

# Tartalomjegyzék

- 1 Feladatok
  - ◆ 1.1 f-string
    - ◇ 1.1.1
      - 1.
      - Bemutakozás
    - ◇ 1.1.2
      - 2.
      - Szorótábla
  - ◆ 1.2
    - Listagenerálás
      - ◇ 1.2.1
        - Zéró
        - mátrix
  - ◆ 1.3 Szótárak
    - ◇ 1.3.1
      - 1.
      - Gólkirály
    - ◇ 1.3.2
      - 2.
      - Prímszám-e
    - ◇ 1.3.3
      - 3.
      - Leltározás
    - ◇ 1.3.4
      - 4.
      - Csere
    - ◇ 1.3.5
      - 5.
      - Zárt terület
      - kifestése

## Feladatok

### f-string

#### 1. Bemutakozás

A bemutatkozás() függvény egyetlen paramétere legyen egy szótár, ami {'név':\*\*, 'kor':\*\*} alakú. A függvény a következ? módon m?ködjön:

```
bemutakozás({'név':Péter, 'kor':19})
Hello, Péter vagyok, 19 éves.
```

A megoldáshoz f-stringet használjunk.

#### 2. Szorótábla

Nyomtassa ki az alábbi szorótáblát f-string-ek segítségével:

```

1 2 3 4 5 6 7 8 9
+++++
1: 1 2 3 4 5 6 7 8 9
2: 2 4 6 8 10 12 14 16 18
3: 3 6 9 12 15 18 21 24 27
4: 4 8 12 16 20 24 28 32 36
5: 5 10 15 20 25 30 35 40 45
6: 6 12 18 24 30 36 42 48 54
7: 7 14 21 28 35 42 49 56 63
8: 8 16 24 32 40 48 56 64 72
9: 9 18 27 36 45 54 63 72 81

```

## Listagenerálás

### Zéró mátrix

A  $\text{zéró}(n,m)$  függvény adjon vissza egy  $n \times m$ -es mátrixot. A feladatot listagenerálással hajtsuk végre!  
Például:

```

zéró(2,3)
[[0,0,0],[0,0,0]]

```

## Szótárak

### 1. Gólkirály

Koppány és barátai minden hétvégén játszanak egy barátságos focimeccset az egyik helyi focipályán. Szeretnék tudni az év végén hogy ki rúgta közülük a legtöbb gólt, hogy egy kicsit megünnepelhessék az illet? teljesítményét. Koppány megkért minket hogy írjunk egy python függvényt, ami segít a meccs végén elkönyvelni valaki góljait.

A függvény neve legyen `golkiraly`, és három paramétere legyen

- *eredmenyek*, az év folyamán eddig l?tt gólok száma, szótár formájában, amiben mindenkinek a nevéhez hozzá van rendelve hogy eddig mennyit l?tt
- *jatekos* és *darab* az elkönyvelend? gólok löv?jének neve, és a góljainak száma

A függvény adja vissza az *eredmenyek*-nek megfelelően módosított változatát, azaz ha eddig nem szerepelt benne játékos, akkor most szerepeljen benne *darab*-bal, ha eddig is szerepelt benne, akkor pedig növeljük meg a hozzá könyvelt gólok számát *darab*-bal.

### 2. Prímszám-e

Készítsünk egy szótárat, melyben a prímszámok vannak, mint kulcsok 2-t?l 100-ig, az értékek pedig igaz-hamis értékek, hogy az adott prím Mersenne-prím-e.  
Ehhez 2 segéd függvényt érdemes írni:

- Visszaadja 2-t?l  $n$ -ig a prímek listáját (ezt már volt korábban)
- Megmondja egy prímszámról, hogy Mersenne prím-e: adjunk a számhoz 1-et, majd azt, hogy 2 hatvány-e tesztelhetjük úgy, hogy vizsgáljuk milyen maradékot ad 2-vel osztva, ha 0-t, osztjuk 2-vel, és vizsgáljuk tovább.

### 3. Leltározás

A következ? probléma merül fel a boltokban: Egy leltáros felírja egy listában szépen sorjában, hogy mit lát maga el?tt. Ezekb?l a termékekb?l egy többször is szerepelni fog, attól függ?en, hogy hány darab van bel?le. Feladat: Írjunk egy függvényt, aminek a bemenete egy lista, a kimenete pedig egy szótár, aminek a kulcsai a listában szerepl? nevek, a kulcsokhoz pedig az ? darabszámuk tartozik.

### 4. Csere

Írjon egy kétargumentumú substitute() függvényt, aminek els? argumentuma egy string, a második egy szótár lesz, amelyben a kulcsok karakterek (1 hosszú stringek), az értékek pedig stringek. Olyan új stringet kell visszaadnia, ami az els? argumentumának másolata, kivéve, hogy a szótárban kulcsként eloforduló karakterek a megfelelo értékekre cserél?dnek. Például:

```
substitute("http://wiki.math.bme.huacbcade"http://wiki.math.bme.hu, {'a': 'xyz', 'c': 'zyx'})
'xyzzzyxbzyxyzde'
substitute("http://wiki.math.bme.huacbcade"http://wiki.math.bme.hu, {'a': 'c', 'c': 'a'})
'cabacde'
```

### 5. Zárt terület kifestése

Olvassuk be a picture.txt fájlt listák listájába (minden karakter egy elem)! Írjunk egy fill(x,y) függvényt, ami ugyanazt csinálja, mint a Paint kitölt? funkciója! Az (x,y) pontból kiindulva a . helyére # jelet tesz, amíg a # jel által jelölt falba nem ütközik! A módszer rekurzív: kifestjük az (x,y) pontot, majd a szomszédait, ha azok nem # jelek. Hívjuk meg a szomszédokra (akik nem # jelek) a függvényt rekurzívan. Ha nincs kit kiszínezni, akkor álljunk meg!

```
.....
..#####.....
..#.....#.....
..#.....#.....
..#.....#.....
..#.....#.....
..#.....#.....
..#.....#####.....
..###.....##.....#.....
..#..##.....##.....#.....
..#.....##.....##.....#.....
..#.....#####.....#.....
..#.....#.....#.....#.....
..#.....##.....#.....
..#..##.....#.....
..#####.....
.....
.....
.....
.....
```