Lab 5 – CSE 101 (Spring 2019)

1. Objectives

The primary objectives of this lab assignment are:

- To revise and practice PythonLabs isearch function.
- To understand how to edit a broken program to fix some logical errors.
- To practice with for-loops to solve problems.
- To practice with lists to solve problems.

2. Tutorial project

Go to page 95 of the book Explorations in Computing by John S. Conery. Start a new shell session on command prompt/terminal and type the following statement:

```
>>> from PythonLabs.IterationLab import *
```

Complete all the tasks T1 – T19 from the tutorial.

3. Interleave: Fix error(s)

Download lab5problems.py into your CSE101/`lab5` folder. Open `lab5problems.py` in **PyCharm**.

In the **Fix Error** type of problems, you will be given a problem as usual, but with a buggy "solution" provided for you. Your task is to find the error(s) and make the function work properly. Note that you are free to just rewrite the solution yourself, but as the problems get harder, it might be easier to find the error(s) rather than rewrite the entire solution.

Fix the function `interleave` which takes as its only parameter a list called `nums`, which contains sublists of integers. Create a new list by *interleaving* the lists of integers and return the interleaved list. This process is most easily understood by looking at an example.

Example: Let's consider a list of two sublists, with three elements per sublist: `[[1, 2, 3], [4, 5, 6]]`:

- The first element of the first list (denoted `nums[0][0]`) will be the first element of the new list.
- The first element of the second list (denoted `nums[1][0]`) will be the second element of the new list.
- The second element of the first list (denoted `nums[0][1]`) will be the third element of the new list.
- The second element of the second list (denoted `nums[1][1]`) will be the fourth element of the new list.
- The third element of the first list (denoted `nums[0][2]`) will be the fifth element of the new list.
- The third element of the second list (denoted `nums[1][2]`) will be the sixth element of the new list.

The final list will be [1, 4, 2, 5, 3, 6].

Note: You may assume that all sublists in nums will have the same number of elements. If the list or the sublists are empty list(s), the return value should be an empty list.

Example calls:

interleave([[1, 2, 3], [4, 5, 6], [7, 8, 9]]) returns [1, 4, 7, 2, 5, 8, 3, 6, 9]

interleave([[0, 1, 0], [1, 0, 0], [0, 0, 1]]) returns [0, 1, 0, 1, 0, 0, 0, 0, 1]

interleave([[1, 2], [3, 4], [5, 6], [7, 8], [9, 0]]) returns [1, 3, 5, 7, 9, 2, 4, 6, 8, 0]

4. Weighted GPA Calculator

To lab5problems.py add a function named gpa_calculator that takes two parameters, in this order:

1. grades: A list of strings that represent the letter grades. For simplicity, you only need to think about the letter grades A, B, C, D, and F. There will be no grades like B+ or C-, and the grades can be either uppercase or lowercase.
2. credits_worth: A list of integers which represent the corresponding credits of the courses with the letter grades given in the grades list.

Let's look at an example about how the arguments work together. Suppose grades = ['A', 'B', 'f'], and credits worth = [4, 3, 1]. This means for the first course the student took and got an 'A' in, that course counts for 4 credits; for the second course, for which the student got a 'B', the course counts for 3 credits; lastly, for the third course, for which the student got a 'f', that course counts for 1 credit.

Your function should calculate and return the weighted grade point average after converting each grade to a "grade point" in the range 0.0 through 4.0. If the list of grades contains a grade not listed in the table below, or the list of grades is empty, simply return None. Note that the general formula when calculating the GPA is shown below:

```
total number of grade points earned
-----------------------------------
 total number of credits attempted
```

You can also check out http://academicanswers.waldenu.edu/faq/73219 for an example on how to calculate the GPA.

Grade	Grade Points
A	4.0
B	3.0

C	2.0
D	1.0
F	0.0

Note: You don't need to worry about invalid course credits, like `0` credits for a course. You may also assume that `grades` and `credits_worth` have the same length. Also, the returned value will always be a **floating-point** number.

Example calls:

```
gpa_calculator(['A', 'A', 'A', 'A'], [4, 3, 2, 3]) returns 4.0
gpa_calculator(['F', 'F', 'F', 'F'], [2, 4, 3, 5]) returns 0.0
gpa_calculator(['F', 'A', 'B', 'A', 'A'], [2, 5, 5, 3, 1]) returns
3.1875
```

5. Submission

Submit completed `lab5problems.py` program on blackboard. Your program should have correct implementation of `interleave` and `gpa_calculator` functions. Each of correct functions will receive 2.5 points.