

## Output #1

```
landerson_assignment01.py > ...
// |     return sum([k*k for k in range(1, n)])
78
79 def oddSquare(n):
80 """
81     Takes a positive integer n and returns the sum of the squares of all the odd positive
82     integers smaller than n.
83
84     :param int n: Number to evaluate.
85     :return int: Result of the evaluation.
86     """
87
88     sum = 0
89
90     for x in range(1,n,2):
91         sum += x * x
92
93     return sum
94
95 def oddSquareL(n):
96 """
97     A single command that computes the sum from Exercise R-1.6, relying on Python's
98     comprehension syntax and the built-in sum function.
99
100    :param int n: Number to evaluate.
101    :return int: Result of the evaluation.
102    """
103
104    return sum([k*k for k in range(1, n) if k % 2 != 0])
105
106 def quickRange():
107 """
108     The answer to the question in function form:
109
110     What parameters should be sent to the range constructor, to produce a range with values
111     50, 60, 70, 80?
112     """
113
114     for x in range(50, 81, 10): # The answer. (Start: 50, End + 1: 81, Iteration: 10)
115     |     print(x, end=' ')
116     print("The answer: range(Start: 50, End + 1: 81, Iteration: 10)")
117
118 """
119 #Main
120
121 print(is_multiple(2, 4))      # R-1.1
122 print(is_even(5))            # R-1.2
123 print(minmax((10, 2, 4, 11))) # R-1.3
124 print(nSquare(5))           # R-1.4
125 print(nSquareL(5))          # R-1.5
126 print(oddSquare(6))          # R-1.6
127 print(oddSquareL(6))         # R-1.7
128 quickRange()                # R-1.9 Note this question has only 1 correct answer/output.

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
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

```
landerson_assignment01.py ...
// |     return sum([k*k for k in range(1, n)])
78 |
79 def oddSquare(n):
80     """
81     Takes a positive integer n and returns the sum of the squares of all the odd positive
82     integers smaller than n.
83
84     :param int n: Number to evaluate.
85     :return int: Result of the evaluation.
86     """
87     sum = 0
88
89     for x in range(1,n,2):
90         sum += x * x
91
92     return sum
93
94 def oddSquareL(n):
95     """
96     A single command that computes the sum from Exercise R-1.6, relying on Python's
97     comprehension syntax and the built-in sum function.
98
99     :param int n: Number to evaluate.
100    :return int: Result of the evaluation.
101   """
102   return sum([k*k for k in range(1, n) if k % 2 != 0])
103
104 def quickRange():
105     """
106     The answer to the question in function form:
107
108     What parameters should be sent to the range constructor, to produce a range with values
109     50, 60, 70, 80?
110     """
111     for x in range(50, 81, 10): # The answer. (Start: 50, End + 1: 81, Iteration: 10)
112         print(x, end=' ')
113     print("The answer: range(Start: 50, End + 1: 81, Iteration: 10)")
114
115     """
116     #Main
117     """
118     print(is_multiple(3, 10))      # R-1.1
119     print(is_even(12))           # R-1.2
120     print(minmax((11, 22, 5, 9))) # R-1.3
121     print(nSquare(6))           # R-1.4
122     print(nSquareL(7))          # R-1.5
123     print(oddSquare(8))          # R-1.6
124     print(oddSquareL(9))         # R-1.7
125     quickRange()                # R-1.9 Note this question has only 1 correct answer/output.
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\leyon\Documents\Programming\COP3410C> & C:/Users/leyon/AppData/Local/Programs/Python/Python314/python.exe c:/Users/leyon/Documents\Programming\COP3410C>
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>
```