

Username

CODE CORRECTNESS AND APPROACH:		
Q1: e_estimator:	Test case correctness (max = <input type="text" value="0.5"/>)	0.5
	Approach: <input type="text" value="0.5"/> = Appropriate approach; <input type="text" value="0.25"/> = Over-complicated/overly simplistic approach; <input type="text" value="0"/> = No real attempt made	0.5
Q2: spiral_word:	Test case correctness (max = <input type="text" value="1"/>)	1.0
	Approach: <input type="text" value="0.5"/> = Appropriate approach; <input type="text" value="0.25"/> = Over-complicated/overly simplistic approach; <input type="text" value="0"/> = No real attempt made	0.5
Q3: authattr_worddict:	Test case correctness (max = <input type="text" value="1"/>)	1.0
	Approach: <input type="text" value="1"/> = Appropriate approach and use of data structures; <input type="text" value="0.5"/> = Over-complicated/overly simplistic approach; <input type="text" value="0"/> = No real attempt made	0.75
Q4: authattr_oop:	Test case correctness (max = <input type="text" value="1"/>)	1.0
	Approach: <input type="text" value="1"/> = Appropriate approach and use of data structures; <input type="text" value="0.5"/> = Over-complicated/overly simplistic approach; <input type="text" value="0"/> = No real attempt made	1.0
Q5: authattr_authorpred:	Test case correctness (max = <input type="text" value="1"/>)	1.0
	Approach: <input type="text" value="0.5"/> = Appropriate approach and use of data structures; <input type="text" value="0.25"/> = Over-complicated/overly simplistic approach; <input type="text" value="0"/> = No real attempt made	0.5
Adherence to style guide (—comments):		
	<input type="text" value="1"/> Strong adherence <input type="text" value="0.5"/> Partial adherence <input type="text" value="0"/> Little or no adherence	1.0
Commenting:		
	<input type="text" value="1"/> Helpful, insightful and succinct <input type="text" value="0.5"/> Somewhat helpful, but sometimes sparse/overly verbose <input type="text" value="0"/> No comments, randomly sprinkled and unhelpful, or too verbose	1.0
TOTAL (/10):		9.75

Question No.	Line(s)	Comment
Q1	4-9	There's no real need for this to be an 'inner function' instead of just a normal function, right? Please avoid using functions define within functions as it makes the code a little more complicated than necessary. You can just define the function on the outside of your main function.
Q1	all	Otherwise, great work!
Q1	7-10	Nice use of recursion to calculate the factorial.
Q2	all	A nice approach, well done.
Q2	19-21	This final line is overly complex—there is too much complexity concentrated into the one (logical) line. You should try to break this up, possibly using multiple variables with meaningful names to store the intermediate parts of the result, and combining the variables afterwards, thus spreading out the complexity over multiple logical lines and making it a lot easier to grasp.
Q3	all	Your program passes all the tests, but I can see a potential problem with its correctness.
Q3	17	You have to be careful when using the 'index' method. It only returns the index of the first matching character in the string. That means if there are multiple occurrences of your character in the string, your value of 'c2' might be wrong. If you want to iterate through the indices as well as the letters, you should just use a 'range' or look into using 'enumerate'.
Q3	18	After line 18, won't 'letter' now just store the same thing as the variable 'char' already stores? What was the point of having both of these variables?
Q4	all	A very clean approach, well done. In particular, good job solving this problem with the use of a helper function to avoid repeating large chunks of code.
Q4	all	I think you could have taken this further and broken the problem down into more sub-problems, each with your own function. This would help make your program simpler overall. No marks deducted this time, but we'll be on the lookout for this kind of thing in future projects.
Q4	all	Is there any particular reason why you had to define your helper function inside your main function? Couldn't you have just defined a regular function that does the same thing? 'Nested' function definitions like this make code hard to follow for other programmers because the functions can rely on variables other than the ones explicitly provided in their argument list. You should avoid these nested definitions where possible.
Q5	all	Nice work!
Q5	16-18	Could you simplify this part by using the join method?
Q5	all	The method of sorting the authors by reversing each tuple before sorting is clear enough. However, for a more conventional method to sort a list of tuples (or anything) by the values in reverse (or with any non-default ordering) is to use the sorted function's 'key' keyword argument. You might like to look into that, for future reference.
general	all	Great use of docstrings. Good choices of variable names. Well-placed comments punctuate your approach to each problem. Excellent use of whitespace (blank lines) to separate logical sections of your code. All of these things make your code a pleasure to read. Well done!