Some updates for Coursework questions

27 November 2023

Question 1

This question is asking you to create a function `all_rounder()` with three arguments:

1. the sequence to be updated/used
2. the sequence method name **(always in `str`)**
3. the required arguments for the sequence method in (2)

Here the main problem is caused by the third argument, especially due to the test case of

```
>>> all_rounder('#', 'join', '(\'Jack\', \'Mahmut\')')
'Jack#Mahmut'
```

Here my explanation is: the third question might be passed in formats like this:

- `all_rounder('#', 'join', ('Jack', 'Mahmut'))`
- `all_rounder('#', 'join', ['Jack', 'Mahmut'])`
- `all_rounder('#', 'join', "('Jack', 'Mahmut')")`

Your function should understand the passed argument is a sequence even if I pass it as a `str` as given below.

This should never return `"(#'#J#a#c#k#'#,# #'#M#a#h#m#u#t#'#)"` *if you code it correctly!!!*

Similarly, your function should work for the first test case as given below:

```
>>> all_rounder([1, 5], 'append', "[3, 4, 2, 104]")
[1, 5, [3, 4, 2, 104]]
```

Lastly, for the quotation usages, I will confirm that I am not going to use the same quotation in the single definition requires **string within a string** (e.g. the case in `'(\'Jack\', \'Mahmut\')'`). A usage like this requires using escape sequence for quotations and makes the inout highly complicated. For the test cases, this usage will be ignored and quotations will always be different. Thus, the second test case has been changed to:

```
>>> all_rounder('#', 'join', "('Jack', 'Mahmut')")
'Jack#Mahmut'
```

Question 3

Issue 1

Whilst finding/counting a string in the given text file, you have **consider and exclude overlapping cases** from the counting process.

An example: if `t = "Oktay"`, then your count should **exactly** find/count this case and **exclude cases such as `"Oktayy"`, `"Oktay!"`, etc**.

Issue 2

The variable `sel` has only two options: `"find"` or `"count"`. Any other options will not be considered.

Question 4

Issue 1

in the first version of the coursework, it is clear that we do not explain the procedure requires for the cases where `check_ties = False`. Please carefully read the given below for this case:

*If this `flag` is set to `False`, your code should only compare **the first letters of two consecutive words**, and if any two words start with **the same letter, they are not considered** to be in alphabetical order, and so **the whole sentence should also not be considered** to be in alphabetical order.*

Issue 2

This question requires you to answer two small questions at the end (1 mark each). For these small questions, your answers should be added as a comment into the `find_alphanumerical_order()` function definition.

E.g.

```
# Commentary languages (1 Mark): How many English comments did we find:
# My answer is 12423523252 ....
#
```

```
#Name 3 footballers (1 Mark): Name 3 footballers that appear in comments:
# My answer is: Oktay, Luis and Sam.
```

Question 5

Test case given for this question shows a wrong return where the number of superheros return is only 21 after calling:

```
>>> a,b,c,d = pd_query()
>>> print(d['Name'])
```

in the normal circumstances, the returned number of superheros should be the length of 44 (indexes from 0 to 43), where **Captain Marvel, Hulk,** and **Wonder Women** are at indexes **8, 20,** and **43**, respectively.

```
In [ ]:
```