This quiz has 3 questions, for a total of 10 points.

1. **3 points** Suppose that L is a Python “list” (array) of length \(n\). Categorize the worst-case execution time of each of the below operations as either

   1. constant time (takes the same amount of time no matter what \(n\) is).
   2. logarithmic time (takes time proportional to \(\log n\)).
   3. linear time (takes time proportional to \(n\))
   4. quadratic (takes time proportional to \(n^2\))

Label each operation with the above item number.

- \(\text{len}(L)\)
- \(\text{L.remove(7)}\)
- \(5000 \text{ in } L\)

**Solution:**

- (1), \(\text{len}(L)\) is constant time, (1 point)
- (3), \(\text{L.remove(7)}\) is linear time, (1 point)
- (3), \(5000 \text{ in } L\) is linear time, (1 point)

2. **3 points** What is the output of the following Python program?

```python
D = { (1,0): 'red', (0,1): 'green', (0,0): 'blue' }
print(D[(0,0)])
D[(0,1)] = 'red'
print((1,1) in D)
print(D[(0,1)])
```

**Solution:**

- blue
- False
- red

Grading rubric: Looking up the key \((0,0)\) in \(D\) returns the associated string **blue** (1 point). The key \((1,1)\) is not in \(D\) (1 point). The Finally, the value for \((0,1)\) is overwritten with **red**, so that’s the last output (1 point).
3. **4 points** Complete the following implementation of the `ListIterator`'s `__next__` method. If the iterator is pointing to a node, it should return the data in the node and then move to the next node. Otherwise, the iterator should stop iterating.

```python
class Node:
    def __init__(self, data, next):
        self.data = data
        self.next = next

class List:
    def __init__(self):
        self.head = None
        self.tail = None
    def __iter__(self):
        return ListIterator(self.head)

class ListIterator:
    def __init__(self, node):
        self.node = node
    def __next__(self):
        if ___(a)___:
            raise StopIteration
        else:
            data = ___(b)___
            self.node = ___(c)___
            return ___(d)___
```

**Solution:**

(a) `self.node == None` (1 point)
(b) `self.node.data` (1 point)
(c) `self.node.next` (1 point)
(d) `data` (1 point)