**Group HW: Pig Latin Translator**

For this problem, you will write a program to perform English to Pig Latin translation. You will also need to do some work with files and dates.

**‘Pig Latin’ Translation rules**

1. If a word has no letters, don't translate it.

2. All punctuation should be preserved.

3. If the word begins with a capital letter, then the translated word should too.

4. Separate each word into two parts. The first part is called the ***prefix***and extends from the beginning of the word up to, but not including, the first vowel. (The letter *y* will be considered a vowel for this.) The Rest of the word is called the ***stem***.

5. The Pig Latin text is formed by reversing the order of the ***prefix***and ***stem***and adding the letters ‘*ay’* to the end. For example, *sandwich* is composed of *s* + *andwich* which would translate to *andwichsay.*

6. If the word contains no consonants or starts with a vowel, let the **prefix** be empty and the word be the **stem**. The word ending should be *yay* instead of merely *ay*. For example, *I* would be *Iyay* and *understood* would be *understoodyay*.

**Phase 1 (10 points)**

Your first task is to produce a function called **pl\_prefix** that takes one argument, a string containing a word, and returns the portion of the word up to, but not including the first vowel. For example, if you sent 'banana' to your function, it should return 'b'. Sending 'Sibley' should return 'S', 'stream' should return 'str', and 'break' should return 'br'. Test your function with other words and print out the results.

**Phase 2 (10 points)**

Using what you learned from Phase 1, write a function called **pl\_reverse** that takes a single word as an argument and returns the word with the prefix and stem reversed.

**Phase 3 (30 points)**

Using what you learned from Phase 2, write a function called **pl\_translate** that properly handles adding the *ay* and *yay* word endings, capital letters, and punctuation.

**Phase 4 (15 points)**

Using the function from Phase 3, write a function called **pl\_file** that takes a string with an input file name and a string with an output file name and translates all the words in the input file according to the rules, then outputs this into the output file. Catch exceptions properly for files not existing, etc.

**Phase 5 (20 points)**

Have your program give the user a listing of all the text files (\*.txt) in the current directory (cwd). Allow the user to specify which file they want to translate. Have your program save the output file in a directory called “translations” inside the current directory. (You can assume that directory exists, which means you may need to create it manually to test your code) Save the output file as [input file name] (pig).txt (So if the user chose to translate **novel.txt** you would save it as <cwd**>/translations/novel(pig).txt**). Make sure your program works on the Burrow UNIX environment as well!

**Phase 6 (15 points)**

Print the current day and time at the top of the output file. Print “Thank you for using the Pig Latin Translator” at the end of the output file.

**Submission**

Upload your program (.py) to Oncourse under Assignments -> Assignment 3 (Group) as a ***.py file***. Name your file as ***YourUsername\_A3\_group.py***

e.g. If I were to upload, it would be johfdunc\_A3\_group.py

Include the following information as a comment at the top of the file you submit:

 Your name

 Your group number

Make sure you submit a copy of the I211 Homework Team Feedback Form (found on Oncourse). You must submit a copy of this form with every homework or lose 50% of your grade!

**Each member of the group must submit their own copy of the homework!**

Test your code thoroughly!