1	2	3	4	5	6	Σ

Informatics 2, 3rd midterm (2016-05-17)

The answers should fit next to the questions, if you used a separate paper let us know clearly!

1. What will be printed after these python commands? *(3 points)*

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 littlepuppy(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]
littlepuppy(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. Solve the following exercises using the `numpy` package. *(4 points)*

a) Estimate the following integral with left Riemann sum. Use a $\Delta x = 0.01$ step.

$$\int_{-5}^{2} e^{\sin(x)}\,\mathrm{d}x$$

b) Generate 100 random points in the interval $[-3, 4]$. Calculate their mean (average).

c) Generate a random vector of unit length in 7 dimensions.

3. Theoretical questions *(2 points)*

a) What is stored in `sys.argv`? What is `sys.argv[0]`?

b) What is the `csv` module for?

c) What is the *halt condition* in recursion?

d) What is the characteristic property of ordered binary trees?

4. Given a file `code.txt` and you want to delete all the comments in parenthesis. Write a file called `nocomment.txt` in which you copy every part of `code.txt` which are net enclosed in a parenthesis. The comments do not span across lines but you can have several comments in one line. *(5 points)*

`code.txt`:

```
int a = 6; (puppy cat int 5)
float g; (double g;) int h = 4; (6)
```

`nocomment.txt`:

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

5. We play a special game where the score of a player is determined from a string. The rules are as follows:

- Every non-digit character worth 1 point, expect `S`.
- `S` marks the beginning of a worthy part, everything before an `S` worth 0 points. There will be only one `S` in the text, that character itself does not worth any points.
- The scores are multiplied with 1, but if you encounter a digit then the multiplier is increased with that number. For example the digit 2 raises the multiplier to 3. But the digits themselves do not worth any points.
- This multiplier is used for the next character only.
- The digit 0 resets the multiplier to 1.

Write a function called `point` which calculates the score according to these rules. *(4 points)*

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

Hint: `c in "0123456789"`

6. There are 6 mistakes in the implementation of a binary tree. There are 2 distinct mistakes and 2-2 very similar ones. Find these! The `insert` method adds a new element, the `is_list` checks whether the tree is a list (without conjunctions), the `sum` method returns the sum of elements. *(2 points)*

```
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
```