

## Tartalomjegyzék

- 1 Informatika 1. 1. zárthelyi, mintamegoldásokkal
  - ◆ 1.1 1. Numerikus gyökkeresés
    - ◇ 1.1.1 Feladat
    - ◇ 1.1.2 Mintamegoldás
  - ◆ 1.2 2. Mit ír ki?
    - ◇ 1.2.1 Feladat
    - ◇ 1.2.2 Mintamegoldás
  - ◆ 1.3 3. Rajzolás
    - ◇ 1.3.1 Feladat
    - ◇ 1.3.2 Mintamegoldás
  - ◆ 1.4 4. Váltott el?jel? összeg
    - ◇ 1.4.1 Feladat
    - ◇ 1.4.2 1. mintamegoldás
    - ◇ 1.4.3 2. mintamegoldás
    - ◇ 1.4.4 3. mintamegoldás
    - ◇ 1.4.5 4. mintamegoldás
  - ◆ 1.5 5. Egyszer? ciklus
    - ◇ 1.5.1 Feladat
    - ◇ 1.5.2 Mintamegoldás
  - ◆ 1.6 6. Relatív prímek
    - ◇ 1.6.1 Feladat
    - ◇ 1.6.2 1. mintamegoldás
    - ◇ 1.6.3 2. mintamegoldás
  - ◆ 1.7 7. Listák
    - ◇ 1.7.1 Feladat
    - ◇ 1.7.2 Mintamegoldás

## Informatika 1, 1. zárthelyi, mintamegoldásokkal

2010. október 15.

### 1. Numerikus gyökkeresés

#### Feladat

Írjunk Sage kódot, ami numerikus közelít?értéket számol a  $10 = 3^x/x$  egyenlet azon gyökére, amire  $2 < x < 4$ .

#### Mintamegoldás

```
(3^x/x == 10).find_root(2, 4)
```

### 2. Mit ír ki?

#### Feladat

Mit válaszol a Sage, ha a következ?t futtatjuk le?

```
(x / 2^x).subs(x = 3).n()
```

## Mintamegoldás

0.3750000000000000

## 3. Rajzolás

### Feladat

Rajzoljunk a Sage segítségével köröket, melyeknek középpontjai az  $y = x$  egyenesen helyezkednek el, vízszintesen 1-1 távolságra egymástól, a  $(-5, -5)$ -től kezdve az  $(5, 5)$ -ig. A körök sugara legyen 2. A körök színe legyen kezdetben fekete, a végén kék.

### Mintamegoldás

```
abra = []
for k in range(11):
    abra.append(circle((k - 5, k - 5), 2, color=(0, 0, k/10)))
show(sum(abra), aspect_ratio = 1)
```

## 4. Váltott elj?jel? összeg

### Feladat

Írjunk függvényt, aminek a bemenete egy számokból álló páratlan hosszú lista, eredménye pedig a számok váltakozó elj?jel? összege. Például

```
valtott_osszeg([18, 17, 4, 13, 7, 20, 6])
```

eredménye legyen  $-15$ , mivel  $-15 = 18 - 17 + 4 - 13 + 7 - 20 + 6$ .

### 1. mintamegoldás

```
def valtott_osszeg(sor):
    ossz = 0
    for el in sor:
        ossz = el - ossz
    return ossz
```

### 2. mintamegoldás

```
def valtott_osszeg(sor):
    ossz = sor[0]
    for fel in range((len(sor) - 1) / 2):
        ossz = ossz - sor[1+2*fel] + sor[2+2*fel]
    return ossz
```

### 3. mintamegoldás

```
def valtott_osszeg(sor):
    return sum([(-1)^k * sor[k] for k in range(len(sor))])
```

### 4. mintamegoldás

```
def valtott_osszeg(sor):
    return 2 * sum(sor[::2]) - sum(sor)
```

## 5. Egyszer? ciklus

### Feladat

Mit válaszol a Sage, ha a következőt futtatjuk le?

```
n = 0
while n < 10:
    if n < 5:
        n = n + 3
    else:
        n = n + 2
    print n
```

### Mintamegoldás

```
8
10
```

## 6. Relatív príme

### Feladat

Írjunk függvényt, ami eldönti, hogy két szám relatív prím-e. A függvény bemenete két természetes szám, kimenete 1, ha ezek relatív príme; 0, ha a két számnak van 1-nél nagyobb közös osztója. Például

```
relativ_prim(9, 12)
```

eredménye 0.

### 1. mintamegoldás

```
def relativ_prim(bal, jobb):
    eredm = 1
    if 0 == bal and 1 != jobb:
        eredm = 0
    for osztó in range(2, bal + 1):
        if 0 == bal % osztó and 0 == jobb % osztó:
            eredm = 0
    return eredm
```

### 2. mintamegoldás

```
def relativ_prim(bal, jobb):
    while 0 != jobb:
        mar = bal % jobb
        bal = jobb
        jobb = mar
    if 1 == bal:
        return 1
    else:
        return 0
```

## 7. Listák

## Feladat

Legyen

```
lista = ['a', 1, 2, 3, 'b', 4]
```

A fenti utasítás kiadása után mit válaszol a Sage?

```
lista[1]
lista[1:3]
lista[-1]
lista.count(1)
tuple(lista)
'b' in lista
```

## Mintamegoldás

```
1
[1, 2]
4
1
('a', 1, 2, 3, 'b', 4)
True
```