работа на уроке (20.02)

1. `for-loop`
   `break` `continue` `[-1]`
   Напишите функцию `merge_lists(li1, li2)` она должна получив два упорядоченных по возрастанию списка возвращать один, упорядоченный тоже по возрастанию, составленный из элементов полученных списков.
   Например, для `[1,3,4]` и `[2,4,6,7,8,9]` нужно получить в результате `[1,2,3,4,4,6,7,8,9]`.
   NB! НЕ используйте встроенные в питон сортировки.

2. `def is_balanced()`
   `string[]` `for-loop`
   Напишите функцию `is_balanced()`, которая вычислит все ли открытые скобки правильно закрыты.

   Примеры запуска:

   ```
   is_balanced('(()')
   False
   ```

   ```
   is_balanced('))((')
   False
   ```

   ```
   is_balanced('((()))')
   True
   ```

3. Дан `.md` файл в формате

   ```
   # заголовок
   text1
   ## подзаголовок
   ```

```
text2
## подзаголовок
text3

# ещё заголовок
## ещё подзаголовок
### и ещё
text4
```

Пожалуйста, научитесь превращать его (и ему подобных) в `json`, где вложенность заголовков — это вложенность документов. Формат должен стать таким:

```
[
    {
        "title": "заголовок",
        "text": "text1",
        "children": [
            {
                "title": "подзаголовок",
                "text": "text2",
                "children": []
            },
            {
                "title": "подзаголовок",
                "text": "text3",
                "children": []
            }
        ]
    },
    {
        "title": "ещё заголовок",
        "text": "",
        "children": [
            {
                "title": "ещё подзаголовок",
                "text": "",
                "children": [
                    {
                        "title": "и ещё",
                        "text": "text4",
                        "children": []
                    }
                ]
            }
        ]
    }
]
```

Ваше решение должно содержать функцию `md2json()`. Она должна принимать два аргумента:

1. имя `.md` файла, который надо превратить в `json`
2. имя файла, в который должен быть записан результирующий `json`.

Примечание:

- Игнорируйте пустые строки.
- Вложенность определяется количеством решёток.

4. Дано:

Ваша задача — написать две функции `encode()` и `decode()`, которые позволят обмениваться зашифрованными сообщениями с помощью кнопочного телефона.

Функция `encode()` принимает на вход слово, на выходе выдает последовательность цифр, соответствующие позициям букв в кнопочном телефоне.
Например, `carrot` транформируется в `227768`.

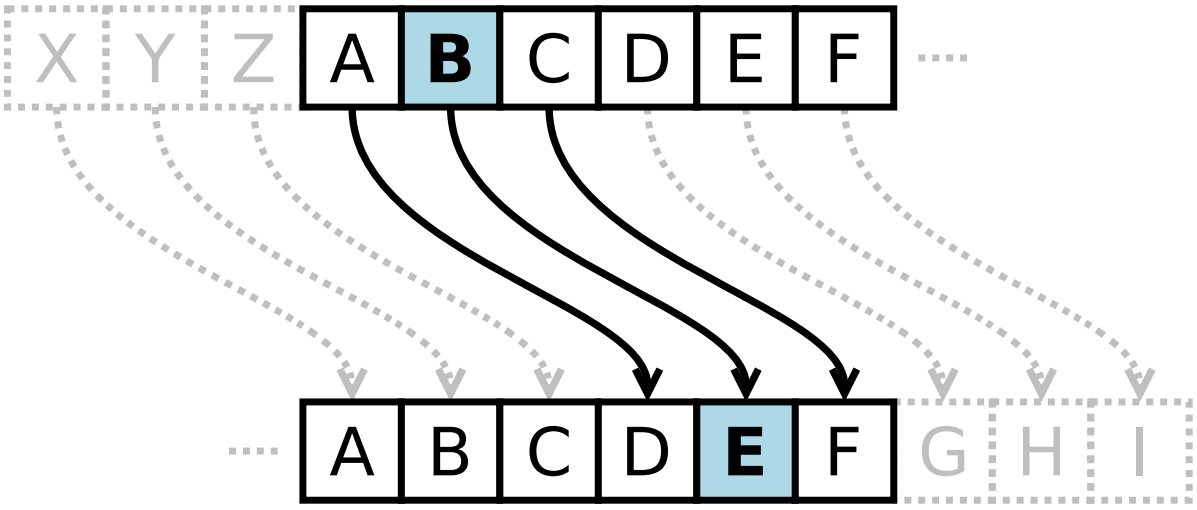
Функция `decode` получается на вход последовательность цифр `23` и переводит ее в последовательность из всех возможных
комбинаций `["ad","ae","af","bd","be","bf","cd","ce","cf"]`. Далее, из всех вариантов выбирается та комбинация(-и) которая(-ые) существуют в языке (для `23` — это `"be"`).
Определить это можно по частотности слова в Брауновском корпусе (скачать отсюда).

На выходе функции `decode()` ожидается список кортежей. Первым элементом кортежа является декодированное слово-кандидат, вторым элементом — его частотность в Брауновском корпусе. Слова, не найденные в корпусе в ответ не включать. Если ни одно слова не удается собрать, то возвращать пустой список.

5. `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.myicrpylq() gq
pcammmclbcb))"
```

   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.