Write append method

first...

Read over the _**init**_ method for class **LLNode** and **LinkedList**:

```
class LinkedListNode:
    """
    Node to be used in linked list

    === Attributes ===
    @param LinkedListNode next_: successor to this LinkedListNode
    @param object value: data this LinkedListNode represents
    """

    def __init__(self, value, next_=None):
        """
        Create LinkedListNode self with data value and successor next_.

        @param LinkedListNode self: this LinkedListNode
        @param object value: data of this linked list node
        @param LinkedListNode|None next_: successor to this LinkedListNode.
        @rtype: None
        """
        self.value, self.next_ = value, next_

class LinkedList:
    """
    Collection of LinkedListNodes

    === Attributes ==
    @param: LinkedListNode front: first node of this LinkedList
    @param LinkedListNode back: last node of this LinkedList
    @param int size: number of nodes in this LinkedList
                        a non-negative integer
    """

    def __init__(self):
        """
        Create an empty linked list.

        @param LinkedList self: this LinkedList
        @rtype: None
        """
        self.front, self.back self.size = None, None, 0
```

(continued on next page)

next...

Now, read the header and docstring for the method **append**, and then answer the questions that follow it.

```
def append(self, value):
    """
    Insert a new LinkedListNode with value after self.back.

    @param LinkedList self: this LinkedList.
    @param object value: value of new LinkedListNode
    @rtype: None

    >>> lnk = LinkedList()
    >>> lnk.append(5)
    >>> lnk.size
    1
    >>> print(lnk.front)
    5 ->|
    >>> lnk.append(6)
    >>> lnk.size
    2
    >>> print(lnk.front)
    5 -> 6 ->|
    """
    pass
```

1. What if this is the first node being appended? Show this with a diagram.

2. What if there are already some nodes in the list? Show this with a diagram.

Now implement the body of **append**