Specification for Assignment 1

Your submission *for this assignment* ***must be commented*** *and* ***must include both your name*** *and your* ***student number*** *as a comment at the top of* ***every source file you submit****. Each of your submitted files must use a* ***file name beginning 'comp3007_w23_#########_assignment_01'*** *(replacing the number signs with your own student number) and any* ***submissions that crash*** *(i.e., terminate with an error) on execution will automatically* ***receive a mark of 0****.*

Officially, the Due Date for this Assignment is:

Friday, January 27th, 2023*, at* ***11:59pm EST****.*

Late Submissions are Accepted Without Penalty Until Sunday, January 29th, by 11:59pm EST.
Submissions received after that will not be accepted and will receive a mark of 0.

The objective of this assignment is to have you revisit the paradigm shift from the first lecture, where you focus your efforts, not upon describing "how" a result might be computed, but instead upon "what" each component is. For this assignment you will design and implement a program in Python 3 that uses several recursive algorithms to produce a string of words in "Alfabeto Farfallino" from a string in "Snake Case".

A string that is written in "Snake Case" uses only lowercase letters and each word is separated by an underscore (e.g., "robert_collier"). Any word can be written in "Alfabeto Farfallino" **(as it is defined here)** by splitting it into multiple words after each vowel and then doubling each vowel and inserting the letter 'f' between them. For this assignment, we operationally define vowel as any character 'a', 'e', 'i', 'o', or 'u', but not 'y'. As a clarifying example, the word "robert" contains two vowel sounds "o" and "e" and so would be written in "Alfabeto Farfallino" as "rofo befe rt", and the word "collier" contains the three vowels "o", "i", and "e", and so would be written as "cofo llifi efe r".

The conversion of a string in "Snake Case" into "Alfabeto Farfallino" that you do for this assignment requires transforming an input "Snake Case" string into a list of words, and then transforming each word using the rules above. As a clarifying example, the "Snake Case" string "robert-is-forty-two" must first be parsed into the list of words ["robert", "is", "forty", "two"] and then each of those must be split and converted into "Alfabeto Farfallino" word chunks and finally combined to produce the string "rofo befe rt ifi s fofo rty twofo".

For this assignment:

- you must write **three recursive functions**: the first for converting a "Snake Case" string argument into a list of strings return (i.e., the strings separated by underscores), the second for converting a single string argument of one word into **a new string containing multiple word chunks** in "Alfabeto Farfallino" return, and the last for actually applying your second function to a list argument (presumably the list that is returned by your first function) and returning the final string.
- these functions must be RECURSIVE and must mirror the generic recursive design pattern presented. They may NOT use looping control structures, and **nonrecursive functions will receive zero**
- you may use string contatenation and the indexing and slicing operators, but you must NOT import any libraries or use anything that would circumvent the need for creating recursive definitions (e.g., list comprehensions)
- you must include a simple interface in a main function that will allow the teaching assistants to test your code by running it against a valid "Snake Case" string that they will provide as a command-line argument and examining the corresponding "Alfabeto Farfallino" string that you provide as output