

## EX03 - Booksortation

### Ülesanne

Enne ülesande lahendamist palun täida see: <https://moodle.taltech.ee/mod/feedback/view.php?id=311630>

Kaust Gitis: `EX/ex03_booksortation`

Fail Gitis: `booksortation.py`

Oled Twilight Sparkle'i hea sõber ja aitad teda ta lemmiktegevusega - booksortationiga.

`booksortation(books: list)` - peamine funktsioon.

Funktsiooni sisendiks on järjend (list) raamatutest (õigemini raamatute pealkirjadest, mis on sõned).

Sinu ülesandeks on kõik need raamatud jagada 5 kategooriasse: `spell books`, `history books`, `relics books`, `potion books` ja `other books`.

Raamatu kategooria selgestegemise nõuded on allpool järgnevatel meetoditel. Juhul, kui üks raamat kuulub tingimuste järgi mitmesse

erinevasse kategooriasse, tuleb see panna esimesse kategooriasse, mis mainitud. Ehk prioriteetjärjestus on `spell books`, `history books`, `relics books`, `potion books`. Juhul, kui raamat ei kuulu ühtegi kategooriasse nimetatud neljast, läheb ta kategooriasse `other books`.

Funktsiooni väljundiks on sõnastik (dict), kus võtmeteks on kategooriad (sõnena) ning väärtusteks järjend raamatutest, mis antud kategooriasse kuuluvad. Tagastavas sõnastikus peaksid olema kõik järjendid tähestiku järgi sorteeritud.

Juhul, kui mingi kategooria raamatuid pole üldse, ei tohi seda kategooriat ka sõnastikus esineda (ehk ei ole mingit sellist asja `{"potion books": []}`).

Sõnastiku elementide järjekord pole oluline ehk samaväärsed on nii `{"spell books": ["*Spells*"], "relics books": ["The Staff"]}` kui ka `{"relics books": ["The Staff"], "spell books": ["*Spells*"]}`.

`add_book_to_category(book: str, category: str, categorised_books: dict)` - abimeetod lisamaks raamatut sõnastikku õige kategooria alla.

Funktsiooni sisendiks on parameetrid `book` (sõne - raamat, mis tuleb sõnastikku lisada), `category` (sõne - kategooria, millesse antud raamat kuulub), `categorised_books` (sõnastik, kuhu tuleb raamat lisada).

Raamatu lisamine õigesse kategooriasse võiks käia umbes nii:

- kui antud kategooria on juba sõnastiku võtmete hulgas, siis lisa sellele kategooriale vastavasse järjendisse antud raamat
- kui antud kategooriat pole veel sõnastikus, siis lisa see kategooria sõnastikku ning pane tema väärtuseks järjend, milles on antud raamat

Funktsioon peab tagastama muudetud sõnastiku.

### Järgnevad neli funktsiooni on sarnased.

Need kõik kontrollivad, kas antud raamat kuulub vastavasse kategooriasse. Funktsioon saab sisendiks raamatu nime (sõne) ning kontrollib selles kuuluvust antud kategooriasse. Kui raamat kuulub sellesse kategooriasse, tagastab funktsioon `True`, vastasel juhul `False`.

`is_spell_book(book: str)`

Raamat kuulub kategooriasse "spell books", kui ta nimi algab ja lõpeb sümboliga `*`, kuid alguse ja lõpu sümbol ei tohi olla üks ja sama `*`.

Näiteks kuulub sellesse kategooriasse raamat nimega `"*The Horrible Spells*"`.

`is_history_book(book: str)`

Raamat kuulub kategooriasse "history books", kui iga selle raamatu nimes esinev sõna (k.a. esimene sõna) ei alga väikese algustähega. Siin juhul loeme sõnadeks kõike, mis algab peale tühikut (ehk lihtsamalt öeldes on sõnade eraldajaks tühikud).

Näiteks kuulub sellesse kategooriasse raamat nimega "The Mighty King", kuid siia ei kuulu "the Ugly Duckling", sest sõna 'the' ei alga suure tähega.

`is_relics_book(book: str)`

Raamat kuulub kategooriasse "relics books", kui ta nimi järgib mustrit, kus kõik tähed on vaheldumisi suured ja väikesed (võib alata nii suure kui ka väikse tähega). Ettevaatust tühikutega - tühikud ei ole erandiks.

Näiteks kuuluvad sellesse kategooriasse raamatud "ThE StAff" ning "rAiNiNg dUmPlInGs", kuid siia ei kuulu "ThE sTaFf" ega "rAiNiNg DuMpLiNgS" (tühikud!).

```
is_potion_book(book: str)
```

Raamat kuulub kategooriasse "potion books" juhul, kui tema nimes on võrdselt kaas- ja täishäälikuid või kui nende kogus erineb ühe võrra.

Siin ülesandes loeme täishäälikuteks ainult a, e, i, o, u ning kaashäälikuteks b, c, d, f, g, h, j, k, l, m, n, p, q, r, s, t, v, x, z, w, y.

Kõiki teisi tähti ja sümboleid võib nimes esineda kui tahes palju - need ei mõjuta raamatu kategoriseerimist.

Siia kategooriasse sobiksid raamatud "The Banana Juice" (7 täishäälikut, 7 kaashäälikut) ja "The tomato potion" (7 täishäälikut, 8 kaashäälikut - kogus erineb ühe võrra). Sellesse kategooriasse ei kuulu aga "The Green Liquid" (6 täishäälikut, 8 kaashäälikut - kogus erineb kahe võrra).

## Mall

```
"""Booksortation."""
```

```
def booksortation(books: list) -> dict:
```

```
    """
```

```
    Given a list of books (strings). Your task is to categorize and sort them.
```

```
    There are five books categories: spell books, history books, relics books, potion books and other books.
```

```
    If a book doesn't belong to any named categories, it goes to 'other books' category.
```

```
    However, if one book belongs to multiple categories, they should appear in only one  
(starting from up, whichever occurs first).
```

```
    :param books: given books as a list, list contains of strings
```

```
    :return: categorised and sorted books as a dict, where keys are categories and values are  
    list of books that match this category. Lists should be sorted alphabetically.
```

```
    """
```

```
    pass
```

```
def add_book_to_category(book: str, category: str, categorised_books: dict) -> dict:
```

```
    """
```

```
    :param book:
```

```
    :param category:
```

```
    :param categorised_books:
```

```
    :return:
```

```
    """
```

```
    pass
```

```
def is_spell_book(book: str) -> bool:
```

```
    """
```

```
    Book is a spell book if its title starts with '*' (a star, without quotes) and ends with '*' (a star, no quotes).
```

```
    However, if the starting and ending star is the same star, it is not a spell book.
```

```
    For example: '*The Horrible Spells*' is a spell book.
```

```
    :param book: given book as a string
```

```
    :return: True if given book is a spell book, False otherwise
```

```
    """
```

```
    pass
```

```
def is_history_book(book: str) -> bool:
```

```
    """
```

```
    Book is a history book if its title matches the pattern where each new word does not start with a lowercase letter.
```

```
    Word is considered anything after a whitespace.
```

```
    For example: 'The Mighty King' and 'The Age Of The Wonderbolts' are both history books.
```

```
    Then again, 'the Ugly Duckling' isn't a history books because the word 'the' doesn't start with a capital letter.
```

```
    :param book: given book as a string
```

```
    :return: True if given book is a history book, False otherwise
```

```
    """
```

```
    pass
```

```
def is_relics_book(book: str) -> bool:
```

```
    """
```

```
    Book is a relics book if its title matches the uppercase-lowercase-uppercase-lowercase... pattern.
```

```
    It can start from both upper- and lowercase letters.
```

```
    PS! Pay attention to whitespaces.
```

```
For example: 'The Staff' and 'rAiNiNg dUmPlInGs' are both relics books.
However 'The sTaFF' and 'rAiNiNg DuMpLiNgS' are not relics books.
```

```
:param book: given book as a string
:return: True if given book is a relics book, False otherwise
"""
pass
```

```
def is_potion_book(book: str) -> bool:
    """
    Book is a potion book if its title contains the same amount of vowels and consonants or the amount differs by one.

    However, it may contain as many symbols as it likes.

    The vowels are a, e, i, o, u.
    The consonants are b, c, d, f, g, h, j, k, l, m, n, p, q, r, s, t, v, x, z, w, y.

    For example: 'The Banana Juice' is a potion book (7 vowels, 7 consonants)
    and so is 'The tomato potion' (7 vowels, 8 consonants -> differ by 1).
    However, 'The Green Liquid' isn't a potion book (6 vowels, 8 consonants -> differ by 2).

    :param book: given book as a string
    :return: True if given book is a potion book, False otherwise
    """
    pass

if __name__ == '__main__':
    # All True.
    print(is_spell_book('*kana*'))
    print(is_history_book('This Is A History Book'))
    print(is_relics_book('The Staff'))
    print(is_potion_book('The Banana Juice'))
```

|                     |     |             |
|---------------------|-----|-------------|
| 2020-09-28 09:00:00 | 30% | Kõik grupid |
| 2020-09-29 09:00:00 | 20% | Kõik grupid |
| 2020-09-30 09:00:00 | 10% | Kõik grupid |
| 2020-10-01 09:00:00 | 0%  | Kõik grupid |

## Esitused ↺

My registrations

◀ EX03 - esimene etapp

Mine...

3. nädala quiz ▶

[eesti \(et\)](#)

[English \(en\)](#)

[Lae alla mobiilirakendus](#)

[Kasutustingimused](#)