## CE151 ASSIGNMENT 1 2018

Credits: 20% of total module mark

Deadline: `14-Nov-2018 11:59:00`
Submission of this assignment will be via FASER: your program must be demonstrated during your lab in week 8 or 9.

You should refer to sections 5 and 7 of the Undergraduate Students' Handbook for details of the University policy regarding late submission and plagiarism; the work handed in must be entirely your own.

### Introduction

This assignment involves 8 individual exercises. You must use the template file **`assignment1.py`** supplied through the CE151 Moodle page. The file contains the definition of the functions you have to write; you should remove the **`return None`** line from each function and put your own code. The file also contains code to facilitate testing your assignment; upon running the file, it will return the number of tests failed, or OK if all of them are passed. You must use Python 3. Code written in Python 2 will receive no marks.

### Submission

You should submit your `assignment1`.py file to the online FASER submission system before the deadline. In addition, you should have your code ready to be tested in your lab in week 8 or 9, so if you develop it at home make sure you have a copy on your M: drive or on a USB stick.

### Marking Scheme

A total of 100 marks is available for the assignment.

10 marks are awarded for completion of the two lab exercises set in the week 3 and week 5 labs (5 marks for each).

90 marks are awarded based on the testing results of your code for the function 1-8 implemented in the assignment1.py file. Specifically, the functions are tested against a suite of 32 test cases; each passed test will earn you 2.5 marks, up to a max of 80. The remaining 10 marks are assigned based on the lab demonstration, which includes short questions on the implemented code.

If you fail to demonstrate your program in the lab 20% will be deducted from your mark.

If you are unable to attend both the week 8 and week 9 labs due to illness or other unforeseen circumstances, you must inform me by email (g.stracquadanio@essex.ac.uk) no later than 14/11/2018.

*Any 20% deduction of marks will not be applied to the marks awarded for the lab exercises.*

**Exercise 1 (fun_exercise_1)**
The function takes in input a number `x` and a list of numbers `y`, and returns a value as follows:

- If `x` is odd, `fun_exercise_1` subtract 1 from all the elements of `y` and then returns its sum.
- If `x` is even, `fun_exercise_1` multiplies each element of `y` by 2 and then returns its sum.
- If `x` is zero, `fun_exercise_1` returns the sum of all the elements in `y`.

**Exercise 2 (fun_exercise_2)**
The function takes in input a list of numbers `y`, and returns a value as follows:

- If the 1st element of `y` is odd, `fun_exercise_2` multiples all the element of `y` by 2 and returns their product.
- If the 3rd element of y is odd, `fun_exercise_2` divides all the element of `y` by 2 and returns their sum.
- Otherwise, it returns the sum of the square of the elements in `y`.

**Exercise 3 (fun_exercise_3)**
The function takes in input a list of numbers `y` of even length, and returns a list `z` of length len(`y`)/2, such that:

- `z[i] = True      if y[-i] < y[i]`
- `z[i] = False     otherwise`

**Exercise 4 (fun_exercise_4)**
The function takes in input a list of numbers `y`, and return the 3rd biggest number in `y`. If len(`y`) < 5, it returns the biggest len(`y`) – 1 element. e.g. if len(`y`) is 3, it returns the 2nd biggest element, and so on. If the list is empty returns **`None`**.

**Exercise 5 (fun_exercise_5)**
The function takes in input a string `x` and returns **`True`** if and only if the string is not palindrome.

**Exercise 6 (fun_exercise_6)**
The function takes in input a string `x` and returns a new string `y`, where letters from A to L, are reported in lower-case, and letters from M to Z are reported in upper-case.

**Exercise 7 (fun_exercise_7)**
The function takes in input a list of strings `x` and returns an integer **`ptr`** if and only if **`x[ptr]`** is a substring of at least one of the other strings in `x`. Otherwise, it returns -1.

**Exercise 8 (fun_exercise_8)**
The function takes in input a string `x`, and returns the second most frequent character in the string.