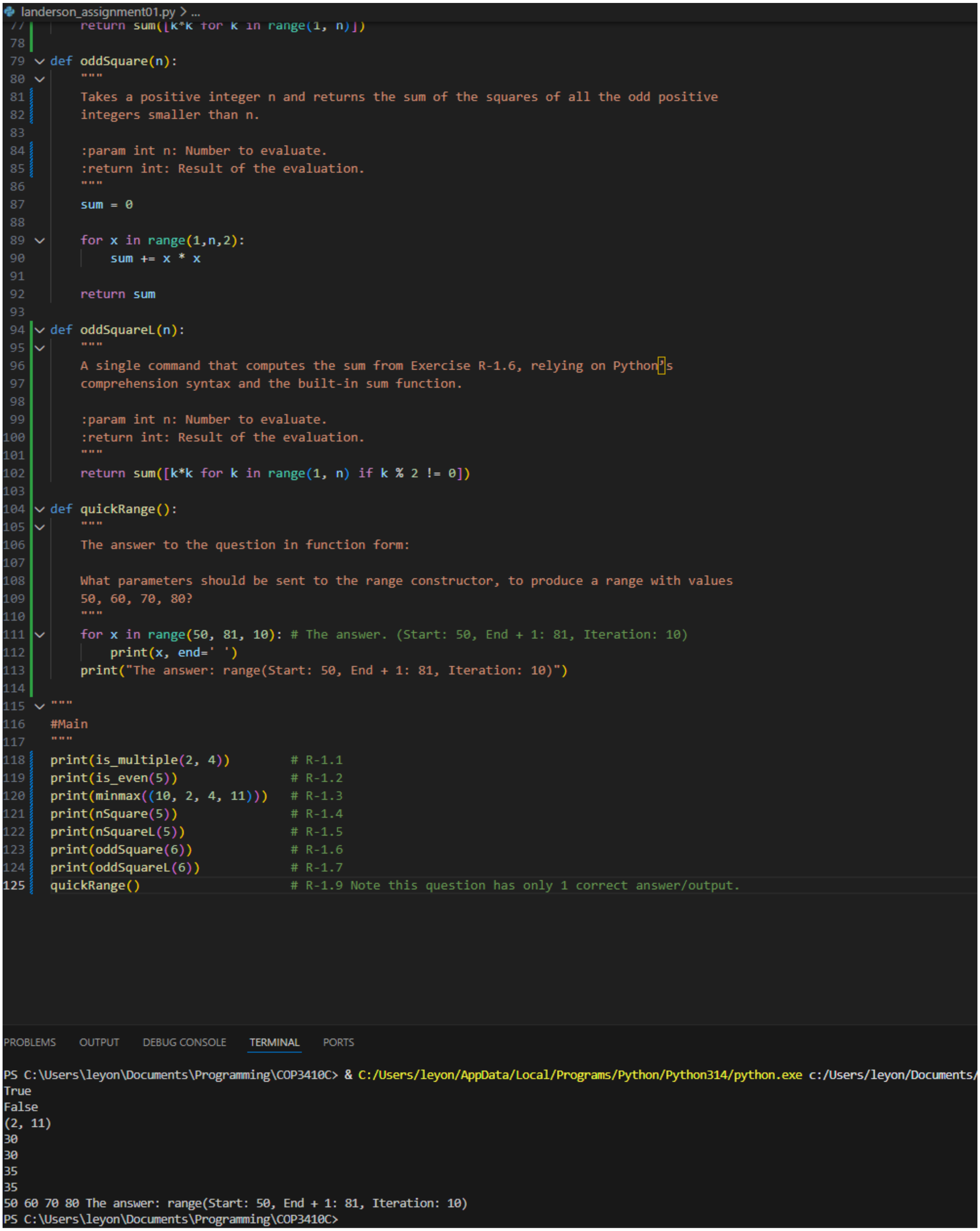
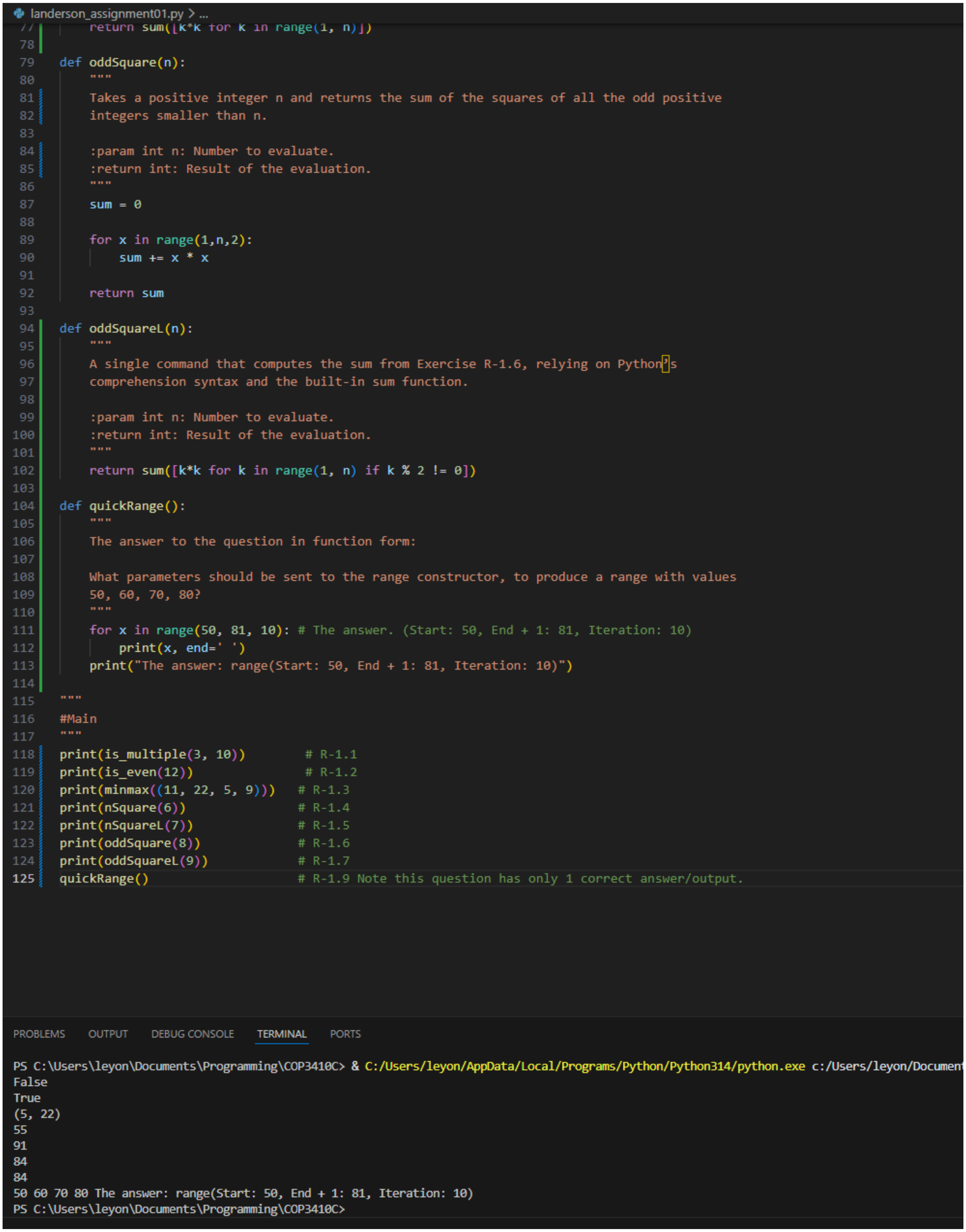
Output #1



```python
    return sum([k*k for k in range(1, n)])

def oddSquare(n):
    """
    Takes a positive integer n and returns the sum of the squares of all the odd positive
    integers smaller than n.

    :param int n: Number to evaluate.
    :return int: Result of the evaluation.
    """
    sum = 0

    for x in range(1,n,2):
        sum += x * x

    return sum

def oddSquareL(n):
    """
    A single command that computes the sum from Exercise R-1.6, relying on Python’s
    comprehension syntax and the built-in sum function.

    :param int n: Number to evaluate.
    :return int: Result of the evaluation.
    """
    return sum([k*k for k in range(1, n) if k % 2 != 0])

def quickRange():
    """
    The answer to the question in function form:

    What parameters should be sent to the range constructor, to produce a range with values
    50, 60, 70, 80?
    """
    for x in range(50, 81, 10): # The answer. (Start: 50, End + 1: 81, Iteration: 10)
        print(x, end=' ')
    print("The answer: range(Start: 50, End + 1: 81, Iteration: 10)")

"""
#Main
"""
print(is_multiple(2, 4))        # R-1.1
print(is_even(5))               # R-1.2
print(minmax((10, 2, 4, 11)))   # R-1.3
print(nSquare(5))               # R-1.4
print(nSquareL(5))              # R-1.5
print(oddSquare(6))             # R-1.6
print(oddSquareL(6))            # R-1.7
quickRange()                    # R-1.9 Note this question has only 1 correct answer/output.
```

```
PS C:\Users\leyon\Documents\Programming\COP3410C> & C:/Users/leyon/AppData/Local/Programs/Python/Python314/python.exe c:/Users/leyon/Documents/
True
False
(2, 11)
30
30
35
35
50 60 70 80 The answer: range(Start: 50, End + 1: 81, Iteration: 10)
PS C:\Users\leyon\Documents\Programming\COP3410C>
```

Output #2



```python
    return sum([k*k for k in range(1, n)])

def oddSquare(n):
    """
    Takes a positive integer n and returns the sum of the squares of all the odd positive
    integers smaller than n.

    :param int n: Number to evaluate.
    :return int: Result of the evaluation.
    """
    sum = 0

    for x in range(1,n,2):
        sum += x * x

    return sum

def oddSquareL(n):
    """
    A single command that computes the sum from Exercise R-1.6, relying on Python’s
    comprehension syntax and the built-in sum function.

    :param int n: Number to evaluate.
    :return int: Result of the evaluation.
    """
    return sum([k*k for k in range(1, n) if k % 2 != 0])

def quickRange():
    """
    The answer to the question in function form:

    What parameters should be sent to the range constructor, to produce a range with values
    50, 60, 70, 80?
    """
    for x in range(50, 81, 10): # The answer. (Start: 50, End + 1: 81, Iteration: 10)
        print(x, end=' ')
    print("The answer: range(Start: 50, End + 1: 81, Iteration: 10)")

"""
#Main
"""
print(is_multiple(3, 10))        # R-1.1
print(is_even(12))               # R-1.2
print(minmax((11, 22, 5, 9)))   # R-1.3
print(nSquare(6))               # R-1.4
print(nSquareL(7))              # R-1.5
print(oddSquare(8))             # R-1.6
print(oddSquareL(9))            # R-1.7
quickRange()                    # R-1.9 Note this question has only 1 correct answer/output.
```

```
PS C:\Users\leyon\Documents\Programming\COP3410C> & C:/Users/leyon/AppData/Local/Programs/Python/Python314/python.exe c:/Users/leyon/Documen
False
True
(5, 22)
55
91
84
84
50 60 70 80 The answer: range(Start: 50, End + 1: 81, Iteration: 10)
PS C:\Users\leyon\Documents\Programming\COP3410C>
```