A series of exercises to work with loops, and with string and list methods.

You can find descriptions of string methods online

You can find descriptions of list methods online

You are free to use string or list methods, unless I expressly forbid it below. You now know TWO kinds of loops - condition controlled (WHILE) and count controlled (FOR). If you choose to use a loop, try to choose the correct loop. Do you KNOW how many times you want the loop to run (count controlled), or should it be a loop that ends under some condition (condition controlled)? REMEMBER, both loops can be made to work, but selecting the right one tells the reader what is supposed to happen.

Ok. Let's get to it.

(1) (a) Get a sentence from the user. Count the number of 'z' in the sentence. Print the count once. You can either use a loop, or if there's an applicable string method, you can use that.

(b) Get a sentence from the user. SPLIT that sentence into words, and count the number of times the word 'zebra' appears in the sentence.

(2) "The temperature today is going to be around >100 degrees F"

Create a variable to hold the sentence above. Use SLICING to extract only the numeric temperature from the string. Print that number.

(3) (a) Use the string method 'find' to search for the letter 'd' in the sentence above. Remind yourself what the method 'find' returns. In the comments of your code, tell me what is returned (what data type, what it represents) when you search for the letter d in the sentence.

(b) The string method find has optional arguments. Optional arguments are like in slicing, when we can add a second : followed by a number to indicate the increment, for example 'nick'[0:3:2], where that 2 means take every second letter. We don't HAVE to include an increment, but we can. It's an optional argument.

Use the command line interface, or lookup the method in the python documentation, to find out what the optional arguments for the find string method are. Optional arguments are shown in

square brackets. When we use them we do NOT include the square brackets, they are just there to indicate that we do not have to provide those arguments if we don't want to. Tell me in the comments of your code what the optional arguments the find method are, and what they represent.

(4) Instead of you writing the index values for question (2), use the string method find to *discover* the index values for slicing the numeric temperature out of the original string IMAGINING THAT YOU DID NOT KNOW IT WAS 100 degrees (that is, it could be 97 degrees. Or 4. or 10045, if we were on the surface of a hot planet).

HINT: Think about what you'd look for to find the temperature in the example above, if you didn't know it was 100 degrees. You can assume that the format of the sentence given above will always hold true. i.e. the following would be examples:

"The temperature today is going to be around >97 degrees F"

"The temperature today is going to be around >4 degrees F"

"The temperature today is going to be around >10045 degrees F"

Your code should work for all of these!

What can you search for before the temperature and after it? How might that help you?

(5) Pallindromes are sentences which read the same forward as backward. For example "Madam I'm Adam". We're going to write some code that asks the user for a sentence, and prints "Yes, it's a pallindrome" if it is a pallindrome, and "No, it's not" if it is not.

DO NOT USE SLICING, OR ANY MEANS TO REVERSE A STRING TO DO THIS!!

(a) Write code that works on something simple to start with like 'racecar'. Start by writing pseudo code. You know something is a pallindrome when...what? THINK about the steps YOU would take. I want you to write down the steps one at a time that you would take to decide something was a palindrome. Write these steps in comments in your code. Then translate it into python. *MAKE THIS WORK USING A LOOP. What does that loop look like? WORK THIS OUT ON PAPER. You can get the first letter from a string. What do you want to compare it to, to see if it's a palindrome. And if they match, then what? And if they don't, then what?*

(b) NOW amend your code to work on more complex problems, like 'Madam I'm Adam'.

HINT: you need to replace some things in your original strings (like spaces and punctuation). There is a string method 'replace' that we can use that is useful. Play with it and find out what it does. To replace something with nothing, you are really saying I want to replace this thing with the empty string - '' (two single quotes next to each other). Read the documentation about it.

HINT: Punctuation is annoying. Let's start by imagining that punctuation is just periods, commas and single quotes. How might you use this to go through the text above, and remove this punctuation?

You want to go through each item of possible punctuation, and use the replace method to replace that piece of punctuation with nothing.

Replace works globally, one call to replace replaces ALL instances of that thing in the string. Make sure you check that what you write WORKS.

HINT: Don't forget about case. 'M' will not match 'm'. If ONLY there were string methods that could help you with case. Oh wait. There are.

(6) We're going to work with a block of text. Create a variable called footyText, as below:

```
footyText = "Football refers to a number of sports that involve kicking a ball \

with the foot to score a goal. The most popular of these sports worldwide is \

association football, more commonly known as just 'football' or 'soccer'. \

Unqualified, the word football applies to whichever form of football is the \

most popular in the regional context in which the word appears, including \

association football, as well as American football, Australian rules football, \

Canadian football, Gaelic football, rugby league, rugby union, and other \

related games. These variations of football are known as football codes.Various \

forms of football can be identified in history, often as popular peasant games. \

Contemporary codes of football can be traced back to the codification of these \

games at English public schools in the eighteenth and nineteenth century. The \

influence and power of the British Empire allowed these rules \
```

of football to spread to areas of British influence outside of the directly \

controlled Empire, though by the end of the nineteenth century, distinct \

regional codes were already developing: Gaelic Football, for example, \

deliberately incorporated the rules of local traditional football games in \

order to maintain their heritage. In 1888, The Football League was founded in \

England, becoming the first of many professional football competitions. During \

the twentieth century, several of the various kinds of football grew to become \

among the most popular team sports in the world."

NOTE that the use of the '\' in the lines allows the string to span over a new line, which makes it easier to read (you can do this with your comments too!). You should be able to copy and paste this text into your code. The '\' does NOT appear in the final string.

Can you remember from class how to split this into words? HINT: There's a handy string method that does that.

THEN what? You end up with WHAT kind of data? Can you use a loop to go through this data and write code that:

- prints how many words are in this text

- counts the number of times the word football is used

DO NOT use string methods (like count or find) to achieve this!

BE careful. COUNT the word football by hand, and see how many there are, and MAKE SURE that your code returns the same number.

WRITE out pseudocode FIRST. It will help!