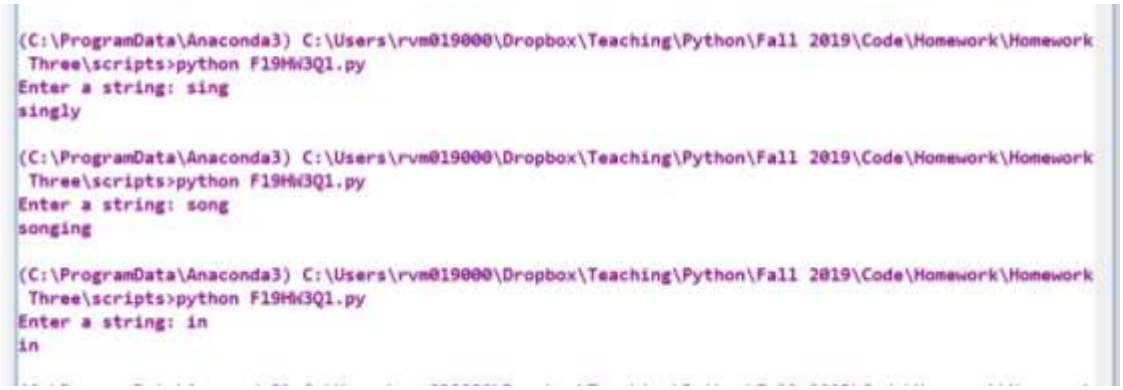
MIS 6382
Object Oriented Programming With Python
Fall 2019
Homework Three

The following guidelines should be followed and will be used to grade your homework:

- The code for each question should be implemented using Jupiter notebooks.
- All the code should be included in one single notebook (.ipynb file).
- This is an individual homework assignment; no group submissions will be accepted.
- Sample runs shown in the question should be used as a guide for implementation. However extensive testing needs to be done on your code to deal with all test cases that might possibly be executed.
- **Submit a zipped folder containing ONLY one .ipynb file for this assignment. The zip file should be named using your name and the chars "hw3". You will be penalized 15% of the grade if your submission does not follow these requirements.**
- **You will get zero points if your program has syntax errors.**

1. Write a Python program to add 'ing' at the end of a given string (length should be at least 3). If the given string already ends with 'ing' then add 'ly' instead. If the string length of the given string is less than 3, leave it unchanged.
   Here is a sample interaction



2. Write a Python function called has_sublist (), that accepts a list as input and checks if any of its elements are themselves lists.
   For example, the two calls to the above function with [5,4,[3,8]] and [5,4,3,8] respectively will produce the following output:

3. Write a Python function concat_list() that accepts a list containing string values and an integer as input and prints out a new list where each element in the list is concatenated with all integers from 1 to n.
   For example, concat_list(['a','x','d'], 5) will produce the following output:

```
['a1', 'a2', 'a3', 'a4', 'a5', 'x1', 'x2', 'x3', 'x4', 'x5', 'd1', 'd2', 'd3', 'd4', 'd5']
```

4. Write a Python program that accepts a comma separated sequence of words as input and prints the unique words in sorted form (alphanumerically)
   Here is a sample interaction:

```
(C:\ProgramData\Anaconda3) C:\Users\rvm019000\Dropbox\Teaching\Python\Fall 2019\Code\Homework\Homework
 Three\scripts>python F19HW3Q4.py
Enter a sequence of words separated by commas hello,how,are,you,you,look,well
['are', 'hello', 'how', 'look', 'well', 'you']
```

5. Write a Python program to create a dictionary from a string by tracking the count of characters in a string. Both upper case and lower-case alphabets should be treated as one. Do not count anything but letters of the alphabet. That is, do not count spaces, or any other special characters.
   Here is a sample interaction:

```
(base) C:\Users\radha\Dropbox\Teaching\Python\Fall 2019\Code\Homework\Homework Three\scripts>python F19HW3Q5.py
Enter a string: Hello, Who are you?
{'H': 2, 'E': 2, 'L': 2, 'O': 3, 'W': 1, 'A': 1, 'R': 1, 'Y': 1, 'U': 1}
```

6. Write a Python function called compare_seq() that accepts two sequences (seq1 and seq2) of numbers and prints out the following:
   - All elements that occur in seq1 but not in seq2
   - All elements that occur in seq2 but not in seq1
   - All elements in the two sequences combined. This should contain only unique occurrences
   - All elements that are present in both sequences
   - Finally:
     - print "Equal" if seq1 and seq2 are exactly the same
     - print "Greater" if all elements in seq2 are in seq1 but not all elements in seq1 are in seq2
     - print "Lesser" if all elements in seq1 are in se2 but not all elements in seq2 are in seq1
     - Otherwise, print "Neither" if not all elements in seq1 are in seq2 AND not all elements in seq2 are in seq1

- Note that your results should all print out as lists, but your input to the function can be any sequence (tuples, lists or sets).

For example, given two sequences ("f", "e", "d", "c", "b") and ("a", "b", "c"), the output will be as follows.

```
(base) C:\Users\radha\Dropbox\Teaching\Python\Fall 2019\Code\Homework\Homework Three\scripts>python F19HW3Q6.py
All elements that occur in seq1 but not in seq2 are: ['e', 'd', 'f']
All elements that occur in seq2 but not in seq1 are: ['a']
All elements in the two sequences combined are: ['e', 'd', 'b', 'a', 'c', 'f']
All elements that occur in both sequnces are: ['b', 'c']
Neither
```