

Informatika 2, 3. zárthelyi (2018-05-14)

1	2	3	4	5	6	Σ

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

1. Mit történik a következő python kódsorok hatására? (3 pont)

a) Mi íródik ki a képernyőre, ha a python test.py macska cica paranccsal futtatjuk?

```
import sys
print len(sys.argv)
with open(sys.argv[1], "w") as f:
    f.write("Hello")
```

b) Mit ír ki,

```
with open("macska.txt", "r") as f:
    f.readline()
    print f.readline().strip()
```

ha a macska.txt tartalma az alábbi?

1. Üdvözöllek
2. kedves
3. hallgató
- 4.

c) Mi kerül a macska.txt fájlba?

```
import csv
with open("macska.txt", "w") as f:
    writer = csv.DictWriter(f, fieldnames=["Neptun", "Nev"])
    writer.writeheader()
    writer.writerow({"Nev": "Bruce", "Neptun": "BATMAN"})
```

2. A lap jobb oldalán található a bináris fák egy implementációja. Egészítsük ki két metódussal. Egy számolja meg a csúcsok számát, egy pedig a fa mélységét. (3 pont)

3. Elméleti kérdések (4 pont)

a) Hogyan keresünk egy rendezett listában bináris kereséssel? Fogalmazzuk meg az algoritmust saját szavainkkal.

b) Hogyan oldjuk meg a Hanoi tornyai feladványt rekurzív algoritmussal? Fogalmazzuk meg az algoritmust saját szavainkkal.

c) Mi a különbség dinamikus programozás és rekurzió között?

d) Mikor kiegyensúlyozatlan egy bináris fa? Írjunk példát!

```
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:
                self.left.insert(data)
        elif self.data < data:
            if self.right is None:
                self.right = Node(data)
            else:
                self.right.insert(data)
```

4. Van egy olyan programnyelvű kód, ahol az idézőjelben (") levő részek kommentek. Szeretnénk kitörölni ezeket a kommenteket. Írjunk függvényt, ami visszaadja a komment nélküli szöveget. *(4 pont)*

```
def torol(szoveg):
```

Például

```
>>> print torol('kiskutya "cica macska" python')
kiskutya python
```

```
class Node(object):
    def __init__(self, kappa):
        i = -1
        if kappa.find("+") != -1:
            i = kappa.find("+")
        elif kappa.find("-") != -1:
            i = kappa.find("-")
        elif kappa.find("*") != -1:
            i = kappa.find("*")
        elif kappa.find("/") != -1:
            i = kappa.find("/")
        if i != -1:
            self.data = kappa[i]
            self.left = Node(kappa[:i])
            self.right = Node(kappa[i + 1:])
        else:
            self.data = float(kappa)
            self.left = None
            self.right = None

    def __repr__(self):
```

5. Egészítsük ki a “paint bucket”-függvény implementációját! *(4 pont)*

Tegyük fel, hogy a beolvasott kép 20 × 20-as és # és . karaktereket tartalmaz.

```
picture = []
with open('picture.txt') as f:
    for line in f:
        picture.append(list(line.strip()))
```

```
def fill(x, y):
```

```
fill(0,0)
for x in picture:
    print ''.join(x)
```

6. A jobb oldalon lévő kód a számológép program implementációja, Egészítsük ki egy reprezentáló függvénnyel, ami kiírja a kifejezést a kifejezésfából visszaépítve. *(2 pont)*

Például:

```
print Node("2*(3+4)")
(2)*((3)+(4))
```