CSE 465/565
Spring 2023
Homework #9
100 points

Submit your code and PDF file in a zipped-up folder. For each exercise, you will see the points in parenthesis.

1. (15 p) Variadic methods (i.e., methods with a variable number of parameters) in Python, Java and C#
Look at the programs below in Python, Java, and C#:

```
def adder(***TODO***):
      sum = 0
      ***TODO***
      return sum

print("Sum 1 is", adder(1,2,3,4,5,6))
print("Sum 2 is", adder(1,5))
print("Sum 3 is", adder())
```

```
public class hw9 {
      public static int adder(***TODO***)   {
            int sum = 0;
            ***TODO***
            return sum;
      }
      public static void main(String [] args) {
            System.out.println("Sum 1 is " + adder(1,2,3,4,5,6));
            System.out.println("Sum 2 is " + adder(1,5));
            System.out.println("Sum 3 is " + adder());
      }
}
```

```
using System;
namespace hw9 {
      class hw9 {
            public static int Adder(***TODO***)
            {
                  int sum = 0;
                  ***TODO***
                  return sum;
            }

            public static int Main(String [] args)
            {
                  Console.WriteLine("Sum 1 is {0}", Adder(1,2,3,4,5,6));
                  Console.WriteLine("Sum 2 is {0}", Adder(1,5));
                  Console.WriteLine("Sum 3 is {0}", Adder());
                  return 0;
            }
      }
}
```

Replace the text `***TODO***` in the three programs above in a PDF file *hw9.pdf* that you will submit with the appropriate code, so that the methods `adder/ Adder` accept a variable number of parameters. The programs are also available to you in the files *hw9.py*, *hw9.java*, and *hw9.cs*.

The expected output for all three programs is:

```
Sum 1 is 21
Sum 2 is 6
Sum 3 is 0
```

2. (85 p) Complete piglatin.py.

You will have to implement a function **ToPigLatin(l)** that takes as an argument a list of characters l representing one word (e.g., the list ['a', 'p', 'p', 'l', 'e'] for the string “apple”) and returns a list of characters representing the Pig Latin version of the original word, according to the instructions below. The list l also has to be changed in place to the Pig Latin version of the original word.

- For our purposes, we will use the following as the rules for translation of a word into "Pig Latin":
  1. A **word** is a consecutive sequence of letters (a-z, A-Z) or apostrophes. You may assume that the input to the function will only be a single "word". Examples: Zebra, doesn't, apple.
  2. If a word starts with a vowel, the Pig Latin version is the original word with "way" added to the end.
  3. If a word starts with a consonant, or a series of consecutive consonants, the Pig Latin version transfers all consonants up to the first vowel to the **end** of the word, and adds "ay" to the end.
  4. The letter 'y' and the letter 'w' should be treated as consonants if they are the first letters of a word, but treated as vowels otherwise.
  5. If the original word is capitalized, the new Pig Latin version of the word should be capitalized in the first letter (i.e. the previous capital letter may not be capitalized any more).
- In the starter file **piglatin.py**, there is a main program that is already provided to you for testing. This program prompts the user to type in 5 words, then it calls the function **ToPigLatin** (which you will implement) for each of these words and prints the converted version of each of the 5 strings. Note that the main program does all of the output -- your function should do NO output (just the appropriate conversion and return). Do not change the main program in any way.
- You may assume that a "word" to be entered into the function is no more than 39 characters long.

Sample Runs (user input is underlined, to distinguish it from output):

Sample Run 1:

```
Input 5 words: Flower yellow bypass apple Igloo

Pig Latin version of the 5 words:

Owerflay ellowyay ypassbay appleway Iglooway
```

Sample Run 2:

```
Input 5 words: watch Hamburger Rhythm queen zipper

Pig Latin version of the 5 words:

atchway Amburgerhay Ythmrhay ueenqay ipperzay
```

Hints:

- The following Python functions may be useful:
  - `isupper()` and `upper()` for characters
  - `append()` and `extend()` for lists
- Changes made to a list `lst` passed as a parameter in a function call are visible once the function finishes, if changes are made using `append()`, `extend()`, or by modifying the value of a list element as in `lst[i] = new_val`
- Changes made to a list `lst` passed as a parameter in a function call are **not** visible once the function finishes, if the function changes the whole list as in `lst = new_list`