CS 121 – Lab #5 – String practice

This exercise will practice with some simple interactive string input, if statements and loops. We will look at these problems, each one accomplished by one short program.

- Checking to see if a word is of a particular length and contains a certain letter.
- Counting vowels in a string.
- A simple game in which the user must enter a word that begins with the same letter that the previous word ended with.

Create a new folder called lab05 on your USB drive or account, to store today's programs.

Part 1: Checking String input (check.py)

1. This program will ask the user to enter a string. But not just any string – we'd like the user to type in something that has 6 characters and contains a lowercase letter 't'. After the program reads the string from input, it will check to make sure the input satisfies these 2 conditions. Which built-in functions would you need to use in this case?

   Note that the program should be divided into 3 parts:

   - Prompting the user and obtaining the string value
   - Checking the length of the string, and complaining if it's not correct
   - Checking to see if the string contains a 't'. If not, print an error message; if so, tell where we found the 't'.

2. Run your program a few times to make sure that it works under the various possible input cases. For example, I've run the program 4 times below. Do you know why we should run this program at least 4 times?

```
Please enter a string with 6 characters, containing the letter 't':  peanut

There is a 't' located at position 5.
```

```
Please enter a string with 6 characters, containing the letter 't':  cat

Sorry, your string has a length of 3.
There is a 't' located at position 2.
```

```
Please enter a string with 6 characters, containing the letter 't':  hello

Sorry, your string has a length of 5.
Your string does not contain a 't' !
```

```
Please enter a string with 6 characters, containing the letter 't':  hockey

Your string does not contain a 't' !
```

Part 2: Counting vowels (vowel.py)

3. For this problem, we want to read a string to see how many vowels it has. You will create a new program to do this.

   In this program, we'll be asking the user to enter a string. This string could be a word or a phrase. Let's assume that the string will be just one line of text. We need to inspect each character in this string to see if it is a vowel: a, e, i, o, u. Note that the vowels could be capital or lowercase letters. In Python, we have string functions to turn words into all capital or all lowercase letters, and this may simplify our comparisons.

4. Write a detailed comment in your source file explaining your approach.

5. Finally, type in the necessary code to complete the implementation. Feel free to ask for help if you get stuck or if you think you are not doing this correctly.

6. Save and run the program. Is the output correct? If so, please make one more modification. If the number of vowels is 1, then your output should say "1 vowel" and not "1 vowels".   √

Part 3: A word game (game.py)

Finally, let's create a little game. First, the user will type in any word. Then, to continue the game, the user will need to enter a new word that starts with the same letter that the previous word ended with. We can cycle through several words as follows: "horse", "elephant", "tiger", "rabbit", "telescope", "everybody", "yogurt", etc.

Let's set up the game so that the user can type in up to 5 words. Each time the user enters a word that is not correct, print an error message and continue.

7. Here are some details you need to address:

   a. Are you going to use a while loop, or a for loop? How would you set the condition so that we do exactly 5 iterations of the game?

   b. How do we extract the last character of the word that has already been entered?

   c. The next thing the program does is read in a new word. How can we tell what the first character is, and that it matches the letter we want?

   d. Finally, if you are using a while loop, don't forget that at the end of the loop, you probably need to increment a count variable. √

8. Create a second version of the game. Copy the source code to a new file called game2.py. Now, modify the program so that allows the user to play the game indefinitely, but will stop as soon as the user enters an incorrect word. Hint: one way to approach this is to read the first 2 input words before the while loop begins. Because this is a significant modification to the program, we need to think about our design, before typing code. Please write your sequence of steps as a comment in your source file, and have the instructor check your work. √