

## Tartalomjegyzék

- 1 Feladatok
  - ◆ 1.1 Római számok
    - ◇ 1.1.1 Konvertálás
    - ◇ 1.1.2 Összeadás
    - ◇ 1.1.3 Keresés
    - ◇ 1.1.4 Végs?
  - ◆ 1.2 Kombinációk
  - ◆ 1.3 Zéró Számhármassok
  - ◆ 1.4 Jelszó ellen?rzés
  - ◆ 1.5 Számláló

## Feladatok

### Római számok

#### Konvertálás

Írjunk egy **konvertalas** függvényt, melynek két opcionális paramétere van, melyek default értéke False. Az egyik paraméterben megadhatunk egy római számot, ekkor visszatér a hagyományos, általunk használt arab számjegyekkel történ? leírásával integer formátumban. A másik paraméterben pedig megadhatunk egy integert és visszatér a római számos alakjával. Római számoknál használjuk a hagyományos írásmódot. Amennyiben megadjuk mindkét paramétert, térjünk vissza mindkett?vel egy listában, ha viszont egyiket sem adjuk meg, akkor térjen vissza None-nal.

*Ha elakadnál a feladat megoldásával, akkor b?ven elég az is, ha 1-t?l 150-ig és 1-t?l CL-ig tudod konvertálni a számokat.*

*Írhatunk külön függvényeket a konvertálásokra, amiket aztán felhasználunk a **konvertalas** függvénynél.*

#### Összeadás

Az el?z? függvény segítségével írjunk olyan *variadikus* függvényt mely tetsz?leges számú római számot tud összeadni.

#### Keresés

Írjunk egy függvényt, mely egy szövegben megkeresi a római számokat és visszaadja azokat egy listában.

Használjunk reguláris kifejezéseket! Importáljuk a re libraryt. (import re)

Segítséget nyújthat az idevágó el?adás notebookja, illetve az alábbi oldalak:

[w3schools](#)

[python docs](#)

Illetve egy kis segítség a megoldáshoz:

Például az alábbi kifejezéssel térhetünk vissza egy szövegből a megadott regexre illeszkedő szavakkal:  
[x.group() for x in re.finditer( r'IX|IV|I{0,3}', 'A megadott regex illeszkedik az egyjegyű a római számokra, tehát például az I és IV és V-re.')] if x.group()!='/

## Végső?

Határozzuk meg az első 10 prímszám összegét római számos alakban az alábbi szöveg segítségével! (Például legyen a keresés függvény inputja az alábbi szöveg.) 'Az első 10 prímszám római alakban így néz ki: Az egyjegyű prímszámok: II, III, V, VII. Kétjegyű prímszámok: XI, XIII, XVII, XIX, XXIII, XXIX.'

## Kombinációk

Írjunk egy *variadikus* függvényt, melynek bemenetei különböző egész számok és kimenete pedig a számok összes lehetséges kombinációja egy listában.

Tipp: Használjunk rekurzív függvényhívásokat!

## Zéró Számhármassok

Írjunk egy *variadikus* függvényt, melynek bemenetei különböző egész számok, kimenete pedig egy listában azon számhármassok, melyeknek összege 0.

## Jelszó ellenőrzés

Írjunk egy függvényt, ami ellenőrzi, hogy a regisztrációnál általunk megadott jelszó helyes formátumú-e. Az oldal a következőket kéri tőlünk:

- At least 1 letter between [a-z] and 1 letter between [A-Z].
- At least 1 number between [0-9].
- At least 1 character from [!@#].
- Minimum length 6 characters.
- Maximum length 16 characters.

## Számláló

Hozzunk létre egy **count** változót, ami számolja, hogy hányszor hívtunk meg egy adott függvényt. A függvényünk legyen egy egyszerű összeadás `add(x, y)`, ami visszatér `x+y`-al és a `count` változó számolja, hogy hányszor hívódott meg a függvény. A `count` változót ne mi magunk növeljük a függvényhívások előtt vagy után, hanem maga az `add` függvény növelje meg az értékét is.