Programming Assignment: Linked Lists

Linked lists are the most fundamental data structure that you are likely to encounter in a programming interview. It is therefore important for you to review it. There are many types of linked lists (single-link, double-link, xor-, circular) and many different types of operations. Use ***only*** the following code as a basis for the exercises. This code stores key-value pairs in a single linked list, in which it inserts at the end.

```
class Node:
    def __init__(self, key, value, next_node):
        self.key = key
        self.value = value
        self.next_node = next_node
    def __str__(self):
        return '{}: {}'.format(self.key, self.value)

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

    def insert(self, key, value):
        new_node = Node(value, None)
        if self.tail:
            self.tail.next_node = new_node
            self.tail = new_node
        else:
            self.head = new_node
            self.tail = new_node

    def write(self):
        current_node = self.head
        while current_node:
            print(current_node.value)
            current_node = current_node.next_node

lista = List()
lista.insert(1, 'one')
lista.insert(3, 'three')
lista.insert(2, 'two')
lista.insert(5, 'five')
lista.insert(4, 'four')
lista.write()
```

Exercises:

Write and test the following methods

(1) A method lookup(self, key) that returns None if the key is not in the list and otherwise the value associated to the key.

(2) A method `update_by_value(self, old_value, new_value)` that changes the first occurrence (and only the first occurrence) of the value by substituting the new_value.

(3) A method `update_by_key(self, key, new_value)` that changes the value associated to the key in the list to the new_value. If the key is present, nothing happens.

Hand-in:

Create a single pdf-file with a description of your solution, your code, and the results of your tests. Your tests should include cases such as the record searched for does or does not exist, the key exists, the key does not exist, et.cet.