



Problem Set:	Assignment: AG08	Semester:	Fall 2017
Points:	See autograder	Due Date:	See autograder
Date Set:	See autograder	Instructor:	Dr. Nauman
Course:	CS101 Introduction to Computing		

1 Lists and Tuples

Since you are reading this, you have already downloaded and extracted the zip file.

Read through the whole description below before starting with the assignment.

1.1 Tasks to do

1. Take a look at `a08.py`
2. There are two main tasks to complete and an additional one if you want to get full score.
 - (a) Write a function with the name `find_cumulative_marks`. This function will be given a list of tuples. For example:

```
results = [  
    ('p101111', 'Ali Khayam', 64, 78.5, 89, 25, 99),  
    ('p101112', 'Mudasser Farooq', 14, 28.5, 83, 76),  
    ('p101113', 'Tamleek Ali', 87, None, 1.6)  
]
```

Each tuple represents the scores of a student. The first element is the roll number of the student and the second is the full name. The rest of the elements are the student's score in assignments.

Some students may have submitted more assignments than the others. For instance, in the example above, the first student has submitted 5 assignments while the third has submitted only 3 (of which, the second got a None – meaning it was invalidated). A None in the score is the same as a 0.

Your mission, should you choose to accept it, is to write `find_cumulative_marks` in a way that it returns the cumulative marks of each student. So, for the case above, it should return a list of tuples as follows:

```
[  
    ('p101111', 'Ali Khayam', 355.5),  
    ('p101112', 'Mudasser Farooq', 201.5),  
    ('p101113', 'Tamleek Ali', 88.6)  
]
```

Your function should be able to handle any number of tuples i.e. student records passed to it. If a None is passed to the function, it should return a None back. If an empty list is passed, it should, obviously return an empty list back.

Make sure you preserve the order of the tuples within the outer list. If the order of students is changed, your tests may not pass.

- (b) The second function you need to write is `find_top_student`. It also takes the same input as the first function. However, it finds the student with the top cumulative marks and returns only that student. For example, if we pass to it the `results` list (shown in the example above), it should return back: `('p101111', 'Ali Khayam')`
- (c) If you want to pass the final 2 tests, you need to ensure the following condition is handled by `find_top_student` function:
If two students have the same total marks, the function should return both of them in a list. So, if for example, we pass this as input:

1.1 Tasks to do

```
results = [  
    ('p101111', 'Ali Khayam', 64, 10),  
    ('p101112', 'Mudasser Farooq', 50, 24),  
    ('p101113', 'Tamleek Ali', 10, 20)  
]
```

It should return back:

```
[('p101111', 'Ali Khayam'), ('p101112', 'Mudasser Farooq')]
```

since both of these students have 'top marks'.

As before, make sure you preserve the order of students.

3. Run local tests and if they pass, submit the assignment using the submission command given on the Autograder assignment page. (Same as the first assignment.)