

## Informatika 2, 1. ZH (2019-03-04)

1	2	3	4	5	6	$\Sigma$

A feladatok megoldása elfér a feladat mellett, ha külön lapra írjuk, tegyük egy jól látható nyilat a helyére!

1. Mit írnak ki a következő python kódsorok? (4 pont)

a) (1 pont)

```
n = 5
i = 1
s = 0
while i <= n:
    s += i
    i += 1
print s
```

b) Írjunk példát! (1 pont)

```
s = raw_input()
print "Szia", s + "!"
```

c) (2 pont)

```
n = 12
while n != 1:
    print n,
    if n % 2 == 0:
        n /= 2
    else:
        n = 3*n+1
print 1
```

2. Írjunk olyan python programot ami bekér a felhasználótól egy számot (tegyük fel hogy nem-negatív egész számot kapunk) és utána bekér még annyi darab számot, amennyi az első volt. A program eztán írja ki a beírt számok közül a legnagyobbat. (4 pont)

Ha 0 darab számot kaptunk, akkor azt írja ki, hogy "nincsen". Példa bemenet:

```
5
-1
1
2
-3
0
```

Kimenet:

```
2
```

3. Elméleti kérdések (2 pont)

a) Mi a hasonlóság string (str) és tuple között?

b) Mi lesz a `type(3.14)` parancs eredménye?

c) Hogyan lehet idézőjelet írni string-be?

d) Mi lesz két lista összege, például `[1,2] + [3,4]`?

4. Írjunk olyan függvényt, ami két szóról eldönti, hogy melyikben van több magánhangzó (aeiou) a hosszához képest. Egy szóra számítsuk ki a magánhangzók számát osztva a betűk számával (ez egy valós szám a  $[0, 1]$  intervallumban) és ha ez elsőben ugyanannyi vagy kevesebb ez a szám, akkor `True`-t adjunk vissza, különben `False`-t. (4 pont)

```
def maganszam(a):
```

```
def magankisebb(a, b):
```

Példa:

```
>>> maganszam("teve")
0.5
>>> magankisebb("teve", "kutya")
False
```

5.

a) Írjunk olyan függvényt, ami egy adott számhoz visszaadja az attól 1-ig terjedő számok listáját! (2 pont)

Például  $5 \mapsto [5, 4, 3, 2, 1]$

```
def visszaszamlalas(n):
```

b) Írjunk olyan függvényt, ami egy szótárhoz és egy kulcshoz megnézi, hogy van-e az adott kulcs a szótárban és ha van akkor adja vissza az ahhoz tartozó értéket, ha nincs akkor tegye be azt a kulcsot a szótárba 0 értékkel és az így kibővített szótárat adja vissza! (2 pont)

```
def ertek(d, k):
```

Példa:

```
>>> ertek({1:2, 3:4}, 3)
4
>>> ertek({1:2, 3:4}, 4)
{1:2, 3:4, 4:0}
```

6. Az alábbi kódban 4 hiba található, karikázzuk be ezeket! (2 pont)

```
d = {12:1, [1,2]:2, (1,2):3, "12":4, }
if 1 in d
    d[1] += 1
else if "2" in d:
    d["2"] += 1
else:
    d.append((12, 1))
print(d)
```