Computer Science 4140
Assignment 2

In this assignment you will complete two different programs. Details are provided below.

Program 1

The following rules define the translation of an English word into *pig Latin*:

1. If the word begins with a nonempty string of consonants, move the initial consonant string to the back of the word and add the suffix **AY**; e.g, **pig** becomes **igpay**.
2. If the word begins with a vowel, add the suffix **YAY**; e.g., **owl** becomes **owlyay**.
3. **U** following **Q** is a consonant.
4. **Y** at the beginning of a word is a vowel if it is not followed by a vowel.
5. One-letter words are not changed.

Write a Python program that takes in a raw text file, reads it in line by line, breaks each line into tokens, and then converts each token that consists only of letters (we will call these tokens *words*) into *pig Latin*. Tokens that have non alphabetic characters should simply be output as is. Further details are provided below.

- A starter template for the file *piglatin.py* is provided. It contains the code for processing the text file. The name of the file is supplied on the command line. Thus, a sample execution of the program would be the following command line.

  python3 piglatin.py input.txt

- The code you write will actually just be the code needed to implement the function *wordmap(token)* that takes a token and returns the token transformed according to the rules given.

- In addition to the definition of the *wordmap* function, you may wish to initialize some global strings that can help in constructing regular expressions to be used in pattern detection.

- Because you are only interesting in looking at patterns that start at the beginning of the string, you may find it simplest to use *re.findall(...)* for pattern matching. It returns a list of strings that match the supplied pattern. The matched string will always be at index [0], the first match (if it exists).

Program 2

Implement the American Soundex encoding algorithm described in Wikipedia at the following location

https://en.wikipedia.org/wiki/Soundex

This program will use the same input conditions as the first program. Tokens that have non alphabetic characters should be output as is. Tokens that consist only of letters should be transformed according to the algorithm. Further details are the following.

- A starter template for the file *soundex.py* is provided. Just like the template for the first program, it contains the code for processing the text file. The name of the file is supplied on the command line. Thus, a sample execution of the program would be the following command line.

  python3 soundex.py input.txt

- The code you write will actually just be the code needed to implement the function *wordmap(token)* that takes a token and returns the token transformed according to the rules given.

- In addition to the definition of the *wordmap* function, you may wish to initialize some global strings that can help in constructing regular expressions to be used in pattern detection.

- An elegant approach for handling the mapping of consonants to digits is to use a dictionary to define the mappings. I would suggest you create a list of tuples where each tuple is of the following form *(<list of letters>, <char>)* (e.g. *(['d','t'],'3')*). Use a nested loop to build the dictionary from the tuple list. Specifying the mapping as a list of tuples is convenient for readability.

Program Deadlines

February 20, 11:00 P.M.

Deadline for submitting completed programs. The *<assignment_number>* should be *hw2* for both programs.

Program Evaluation

The assignment is worth 50 points (25 points for each program). Your grade on this assignment will be weighted as follows:

Code Correctness	70%
Coding Style and Documentation	30%