,4"

Задача 6.

Дана строка (возможно, пустая), состоящая из букв A-Z:

AAAABBBBCCXYZDDDDDEEEFFFAAAAAABBBB
BBBBBBBBBBBBBBBBBBBBBBBBBBBBBB

Нужно написать функцию RLE,
которая на выходе даст строку вида:

A4B3C2XYZD4E3F3A6B28

И сгенерирует ошибку, если на вход пришла невалидная строка.

Пояснения: Если символ встречается 1 раз, он остается без изменений; Если символ повторяется более 1 раза, к нему добавляется количество повторений.

Задача 7.

Слияние отрезков:

Вход: [1, 3] [100, 200] [2, 4]

Выход: [1, 4] [100, 200]

Задача 8.

Даны даты заезда и отъезда каждого гостя. Для каждого гостя дата заезда строго раньше даты отъезда (то есть каждый гость останавливается хотя бы на одну ночь).

В пределах одного дня считается, что сначала старые гости выезжают, а затем въезжают новые.

Найти максимальное число постояльцев, которые одновременно проживали в гостинице (считаем, что измерение количества постояльцев происходит в конце дня).

```
sample = [ (1, 2), (1, 3), (2, 4), (2, 3) ]
```

Задача 9.

Даны две строки. Написать функцию, которая вернёт True, если из первой строки можно получить вторую, совершив не более 1 изменения (== удаление / замена символа).

Задача 10.

Дается массив, каждый элемент которого состоит из двух чисел - координаты X и Y.

Этот массив это облако точек на координатной плоскости.

Нужно написать функцию, которая возвращает True, если через это облако точек можно провести вертикальную асимптоту, и если все точки симметричны относительно этой асимптоты.

Задача 11.

Дан список интов и число-цель. Нужно найти такой range, чтобы сумма его элементов давала число-цель.

```
elements = [1, -3, 4, 5]
```

```
target = 9
```

```
result = range(2, 4) # because elements[2] +  
elements[3] == target
```

Задача 12.

Дан массив точек с целочисленными координатами (x, y) . Определить, существует ли вертикальная прямая, делящая точки на 2 симметричных относительно этой прямой множества.

Note: Для удобства точку можно представлять не как массив $[x, y]$, а как объект $\{x, y\}$

Задача 13.

```
// Палиндром v2.0
// ("Ш4л4ш") => true
// ("Eva, can I see bees in a cave?") => true
// ("Шляпа") => false
```

Задача 14.

```
// Проверка массива на монотонность
// ([0, 1, 5, 9, 15]) => true
// ([0, 1, 1, 5, 9, 9, 15]) => true
// ([15, 8, 4, 2, 1]) => true
// ([0, 1, 5, 15, 4]) => false
```

Задача 15.

Нужно написать функцию, которая принимает две строки, а в результате возвращает True или False.

Например:

- * если `str_1 = 'ctr'` и `str_2 = 'catrow'`, то нужно вернуть True

- * если `str_1 = 'ctr'` и `str_2 = 'cartow'`, то нужно вернуть False

Задача 16.

Есть доска $M \times N$ ($2 < M, N < 1000000$) есть белая и черная лошадь. На входе начальная точка черной и белой лошади.

Нужно совершить минимальное количество ходов, чтобы они оказались в одной точке и вывести это количество.

Задача 17.

Вводится n и нужно посчитать сумму всех чисел от 1 до $n < 1000000$, где цифры не встречаются более одного раза.

Задача 18.

Есть пять человек: Т, М, А, Л, Д.
Они стоят в очереди за магической колой,
т.е. тот, кто выпивает его удваивается.
На входе n номер следующего, кто должен
выпить колу. Нужно найти того,
кто будет n -ным.

Задача 19.

Есть бинарное дерево поиска.
Нужно найти сумму элементов лежащих
в сегменте от L до R . Функция на вход
получает корневой элемент, левую и правую
границу. Элементы не повторяются
и границы включены.

@vlad.kolotvin