Notes

- You will need the "wordlist.txt" file in the same folder as your programs for this lab.
- Download the Lab_9_1 source code file. Download the Lab_9_2 source code file.
- You can use either PyScripter or IDLE for this lab.
- You will be uploading two programs to Sakai as Assignments when you complete this lab. The files should be named LastName_Initial_Lab_9_1.py and LastName_Initial_Lab_9_2.py.

Part I

- Open Lab_9_1.py. When this program is complete it will allow the user to practice solving anagrams. When you download it, the program will do everything except actually generate an anagram. That's your job.
- The program contains an incomplete function named "scramble". The intent of this function is to take a word (stored in a string) passed as a parameter and return a scrambled or randomized version of the string. Your job is to make it work correctly. You don't need to make any changes to the program outside of the function named scramble. Some of the following facts may be useful:
  - `''.join(list of strings)` takes a list of strings and returns a single string
  - `list( string )` takes a string and returns a list of the characters in the string
  - `alist.sort()` sorts the elements of a list named alist
  - `x in alist` and `x not in alist` are simple ways to determine if a value x is or is not in a list named alist.
  - `str.split()` where str is a string, returns a list of the individual "words" in str, assuming that "words" are separated by one or more blanks
  - `random.shuffle( alist )` a mutator function that randomly shuffles the elements of a list named alist
  - `random.randint(a,b)` returns an integer in the range [a..b]
- TEST YOUR FUNCTION USING PRINT STATEMENTS. For example, you should be able to put the statement `print( scramble("eggs") )` in your program to see what is printed.

Scoring Rubric

- 5 points – Function correctly scrambles a string
- 3 points – Function is well documented, including a triple-quoted (""") comment describing the purpose and operation of the function.

Part II

- Before you do anything on this part you need to develop an algorithm to solve the following problem: Given a scrambled word (anagram) and a long list of words, how can you find a word in the list that is a rearrangement of the letters in the anagram. Discuss your algorithm with another student or instructor. Your algorithm should list the steps you are going to take in order to solve the problem. Don't just start writing Python code. If you do, you will never finish this program!

- When you have an algorithm in mind, work in small steps. Write a few Python statements or a short function and test that code. Then do a few more steps and some testing. Repeat until finished. Note that some of the facts listed in Part I will also be useful here. However, there's nothing about this part of the lab that requires random values.
- Open Lab_9_2.py. When this program is complete it will allow the user to type in an anagram and it will "solve" the anagram, if possible. When you download it, the program will not do much except read a list of words from a file and prompt the user to enter a scrambled word. Your job is to complete the program.
- Some scrambled words will not correspond to any word in the word list. When this happens, make sure that your program prints an appropriate message. If a "solution" is found, that solution should be displayed for the user.
- Enjoy the challenge.

Scoring Rubric

- 9 points – The program correctly determines what word in the word list corresponds to the scrambled word or reports that no solution was found.
- 3 points – The program is well-documented.

Upload your two programs to Sakai as Assignments when you complete this lab. The files should be named LastName_Initial_Lab_9_1.py and LastName_Initial_Lab_9_2.py.