CSCE 1035 Lab 08 Fall 2021

The goal of this lab is to help you get comfortable working with string using a variety of the tools and methods available through the Python language. While some of the techniques you will be implementing can be completed using simpler, built-in functionality, the purpose is to familiarize yourself with the processing and manipulating of strings.

1. zyBooks Labs

Please follow the link on Canvas to complete the following zyBooks labs:

- 7.6 *LAB: Name format*
- 7.7 *LAB: Count characters*
- 7.8 *LAB: Mad Lib - loops*
- 7.10 *LAB: Acronyms*
- *7.11 LAB: Contains the character*
- *7.12 LAB: Warm up: Parsing strings*

2. Removing Letters from Strings

In this lab you will write a small, but complete Python 3 program called **`Lab8A.py`** that removes all occurrences of a single character from a given string. This can be accomplished as follows:

a. Create a user-defined function called `remove_letter(sentence, letter)` that accepts two string arguments, a sentence consisting of one or more words separated by a space and a single-character string. This function returns a copy of the sentence string with every instance of the indicated letter removed. For example, `remove_letter("Hello World!", "e")` should return the string `"Hllo World!"`.

- Remember that a string is just a concatenation of individual characters using the `+` symbol and that Python allows you to iterate over the characters of a string using a `for` loop.

b. In the main part of the program, prompt the user for and read in a sentence (that may include several words separated by spaces) as a string.

c. Then, prompt for and read in a single-character string to remove from the sentence.

d. Call the `remove_letter()` function, passing in the sentence and single-character letter input by the user, assigning the result to a new string variable that will contain the copy of the sentence with all occurrences of the letter removed.

e. If no letter or letters were removed from the original sentence, then write an appropriate message, including the single-character and the original sentence in the message. Otherwise, write an appropriate message indicating all occurrences

of the letter were removed, including the single-character and the original sentence in the message.

For example, the output might look like this (input shown in **bold**):

```
$ python3 Lab8A.py
Enter a sentence: It looks like it's going to rain today!
Enter letter to remove from sentence: i
New sentence with letter i removed: It looks lke t's gong to ran today!
$ python3 Lab8A.py
Enter a sentence: It looks like it's going to rain today!
Enter letter to remove from sentence: q
No letter q removed from: It looks like it's going to rain today!
```

Note that you will submit this file to Canvas.

3. Working with Functions and Lists

Python strings have a built-in `replace()` method that can replace all occurrences of a word in a string with another word. But today, we're going to see how to implement this ourselves. Note that we are only replacing full words, not substrings or parts of words.

Write a small, but complete Python 3 program called **`Lab8B.py`** that does the following:

a. Create a user-defined function called `replace_word(sentence, target, replace)` that accepts three string arguments, a sentence consisting of one or more words separated by a space, a target word that will be replaced, and finally, a replacement word for the target word. This function returns a copy of the sentence string with every instance of the target word replaced by the replacement word. For example, `replace_word("I am happy to meet you!", "happy", "angry")` should return the string `"I am angry to meet you!"`. You do not have to worry about handling punctuation correctly.

- Instead of using the `replace()` method, you will iterate over the words in the sentence string. First, use the `split()` method to split your string into a list of words. Then iterate over the length of the list. This will allow you to get the index in the list so that you can replace the target word with the replacement word.
- You will return a copy of the string with all occurrences of the target word replaced by the replacement word. Use the `join()` method to reconstruct the list back into a sentence and return that string.

b. In the main part of the program, prompt the user for and read in a sentence (that may include several words separated by spaces) as a string.

c. Prompt for and read in a target word as a string to remove from that sentence.

d. Prompt for and read in a replacement word as a string to replace the target word in that sentence.

e. Call the `replace_word()` function, passing in the sentence, the target word, and the replacement word input by the user, assigning the result to a new string variable that will contain the copy of the sentence with all occurrences of the word removed.

f. If no word or words were replaced from the original sentence, then write an appropriate message that no changes were made. Otherwise, write an appropriate message that includes the new sentence.

g. Now, let's incorporate the `remove_letter()` function from the `Lab8A.py` file. To do this, we must add the `import Lab8A` line to the top of our file. Doing this will cause the main part of `Lab8A.py` to be executed before our code in `Lab8B.py`, so we must also modify the code in our `Lab8A.py` file to only execute if the `__name__` attribute is equal to `'__main__'`. Note that adding this `if` statement to our `Lab8A.py` file will still allow that lab file to work as expected for that lab component. You can look at our example in my public directory at the start of September for help on doing this.

h. Now, prompt for and read in a single-character string to remove from our newly modified sentence.

i. Call the `remove_letter()` function from the `Lab8A.py` file, passing in the already modified sentence and the single-character letter just now input by the user, assigning the result to a new string variable that will contain the copy of the sentence with all occurrences of the letter removed. Note that you will have to prefix `Lab8A` to the `remove_letter()` function to get it to work.

j. Finally, write an appropriate message indicating all occurrences of the letter were removed, including the single-character and the sentence in the message.

For example, the output might look like this (input shown in **bold**):

```
$ python3 Lab8B.py
Enter a sentence: It will be sunny this week!
Enter a word in sentence to replace: sunny
Enter a replacement word: rainy
Here is the new sentence: It will be rainy this week!
Enter letter to remove from sentence: y
New sentence with letter y removed: It will be rain this week!
$ python3 Lab8B.py
Enter a sentence: It will be sunny this week!
Enter a word in sentence to replace: nothing
Enter a replacement word: something
No changes were made to the original sentence
Enter letter to remove from sentence: i
New sentence with letter i removed: It wll be sunny ths week!
```

Note that you will submit this file to Canvas.

Now that you have completed this lab, it's time to turn in your results. Once you've moved the files to your windows machine (using **WinSCP**), you may use the browser to submit them to Canvas for the **Lab 08** dropbox.

You should submit the following files:

- **`Lab8A.py`**
- **`Lab8B.py`**
- **(Note that the zyBooks labs are submitted separately through Canvas.)**

Ask your TA to check your results before submission.

Now that you've finished the lab, use any additional time to practice writing simple programs out of the textbook, lectures, or even ones you come up with on your own to gain some more experience.