



работа на уроке (13.02)

1. `def log_in()`
`input().keys()`

Дан словарь с парами имя пользователя-пароль:

```
passwords = {'Alex': '123', 'Batman': '5uP3rm4n_5t1Nk5', 'little_cat': 'meow111', 'Michael': 'qwerty123'}
```

Напишите функцию `log_in`, которая запрашивает от пользователя имя пользователя и пароль и сверяет их с данными в словаре. Если имя пользователя есть в словаре и пароль совпадает с соответствующим значением в словаре, то выводим на экран *добро пожаловать*. Если пароль не совпадает, пишем *неверный пароль*. Если введенного имени пользователя нет в словаре, пишем *неизвестное имя пользователя*.

Примеры запуска:

```
>>> log_in()
Имя пользователя: little_cat
Пароль: bow-wow
неверный пароль
```

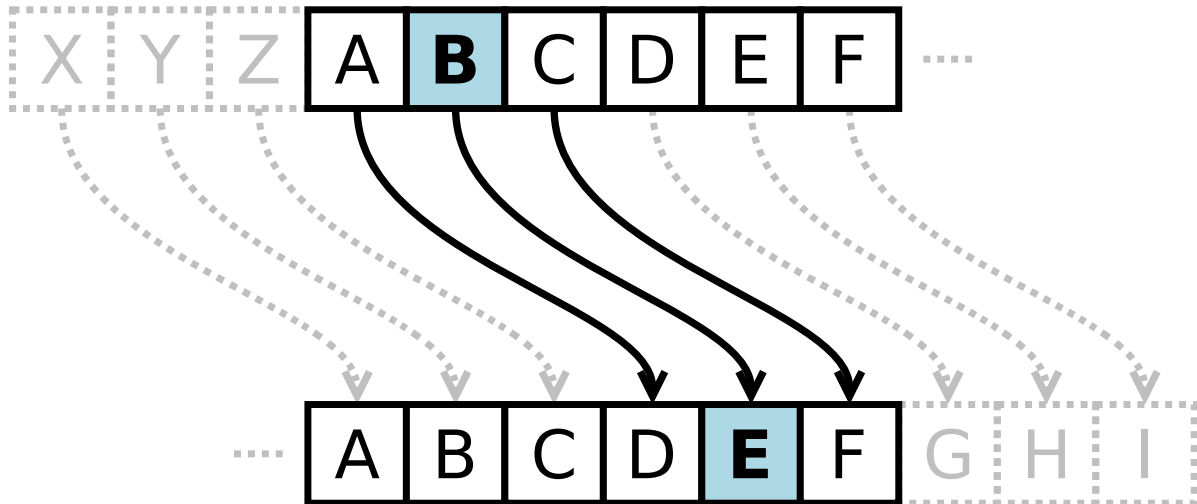
```
>>> log_in()
Имя пользователя: little_dog
Пароль: bow-wow
неизвестное имя пользователя
```

```
>>> log_in()
Имя пользователя: little_cat
Пароль: meow111
добро пожаловать
```

2. `def cypher()`

`ord()` `chr()` `for-loop`

Думаю, все знают, что такое Шифр Цезаря: это тот самый, где мы сдвигаем буквы в одну из сторон, например так:



Напишите функцию `cypher`, которая принимала бы на вход строку, а возвращала бы ее в зашифрованном виде со сдвигом 2 (a → c, b → d...).

P. S. После этого можете написать функцию `decypher`, которая помогла бы вам расшифровать эту строчку:

```
"g fmnc wms bgblr rpylqjyrc gr zw fylb. rfyrq ufyr ammnsrcpq ypc dmp. bmgle gr gl zw  
fylb gq glcddgagclr ylb rfyr'q ufw rfgq rcvr gq qm jmle. sqgle qrpgle.myicrpqlq() gq  
pcammmlcbcb)"
```

P. P. S. Напишите функцию `cypher_upgrade`, которая принимала бы на вход не только строку на шифровку, но и число, равное тому, на сколько надо сдвигать все буквы.

3. `def vowels()`

`with open(...) as ...:` `'r'` `.read()` `for-loop`

Напишите функцию `vowels`, которая получает название файла (он должен находиться в той же папке, что и запускаемый код, поэтому будет достаточно просто прочитать файл по передаваемому названию).

Верните количество гласных и согласных букв в этом файле. Предполагается, что файл будет либо на русском, либо на английском языке.

4. `def find_key()` `for-loop` `tuple()`

Все постоянно жалуются, что в словаре можно найти значение по ключу, но не ключ по значению (что логично, так как это соответствие может не быть уникальным).

Напишите функцию `find_key`, которая принимает на вход словарь и значение из этого словаря и возвращает ключ для данного значения. Если их несколько, вернуть кортеж ключей (`tuple`).

3. `def xxx()` `for-loop`

`re.sub()`

У всех иногда западают клавиши на ноутбуке и получается что-то

такое: `автоooooooooмат`. Давайте напишем функцию `xxx`, которая исправит эту оплошность. Считаем, что не бывает слов, где 2 подряд идущие буквы - это правильно. Для нас это всегда неверно. Ваша функция должна принимать одно слово и возвращать одно слово с исправлением.

P. S. Эта задачка легко решается через регулярки, но не обязательно (я не настаиваю)

...

Примеры работы функции:

```
'a' -> 'a'
'ax' -> 'ax'
'axx' -> 'axx'
'axxxb' -> 'axb'
'axxxbxxxxc' -> 'axbxc'
```