

Informatika 2, 3. zárthelyi (2016-05-17)

1	2	3	4	5	6	Σ

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

1. Mit írnak ki a következő python kódsorok? (3 pont)

a)

```
import argparse
parser = argparse.ArgumentParser()
parser.add_argument("-f", "--filename", action="store", type=str)
parser.add_argument("-w", "--wololo", action="store", type=str)
args = parser.parse_args()
print args.filename, args.wololo
```

Ha a python `test.py -f macska -w cica` paranccsal futtatjuk.

b)

```
def kiskutya(l, i, x):
    if i >= len(l):
        l.append(x)
    else:
        l.append(l[-1])
        for j in range(len(l)-2,i,-1):
            l[j] = l[j-1]
        l[i] = x
L = [1,2,3,4,5,6]
kiskutya(L,6,"X")
print L
```

c)

```
import numpy as np
X=np.arange(1,7,1).reshape(2,3)
Y=np.linspace(1,6,6).reshape(3,2)
print X.dot(Y)
```

2. A numpy csomag segítségével oldjuk meg az alábbi feladatokat: (4 pont)

a) Közelítsük az alábbi integrál értékét téglalap módszerrel, 0.01 szélességű téglalapokkal!

$$\int_{-5}^2 e^{\sin(x)} dx$$

b) Generáljunk 100 véletlen pontot a $[-3, 4]$ intervallumba! Számoljuk ki az átlagukat!

c) Hozzunk létre egy véletlen 7 dimenziós egységvektort!

3. Elméleti kérdések (utolsó 2 bónusz) (2+0 pont)

a) Mit tárol a `sys.argv`? Mi a `sys.argv[0]`?

b) Mire való a `csv` modul?

c) Mit értünk rekurziónál leállási feltételen?

d) Mi a rendezett bináris fák legfontosabb tulajdonsága?

e) Who shot first?

f) Melyik bolygóról származik Chewbacca?

4. Adott egy `code.txt` fájlunk, melyben egy olyan programnyelvű kód van, ahol a zárójelben levő elemek kommentek. Szeretnénk kitörölni ezeket a kommenteket. Hozzunk létre egy fájlt `nocomment.txt` névvel, melyben a `code.txt` kommentek nélküli része van! Nincsenek egymásba ágyazott kommentek és egy komment abban a sorban ér véget ahol kezdődött, viszont egy sorban lehet több komment is. *(5 pont)*

`code.txt`:

```
int a = 6; (kutya macska int 5)
float g; (double g;) int h = 4; (6)
```

`nocomment.txt`:

```
int a = 6;
float g; int h = 4;
```

5. Kitaláltunk egy játékot, ahol a pontot egy string alapján számoljuk. A számolási szabályok a következők:

- Minden nem szám karakter 1 pontot ér, kivéve az `S`.
- `S` jelzi a pontozandó rész kezdetét, előtte semmi nem ér pontot. Csak egy `S` lesz a szövegben, ő nem ér pontot.
- Alapból 1-es szorzóval számoljuk a pontokat, de ha egy számhoz érünk az hozzáadódik a szorzóhoz, például az első 2-es 3-ra növelné a szorzót, de a számok maguk nem érnek pontot.
- Ez a szorzó csak a rákövetkező karakterekre érvényes.
- A 0 speciális jelentésű, visszaállítja a szorzót 1-re.

Írjunk függvényt `pont` néven, mely egy stringet kap és visszaadja a pontértékét! *(4 pont)*

```
ab3cSca2b -> 5
aS3ab2d   -> 14
S54c      -> 10
S54c0b    -> 11
```

Segítség: `c in "0123456789"`

6. Az alábbi bináris fát implementáló kódban 6 hiba található, 1-1 egyedi és 2-2 nagyon hasonló, mik ezek? Az `insert` metódus új elemet ad hozzá a fához, az `is_list` leellenőrzi, hogy listává fajul-e, végül a `sum` összeadja az elemeit. (Az algoritmusban van a hiba, nem a szintaktikában.) *(2 pont)*

```
class Node(object):
    def __init__(self, data):
        self.data = data
        self.left = None
        self.right = None
    def insert(self, data):
        if self.data > data:
            if self.left is None:
                self.left = Node(data)
            else:
                if self.right is None:
                    self.right = Node(data)
```

```
def is_list(self):
    if self.left is not None:
        return self.left.insert(self.data)
    if self.right is not None:
        return self.right.insert(self.data)
    return True
def sum(self):
    s = self.data
    if self.left is not None:
        s += self.left.sum()
    return s
```