Here is some example input text:

'''Mary had a little lamb
its fleece was white as snow
and everywhere that Mary went
the lamb was sure to go'''

Write a `compute_average_word_length(instring,unique=False)` function to compute the average length of the words in (`instring`). If the `unique` option is `True`, then exclude duplicated words. For example, in the example input text above, the should be counted exactly once for the average word length if `unique=True`. Note that the words are case sensitive. Rem carefully validate your inputs using `assert` statements.

Please put your Python code in a Python script file and upload it. Please retain your submitted source files! Remember t practices we discussed in class. You can use any module in the Python standard library, but third-party modules (e.g., Nu are restricted to those **explicitly** mentioned in the problem description.

Tips:

- After you have submitted your file, do **not** use the browser back or reload buttons to navigate or open the page ir browser tabs, as this may cause your `attempts` to decrease unexpectedly. It may take up to thirty seconds for your processed, so please be **patient**.
- If you find yourself back at the main page without any feedback or change in your `attempts` then it means that you out or crashed in some unexpected way.
- Ensure that your development environment does not presume the existence of certain packages for the autograde autograder does not have anything other than the standard library and those third-party libraries **explicitly** name description.
- Do not leave extraneous statements in your code like test cases, print statements, or anything else besides what is evaluate your submission because the the autograder will spend its limited time executing those lines, which may unexpected crashes or timeouts.

浏览... 0103.py Upload Python source code file

Correct! Back to assignments. functional points = 7 /7 and validation points = 3/3