**Project Description**

You have been given the flexibility to structure the

code in the way that you think is best following the principles of re-usability, maintainability,

information hiding and clarity. It is important to break your code into small, meaningful functions

and appropriately comment on your code.

Part 1(70%)

A single .py file containing a working version of all the tasks is required.

For this part, you should import pandas, NumPy and matplotlib ONLY to accomplish

the tasks below.

Your program will detect and plot the start, end and low of downturns in quarterly UK GDP data. The

file series-250318.csv (source: Office for National Statistics (ONS)) contains yearly and quarterly

GDP figures (millions of GBP) over time. We are only interested in the quarterly data for our

project. We will call a downturn, a period that starts with x consecutive quarters of GDP decline

and ends with x consecutive quarters of GDP growth. A downturn low is a quarter within the

downturn period which had the lowest GDP.

**Task 1**: (15 %) The user should be prompted to specify the input file and the data should

be saved in a DataFrame (you can assume that the format of the input file is always similar to

the format of series-250318.csv). The user should also enter the start and the end of the period

(a quarter of a year in both cases) that he/she would like to investigate. Finally, the user should

enter a strictly positive number x, necessary for the determination of the downturn periods.

**Task 2:** (35 %) Your program should return the periods of downturns (start and end) and

their lows.

**Task 3:** (20 %) Your program should visualise each different downturn period using suitable

plots.

Part 2 (30 %)

A single .py file containing a working version of all the tasks is required. Only use the packages NumPy, pandas and math.

A ski instructor has n pairs of skis to assign to n skiers. The ith ski has a length of li units and the

jth skier has a height of hj units where 1 <= i; j \_ n, and li and hj have integer values. How should

the instructor assign skis to skiers so that the resulting sum of the absolute differences between

the height of skiers and the length of the assigned skis is as small as possible?

Implement an efficient Python program for solving the above problem. Use the sample input

files that are provided to test your implementation. Your

implementation should be able to read all lengths and heights from a given CSV file of the form:

<n>

<length 1>,: : : ,<length n>

<height 1>,: : : ,<height n>

The following example shows 3 ski lengths (10,34,25) and skier height (40,23,34):

3

10,34,25

40,23,34

Your program should return and display the sum along with the details of which skis were assigned

to which skiers. For example, assuming that an algorithm returns the same ordering as the input

for the above example, the output should be:

SUM: 50

ASSIGNMENTS (SKI - SKIER):

(1 - 1),(2 - 2),(3 - 3)