Lab7 – Mutability, Tuples and Arrays

Lab Objectives

The aim of this worksheet is to familiarise you with tuples and arrays data types:

- Pay attention to mutability issues on data types;
- Make appropriate use of the data types: lists, tuples and arrays.

Question 1 – Tuple of tuples

Complete the following code as described by the docstrings inside the function

```
#given input data
data = ((45, 'foot'), (21, 'basket'), (10, 'hand'), (24, 'foot'), (21,
'hand'))
def unique_data_items(data):
    ''' input data is made of ((int, string), (int, string), ...)'''
    unique_nums = () #initialising the tuple
    unique_words = ()
    ''' Add code to fill the tuples unique_nums and unique_words with
        numbers and words that are unique'''

    ''' returns the pair (tuple) of unique numbers and words'''
```

A sample run of the function `unique_data_items(data)` will return `(4, 3)`

Question 2 – Arrays vs Lists

Given a list of numbers, we would like to count the number of negative numbers as well as their average. Write the following functions as specified:

- `Count_negatives_1`, which takes in a list of numbers and returns a tuple composed of the number of negatives and the average of those negative numbers. Use list for this function. You can use a loop but avoid the use of a loop; instead use the method sorted on a modified list of the input.
- `Count_negatives_2`, which takes in a list of numbers and returns a tuple composed of the number of negatives and the average of those negative numbers. But, use array operations to implement the function.

A sample run of say the function `count_negatives_1([-2, 4, -3, 1])` will return `(2, -2.5)`

Question 3 – 2D arrays

Download the txt file `data.txt` and save it into your local directory. For each function that you will write, use exceptions whenever appropriate.

1. Open the txt file and observe it contains rows of numbers separated by spaces. Please do not edit nor modify the given txt file.
2. Write the function `read_data`, which takes in a file name and returns a two-dimensional array storing all the data in the given file.
3. Write the function `row_averages` that takes in the data output from the function `read_data` and returns an array composed of the averages for each row of the data.
4. Write the function `col_averages` that takes in the output from the function `read_data` and returns an array composed of the averages for each column of data.