Tartalomjegyzék

- 1 Feladatok
  - 1.1 Kitekintés
    - 1.1.1 1. Mersenn prím
  - 1.2 Variadic
    - 1.2.1 2. Sum
    - 1.2.2 3. Legnagyobb közös osztó
    - 1.2.3 4. Mátrix rendezés
  - 1.3 Hatványhalmaz
    - 1.3.1 5. 0-1 sorozatok
    - 1.3.2 6. n alatt a k

Feladatok

Kitekintés

1. Mersenn prím

Írjuk meg a Mersenn-e() függvényt, melynek paramétere egy természetes szám és kimenetnek False vagy True értékkel térjen vissza attól függ?en Mersenn prím-e. A programot a logaritmus függvény használata nélkül írjuk meg.

Variadic

2. Sum

1. Írjunk egy függvényt, aminek az els? argumentuma **n**, egy **int** típusú változó. A függvény térjen vissza **True**-val, ha annyi extra paraméterrel hívták meg, mint az els? bemeneti paraméter értéke, egyébként térjen vissza **False**-szal.
2. Definiáljunk egy **szumma** függvényt, ami tetsz?legesen sok bemeneti paraméterének összegével tér vissza!
    1. Kezeljük le a kivételt, ha a paraméterek típusa nem azonos!
3. Definiáljunk egy **print_words** függvényt, úgy, hogy a megadott (akármennyi) szavakat annyiszor írja ki, amennyit megadunk bemenetként (szavanként)!
    1. Kezeljük le kivételként, ha a bemeneten nem egész számot adtak meg a szó gyakoriságára!

3. Legnagyobb közös osztó

Írjunk egy lnko() nev? függvényt, aminek a paramétere tetsz?legesen sok pozitív természetes szám és kiszámolja a legnagyobb közös osztójukat.

4. Mátrix rendezés

Adott a már házi feladatként megírt Matrix osztály:

```
class Matrix:
    def __init__(self,L):
        self.row = len(L)
        self.column = len(L[0])
        self.L = L
    def __str__(self):
        k = ''
        for i in self.L:
            for j in i:
                k = k + str(j).rjust(4)
            k += '\n'
        return k
```

Egészítsük ki egy új függvénnyel, aminek két bemenete van: Egy Matrix osztálybeli elemekb?l álló lista, egy string (novekvo vagy csokkeno) és a listában lév? mátrixokat a diagonális rész összege szerint rendezi növekv?, illetve csökken? sorrendben a második paraméter alapján. Úgy írjuk meg a függvényt, hogy ha második paraméternek nem adunk meg semmit, akkor automatikusan növekv? sorrendbe rendezze a mátrixokat.

1. Tipp: Írjuk meg el?ször a diag() metódusát egy mátrixnak, majd a rendezést a .sort() metódussal végezzük. Továbbá a fordított rendezés a sort(reverse = True) metódussal megvalósítható.

Hatványhalmaz

5. 0-1 sorozatok

Írjunk egy hatvanyhalmaz() függvényt, aminek egyetlen paramétere egy n természetes szám és a kimenete az összes n hosszú 0-1 sorozat 1 listába szedve.
Például:

```
hatvanyhalmaz(2) = [[1,1],[1,0],[0,1],[0,0]]
```

Tipp: A listát minden lépésben kétszerezzük meg, azaz f?zzük önmagához és valamelyik elemhez 0-át, valamelyik elemhez 1-est f?zzünk.

6. n alatt a k

Az el?z? feladatot felhasználva írjunk egy függvényt, nalattk()-t, aminek els? paramétere egy k szám, ami megadja, hogy hány elem? halmazokat szeretnénk csinálni az utána következ? paraméterekb?l. Ez egy variadikus függvény.