

Лабораторная работа

Для выполнения лабораторной работы воспользуйтесь сервисом [Google Colaboratory](https://colab.research.google.com/notebooks/intro.ipynb) – <https://colab.research.google.com/notebooks/intro.ipynb>. По окончании лабораторной работы пройдите тест на знание [основных методов работы со строками и файлами в Python](https://quizizz.com/join?gc=614676) – <https://quizizz.com/join?gc=614676>.

Базовый уровень

Задание 1

Напишите функцию **find_substr(subst, st)**, которая определяет имеет ли строка **st** вхождение **subst** без учета регистра. Если вхождение имеется, то функция должна возвращать слово «True», иначе «False».

Проверки:

```
print(find_substr ('Пит', 'пИтон'))
>>>True
print(find_substr ('программирование', 'ПрограммироВаНИЕ'))
>>>True
print(find_substr ('Довод', 'Повод'))
>>>False
```

Задание 2

Напишите функцию **first_last(let, st)**, которая ищет первое и последнее вхождения символа **let** в строку **st** и возвращает кортеж из первого и последнего индекса найденного символа в строке.

Проверки:

```
print(first_last('a', 'abba'))
>>> (0, 3)
print(first_last('a', 'abbaaab'))
>>> (0, 6)
print(first_last('a', 'a'))
>>> (0, 0)
print(first_last('a', 'spring'))
```

```
>> (-1, -1)
```

Задание 3

Напишите функцию **most_common_top3(st)**, в которую передается строка **st**, и которая выводит 3 наиболее часто встречаемых в строке символа. Пробелы при этом не должны учитываться в подсчете. Функция должна возвращать строку вида «символ – количество раз, символ – количество раз...».

Для подсчета количества вхождений символов воспользуйтесь методом Counter из модуля collections

Проверки:

```
print(most_common_top3('Python — высокоуровневый язык программирования  
общего назначения с динамической строгой типизацией и автоматическим  
управлением памятью'))
```

```
>>и - 12, о - 11, а - 10
```

```
print(top3('AaA'))
```

```
>>А - 2, а - 1
```

```
print(most_common_top3('Тише едешь — дальше будешь'))
```

```
>>е - 5, ш - 4, д - 3
```

Задание 4

Напишите функцию **add_text_to_file(st)**, которая добавляет строку **st** в текстовый файл с названием «MyFile.txt». Если такого файла не существует, то функция должна будет его создать, а если существует – найти и добавить передаваемый текст как новую строку. В первом случае функция должна возвращать сообщение «Файл был создан», а во втором – «Строка добавлена к существующим»

Для проверки существования файла воспользуйтесь методом exists из модуля os.path.

Проверки:

```
add_text_to_file("Первая строка")
```

```
>> Файл был создан
```

```
add_text_to_file("Вторая строка")
```

```
>> Строка добавлена к существующим
```

```
add_text_to_file("Третья строка")
>> Строка добавлена к существующим
```

Задание 5

Напишите функцию **read_from_last(lines, file)**, которая открывает для чтения файл **file** и выводит построчно последние несколько строк в количестве **lines**.

Проверки:

```
read_from_last(2, "MyFile.txt")
>>Вторая строка
>>Третья строка
read_from_last(-2, "MyFile.txt")
>>Вторая строка
>>Третья строка
```

Продвинутый уровень

Задание 6

Напишите функцию **camel(st)**, в которую передается строка **st**, и которая возвращает эту же строку с чередующимися верхним и нижним регистром ее букв. Например, «PyThOn». Первый символ этой строки должен начинаться с верхнего регистра.

Проверки:

```
print(camel('Копейка рубль бережет'))
>>КоПейкА рУбЛЬ БеРеЖеТ
print(camel('Из огня да в полымя'))
>>Из ОгНя Да В пОлЬмЯ
print(camel('КРАСОТА'))
>>КрАсОтА))
```

Задание 7

Напишите функцию **remove_braces(st)**, в которую передается строка **st**, содержащая текст со словами в скобках, и которая возвращает этот текст без слов в

скобках. При этом нужно учесть, что самая последняя открывающая скобка должна обязательно иметь после себя закрывающую.

Проверки:

```
print(remove_braces('Шила(лишнее (лишнее) лишнее) в мешке не утайшь  
(лишнее)'))
```

```
>> Шила в мешке не утайшь
```

```
print(remove_braces('(лишнее(лишнее))Шила в мешке не  
(лишнее(лишнее(лишнее)))утайшь'))
```

```
>> Шила в мешке не утайшь
```

Задание 8

Напишите функцию **delete_backspace (st)**, в которую передается строка **st**, со спецсимволом «@» обозначающим удаление одного символа слева от спецсимвола, и которая возвращает строку, очищенную от удаляемых символов и спецсимволом.

Проверки:

```
print(delete_backspace('пп@олс@кр@овт@оде@ец'))
```

```
>> полководец
```

```
print(delete_backspace('сварка@@@@@лоб@ну@'))
```

```
>> слон
```

Задание 9

Напишите функцию **max_length_word(file)**, которая открывает для чтения файл **file** и возвращает из него слово с максимальной длиной или список таких слов, если их несколько.

Для выполнения данного задания создайте файл «title.txt», куда запишите следующий текст:

Мороз и солнце; день чудесный!

Еще ты дремлешь, друг прелестный —

Пора, красавица, проснись:

Открой сомкнуты негой взоры

Навстречу северной Авроры,

Звездой севера явись!

Проверки:

```
print(max_length_word('title.txt'))
```

```
>> ['прелестный', 'красавица,']
```