LAB #7
More Functions and Using Strings

At this time, you may pair with someone in the lab, and finish the rest of the lab as a pair.

(5 pts) String Program – Design First

Design a program to **take two strings as input**, and you will compare the two strings, character by character. You need to **count the number of occurrences where the characters in the two strings match**, and you will **output the percentage of matching characters** between the two strings.

If the two strings are not the same length, then they are not matching for those extra characters. **For example**, Jennifer and Jen match on 37.5% of the characters.

Your program must have a **main()** function, and you must have these functions: **get_user_input()**, **percent_matching(),** and **num_matching_chars().**

Begin by designing the interaction and functionality for the 4 functions above.

- How will these functions interact together, i.e. who will call who?
- What are the parameters for these functions?
- What are the pre and post conditions for these parameters?
- What are the return values for the functions?
- Now design the steps needed in each function?

(5 pts) Implementation Next

Now, implement you design! You must have defined and called the functions above, but you can have more. Make sure each function has a function header (a block of comments describing the function), including the above information. Comments in python are proceeded by the # symbol. This tells the interpreter to ignore the following text on that line. For example:

```
################################################
# Function Name: main
# Description:  Begin execution for program here
# Parameters:  None
# Pre-conditions:  None
# Post-conditions:  None
# Return values:  None
################################################
```

Look at the style guideline for our class, and make sure your program adheres to our style guideline for this class: http://classes.engr.oregonstate.edu/eecs/fall2016/cs160-

001/160_style_guideline.pdf. You should write a program header, as well as all your function headers.

Now, exchange your design with a different group in the lab/class.

- How does your design differ from theirs?
- How is your design similar?
- Try implementing their design.
    - Did they provide enough detail for the interaction, pre and post conditions, and return values?
    - Did they provide enough detail for the steps needed in the function?

Extended Learning:

Since strings are not mutable in Python, make a two list/array of characters to pass to get_user_input(s, s2). Now, instead of calling get_user_input() two times, call it once and change the contents in the function.