# MSC - INF101B
# C Programming Language
# Travaux Pratiques (TP) 4

Panagiotis PAPADAKIS

## 1 Introduction

The practical session TP 4 proposes coding exercises that will allow you to strengthen your understanding of the notions presented during Course 4. In the beginning of each TP, you should create a separate folder that will contain your work. For the fourth TP 4, create a folder named TP4 from a terminal and go into this directory.

## 2 Exercises

### 2.1

Write a program that can retrieve a web-page containing information about the price of stocks and present to the user the array of stocks ordered depending on their properties (*Name*, *Current price (p)*, *Day change*). You may use the **wget** utility which allows to retrieve the content of a web-page by the linux command line. By the following example, you can save the content of a web-page into a text file:

```
wget --quiet --output-document=stocks.txt http://www.site.com
```

For this exercise, you should retrieve the stock market information from the site `https://www.hl.co.uk/shares/stock-market-summary/ftse-100`.

**NOTE1**: the file that you will retrieve by using wget contains HTML (HyperText MarkUp Language) code, which is formatted human-readable web site code. Inside the *stocks.txt* file that will contain the HTML code, the information related to each stock is preceded by a line containing the string "**View equity details for** ". For every appearance of that word in the file, you should create a stock and retrieve the data corresponding to the *Name*, *Current price (p)* and *Day change* of that stock. To do so, you will need to use the C functions `strtok` , `strstr` and `strcmp`.

**NOTE2**: you should define an appropriate C *structure* that will hold each stock and its properties/fields.

**NOTE3**: you will need to use the `qsort` function and define new functions for comparing the different fields of the stock C structure.

An example of file "stocks.txt" and expected program output of ordered stocks depending on the user's choice can be accessed from the moodle `https://bit.ly/2Vbab7o`.

## 2.2

Separate your code to a main source file and a functions source and header file. Create a makefile that can build (compile and link) 2 versions of your program, one for debugging and the other for normal (optimized) execution.

**Finally, add rules in your makefile allowing to run the exercise of the TP, using various possibility of inputs.**

## 2.3

Repeat Exercise 2.2 for all exercises of the previous TPs (TP1-TP3).