



# Тестирование на позицию разработчика

\* Required

## Задание по программированию

Ниже даны несколько задач на знание основ программирования. При решении задач необходимо использовать любой популярный язык программирования с указанием его в комментарии кода (например JS, Java, C++ и др.). Написанный код должен сопровождаться комментариями выполняемых в нем действий. Также следует воздерживаться от использования стандартных функций, так как тест в первую очередь показывает навыки построения алгоритмов.

### Шестая задача

Написать функцию выполняющую сортировку массива строк. На вход функции передается произвольный массив строк, и флаг направления сортировки. Функция должна вернуть результирующий массив.

## Code 1 \*

"""

Коды написан на языке Python 3

"""

# Исходный массив строк

```
text = ['Требуемый опыт работы: не требуется',
        'Полная занятость, полный день',
        'Мы – развивающаяся IT компания',
        'занимающаяся разработкой и внедрением комплексов
распределенных корпоративных ERP и CRM систем',
        'а также аналитических систем и интеграционных решений на базе
продуктов Oracle и IBM',
        'Мы ищем начинающих специалистов (стажеров)',
        'готовых получить интересную и востребованную профессию
разработчика Oracle Siebel CRM.',
        'Гарантируем Вам интересные, амбициозные задачи,
профессиональный рост и дружный коллектив.',
        'Успешным кандидатам будет проведен курс обучения по Oracle Siebel
CRM',
        'с последующим распределением по нашим проектам (при условии
положительной сдачи внутреннего экзамена).']
```

# Результат сортировки

```
result = []
```

# Флаг направления: 0 - по убыванию, 1 - по возрастанию

```
direction = 1
```

```
def bubble_sort(arrayToSort, direction=1):
```

"""

Функция считывает массив строк и сортирует по её длине, флаг направления по умолчанию стоит по возрастанию, при передаче второго аргумента 0 сортируется по убыванию.

"""

```
a = arrayToSort
```

```
if direction == 1:
```

```
    for i in range(len(a),0,-1):
```

```
        for j in range(1, i):
```

```
            if a[j-1] > a[j]:
```

```
                tmp = a[j-1]
```

```
                a[j-1] = a[j]
```

```
                a[j] = tmp
```

```
            #print (a)
```

```
    return a
```



```
elif direction == 0:
    for i in range(len(a),0,-1):
        for j in range(1, i):
            if a[j-1] < a[j]:
                tmp = a[j-1]
                a[j-1] = a[j]
                a[j] = tmp
            #print (a)
        return a
result = bubble_sort(text)
```

---

## Седьмая задача

Написать функцию проверяющую наличие подстроки в массиве строк. На вход передается массив строк и строка, которую необходимо найти в строках массива. Функция должна вернуть номера элементов массива, в которых встретилась указанная строка, перечисленных через запятую.



## Code 2

```
# Исходный массив строк
text = ['Требуемый опыт работы: не требуется',
        'Полная занятость, полный день',
        'Мы – развивающаяся IT компания',
        'занимающаяся разработкой и внедрением комплексов
распределенных корпоративных ERP и CRM систем',
        'а также аналитических систем и интеграционных решений на базе
продуктов Oracle и IBM',
        'Мы ищем начинающих специалистов (стажеров)',
        'готовых получить интересную и востребованную профессию
разработчика Oracle Siebel CRM.',
        'Гарантируем Вам интересные, амбициозные задачи,
профессиональный рост и дружный коллектив.',
        'Успешным кандидатам будет проведен курс обучения по Oracle Siebel
CRM',
        'с последующим распределением по нашим проектам (при условии
положительной сдачи внутреннего экзамена).']
```

```
def searching(text, search):
```

```
    """
```

```
    Функция ищет в массиве подстроку и возвращает
    индексы элементов где была обнаружена подстрока
    """
```

```
    k = 0
```

```
    for i in text:
```

```
        k += 1
```

```
        if search in i:
```

```
            result.append([k])
```

```
    return result
```

```
print(searching(text, 'по'))
```

Результат:

```
[[2], [4], [7], [9], [10]]
```

## Восьмая задача

Необходимо написать функцию, которая к дате прибавит рабочие дни. На вход передается переменная с датой, и количество рабочих дней, которые необходимо прибавить. На выходе результирующая дата. При этом при передаче отрицательного количества дней функция должна их отнять. Под рабочими днями подразумеваются будние дни без учета праздников. (решение с праздниками является опциональным)

### Code 3

Your answer

---



Page 3 of 4

BACK

NEXT

Never submit passwords through Google Forms.

This content is neither created nor endorsed by Google. [Report Abuse](#) - [Terms of Service](#)

Google Forms

