6Question

**Objective:**

The purpose of this lab is for you to become familiar with Python’s built-in text container -- class **str**-- and lists containing multiple strings.  One of the advantages of the **str** class is that, to the programmer, strings in your code may be treated in a manner that is similar to how numbers are treated.  Just like **int**s, **float**s, a *string* object (i.e., variable) may be initialized with a literal value or the contents of another string object. String objects may also be added together (concatenation) and multiplied by an integer (replication). Strings may also be compared for “equality” and arranged into some order (e.g., alphabetical order, ordering by number of characters, etc.).  Finally, strings may be placed in containers which can then be passed as arguments to functions for more complex manipulations.

**Specifications:**

Write an interactive Python program composed of several functions that manipulate strings in different ways.  Your main() function prompts the user for a series of strings which are placed into a list container.  The user should be able to input as many strings as they choose (i.e., a sentinel-controlled loop).  Your main function will then pass this list of strings to a variety of functions for manipulation (see below).

The main logic of your program must be included within a loop that repeats until the user decides he/she does not want to continue processing lists of strings.  The pseudo code for the body of your main() function might be something like this:

# Create the main function  
  
def main():  
    # declare any necessary variable(s)  
  
  
    # // Loop: while the user wants to continue processing more lists of words  
    #       
    #     // Loop: while the user want to enter more words (minimum of 8)  
    #         // Prompt for, input and store a word (string) into a list  
    #         // Pass the list of words to following functions, and perform the manipulations

    #         //  to produce and return a new, modified, copy of the list.  
    #         //  NOTE: None of the following functions can change the list parameter it  
    #         //  receives – the manipulated items must be returned as a new list.  
    #   
      # // SortByIncreasingLength(…)  
      # // SortByDecreasingLength(…)  
      # // SortByTheMostVowels(…)  
      # // SortByTheLeastVowels(…)  
      # // CapitalizeEveryOtherCharacter(…)  
      # // ReverseWordOrdering(…)  
      # // FoldWordsOnMiddleOfList(…)  
    #     // Display the contents of the modified lists of words  
    #   
    # // Ask if the user wants to process another list of words