**I created the following modules for a homework assignment:**

## def nestedRemoval(text: str, leftPattern: str, rightPattern: str) -> str

You will be given three strings, the first string is a sentence known as text that contains at least one occurrence of leftPattern and rightPattern. Left pattern and right pattern are single characters strings that you are attempting to remove from the string. Your goal is to remove only the patterns when left pattern has an equal number of corresponding right patterns. The patterns are removed in pairs only when there is a balanced number of left and right patterns. The patterns may be nested within the text and there is no guarantee that a left pattern will occur before the right pattern.

Example 1:

text = { { Muscat } } { } mecum tollgate } poultry quarrymen pantheon asteria

leftPattern = {

rightPattern = }

return = Muscat mecum tollgate } poultry quarrymen pantheon asteria

Example 2:

text = theretofore [ ] [ ] demography ] ] pirouetting morsel [ [ pesticide

leftPattern = [

rightPattern = ]

return = theretofore demography ] ] pirouetting morsel [ [ pesticide

Example 3:

text = ( castigate ) alfonso ( ) ) ) emitter sourdough ) taco ( schemata

leftPattern = (

rightPattern = )

return = castigate alfonso ) ) emitter sourdough ) taco ( schemata

## Documentation

Each function will be tested to see if it is documented. All of the functions mentioned above should have documentation that is at least 50 characters long. This is a test to make sure you understand how to document your code. The actual content within the string is not going to be tested, feel free to write the lyrics of your favorite song as the documentation. You do not need to write an additional function for this test. Below is an example of a function that has been documented.

def my\_function(x, y):

"""You documentation goes right here"""

# your functional code goes here.