"""Question 1: count\_letters"""

def count\_letters(s, letter):

    return

"""Testing Question 1 implementation"""

def test\_count\_letters():

    print("Testing count\_letters()...", end="")

    assert(count\_letters("Hello", "l") == 2)

    assert(count\_letters("apple", "a") == 1)

    assert(count\_letters("Hi", "h") == 0) *# case sensitive!*

    assert(count\_letters("More letters", "e") == 3)

    print("...done!")

"""Question 2: digit\_sum"""

def digit\_sum(n):

    return

"""Testing Question 2 implementation"""

def test\_digit\_sum():

    print("Testing digit\_sum()...", end="")

    assert(digit\_sum(3) == 3)

    assert(digit\_sum(16) == 7)

    assert(digit\_sum(1234) == 10)

    assert(digit\_sum(0) == 0)

    assert(digit\_sum(-35) == 8)

    print("... done!")

"""Question 3: every\_other"""

def every\_other(s, start):

    return

"""Testing Question 3 implementation"""

def test\_every\_other():

    print("Testing every\_other()...", end="")

    assert(every\_other("Hello", 0) == "Hlo")

    assert(every\_other("blueberries", 3) == "eere")

    assert(every\_other("banana", 1) == "aaa")

    assert(every\_other("apple", 4) == "e")

    assert(every\_other("eat", 4) == "")

    print("... done!")

"""Question 4: alternating\_sum"""

def alternating\_sum(L):

    return

"""Testing Question 4 implementation"""

def test\_alternating\_sum():

    print("Testing alternating\_sum()...", end="")

    assert(alternating\_sum([7]) == 7)

    assert(alternating\_sum([1, 2]) == -1)

    assert(alternating\_sum([1, 3, 1]) == -1)

    assert(alternating\_sum([3, 2, 1]) == 2)

    assert(alternating\_sum([8, 0, 0, 0]) == 8)

    assert(alternating\_sum([5, 5, 5, 5]) == 0)

    print("... done!")

"""Question 5: has\_property\_309"""

def has\_property\_309(n):

    return

"""Testing Question 5 implementation"""

def test\_has\_property\_309():

    print("Testing has\_property\_309()...", end="")

    assert(has\_property\_309(309) == True)

    assert(has\_property\_309(371) == False)

    assert(has\_property\_309(575) == True)

    assert(has\_property\_309(17) == False)

    assert(has\_property\_309(0) == False)

    assert(has\_property\_309(462) == True)

    print("... done!")

"""Question 6: nth\_property\_309"""

def nth\_property\_309(n):

    return

"""Testing Question 6 implementation"""

def test\_nth\_property\_309():

    print("Testing nth\_property\_309()...", end="")

    assert(nth\_property\_309(0) == 309)

    assert(nth\_property\_309(1) == 418)

    assert(nth\_property\_309(2) == 462)

    assert(nth\_property\_309(6) == 662)

    assert(nth\_property\_309(9) == 713)

    print("... done!")

"""Question 7: Fix the Code"""

*# Remove the triple quotes from the following block of code to run this!*

'''

def factorial(n):

    counter = 0

    total = n

    while counter <= n:

        total \*= (n-counter)

    return total

def test\_factorial():

    print("Testing factorial()...", end="")

    assert(factorial(3) == 6)

    assert(factorial(5) == 120)

    assert(factorial(1) == 1)

    assert(factorial(8) == 40320)

    print("...done!")

'''

if \_\_name\_\_ == '\_\_main\_\_':

    test\_count\_letters()

    test\_digit\_sum()

    test\_every\_other()

    test\_alternating\_sum()

    test\_has\_property\_309()

    test\_nth\_property\_309()

*# test\_factorial()         # uncomment for Q7*