MSc / PGD Data Analytics

Database and Analytics Programming (H9DAP)

Continuous Assessment 1: In-class Test

This test contributes towards a maximum of 30% of the marks for the module.

Instructions:

- You may consult any of the notes and code from the classes or the labs.
- All code must be fully commented
- Upload your code to Moodle
- This is an individual assessment – there should be no communication (including verbal and electronic) between any students during the assessment.

Question 1: (50 marks in total)

2miners.com provides an API to access crypto-currency mining information for a number of its crypto-currency mining pools. You have been provided with a JSON file on Moodle (**eth2miners.json**) containing mining information for the miners of the **ETH** crypto-currency.

The structure of the JSON file is as follows:

```
MinerReturnModel{
        hashrate        number($float)
        miners  MinerModel{
                minerUid        MinerUidModel{
                                        lastBeat        integer($int64)
                                        hr              integer($int64)
                                        offline         boolean default: false
                                        currentLuck     number($float)
                                }
                }
        minersTotal     integer($int64)
        now             integer($int64)
        workersTotal    integer($int64)
}
```

a) Create a function to import the **eth2miners.json** JSON data file. Your function should include appropriate exception handling clauses.

[10 marks]

b) Using a loop structure, print the

   i. *lastBeat*
   ii. *currentLuck*

values associated with each minerUid in the JSON data for which the offline is false

[10 marks]

c) Calculate the average *currentLuck* value for the miners (offline both true and false). [Note: you can re-use the loop from part b) if you wish.]

[10 marks]

d) Extract the JSON data for the miners and write it to a CSV file. Your file should also contain the column names: **minerUid**, **lastBeat**, **hr**, **offline**, **currentLuck**.

[20 marks]

Question 2: (30 marks in total)

a) Create a 2-dimensional NumPy *ndarray* filled with 1000 numbers. Ensure that your array has 50 rows and 20 columns.

[5 marks]

Compute the sum of all entries in the first two rows.

[5 marks]

Compute the sum of all entries in the last two columns.

[5 marks]

Get (filter) the values that are greater then 100 and less than 200

[5 marks]

b) Using slicing, split this array into 10 separate arrays. The number of rows in each array should be equal, and there should still be 20 columns.

[5 marks]

Add together corresponding array entry values from each of the 10 arrays.

[5 marks]

Question 3: (20 marks in total)

Given the following string:

```
Tumble Trouble Twixt Two Towns!
Was the Moon soon in the Sea
Or soon in the sky?
Nobody really knows YET.
```

a) Using a single regular expression, write a function to highlight all occurrences of a capital letter followed by at least two lowercase letters. Your answer should look like this:

```
{Tum}ble {Tro}uble {Twi}xt {Two} {Tow}ns!
{Was} the {Moo}n soon in the {Sea}
Or soon in the sky?
{Nob}ody really knows YET.
```

[5 marks]

b) Using another single regular expression, write another function that highlights where two ‘o’ characters appear beside each other in the given string only if there is another occurrence of two ‘o’ characters appearing beside each other subsequently in the same line. Your answer should look like this:

```
Tumble Trouble Twixt Two Towns!
Was the M{oo}n soon in the Sea
Or soon in the sky?
Nobody really knows YET.
```

[5 marks]

c) Given the following text (xml document)

```
<books>
    <book>
        <name>A Song of Ice and Fire</name>
        <author>George R. R. Martin</author>
        <language>English</language>
        <genre>Epic fantasy</genre>
    </book>
    <book>
        <name id=232>Train Dreams</name>
        <author>Denis Johnson</author>
        <language>English</language>
        <genre>Novel</genre>
    </book>
</books>
```

use a single regular expression to extract the names of the books from the <name> tag elements. Your answer should look like this:

A Song of Ice and Fire
Train Dreams

[10 marks]