

Work with your neighbor. (This will be graded for participation only.)

1. Below is the code from last time to add to the end of a linked list:

```
def add_to_end(self, new):
    current = self._head
    prev = None           # initialize prev
    while current != None:
        prev = current     # keep track of previous node
        current = current._next
    prev._next = new       # add new to the end
```

Modify the code `add_to_end(self, new)` to handle the case of inserting into an empty list. **Hint:** If the list is empty, the code would be just like adding to the front of a linked list.

2. Below is the code for `get_element(self, n)`, which returns a reference to the node at position `n` in a linked list:

```
def get_element(self, n):
    elt = self._head
    while elt != None and n > 0:
        elt = elt._next
        n -= 1

    return elt
```

Draw a diagram for a linked list called `my_ll` that has four elements. Walk through the code for `my_ll.get_element(2)` and show on the diagram how the variable `elt` advances; show what that value of `n` is when the element at position 2 is reached and show which node is referred to by `elt` (which is the node that is returned).

Are there any issues with this code? What happens if you ask for the node at position 6 in `my_ll`?

3. Define a new method called `sum_even_positions(self)` that returns the sum of the elements of a linked list at the even positions in the list. Node positions begin at 0. Return 0 if the list is empty.

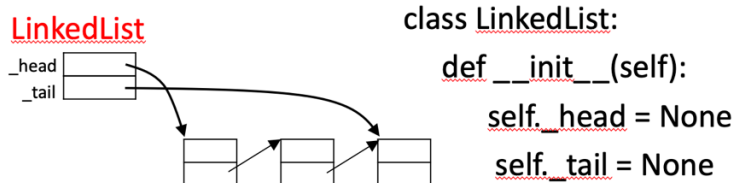
4. Below is the code for `find(self, item)`, which returns `True` if `item` is an element of the linked list and `False` otherwise:

```
def find(self, item):
    current = self._head
    while current != None:
        if item == current._value:
            return True
        else:
            return False

        current = current._next
```

- a) There is a common logic error in this code. What is the bug?
- b) Rewrite the code with the bug fixed:

5. Suppose that we modify the linked list class to maintain a tail reference. The modified code for the `LinkedList` class and a sample linked list are shown below:



Write the `append(self, new)` method for the class.