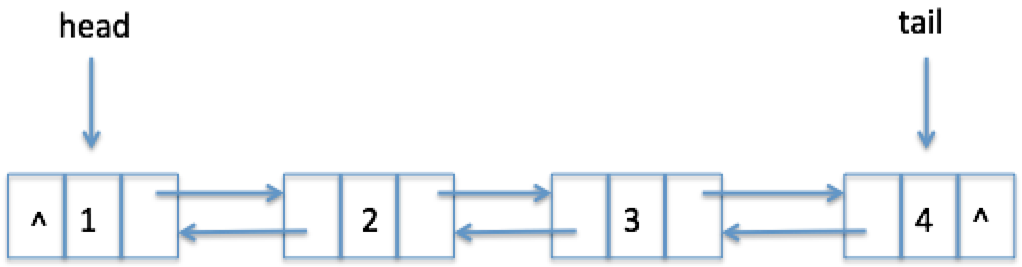
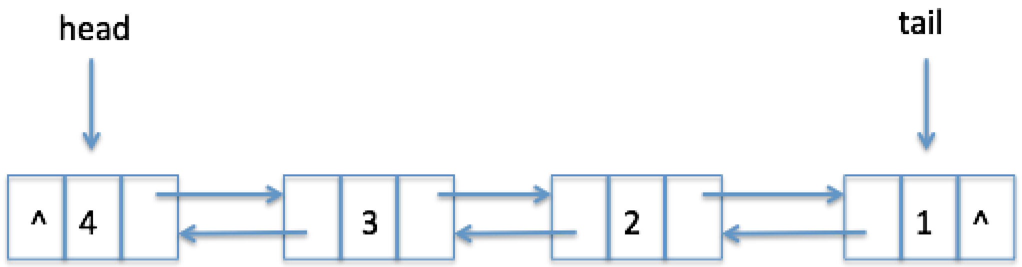
Extra Credit #2: Reverse a Doubly Linked List



Original Doubly Linked List

^: None



Reversed Doubly Linked List

Implement a reverse method for the DoublyLinkedList class to reverse an existing doubly linked list.

```
class ListNode:
    def __init__(self, data, prev = None, link = None):
        self.data = data
        self.prev = prev
        self.link = link
        if prev is not None:
            self.prev.link = self
        if link is not None:
            self.link.prev = self

class DoublyLinkedList:

    def __init__(self, L = None):
        self._head = None
        self._tail = None
        self._length = 0
        for item in L:
            self.addlast(item)

    @property
    def head(self):
        return self._head

    @property
    def tail(self):
        return self._tail

    def _addbetween(self, item, before, after):
        node = ListNode(item, before, after)
        if after is self._head:
            self._head = node
        if before is self._tail:
            self._tail = node
        self._length += 1

    def addfirst(self, item):
        self._addbetween(item, None, self._head)

    def addlast(self, item):
        self._addbetween(item, self._tail, None)

    def _remove(self, node):
        before, after = node.prev, node.link
        if node is self._head:
            self._head = after
        else:
            before.link = after
        if node is self._tail:
            self._tail = before
        else:
            after.prev = before
        self._length -= 1
        return node.data

    def removefirst(self):
        return self._remove(self._head)

    def removelast(self):
```

```
    return self._remove(self._tail)

def __len__(self):
    return self._length

def __str__(self):
    cur = self._head
    s = ''
    while cur:
        s += str(cur.data)
        s += ' '
        cur = cur.link
    s.rstrip()
    return s

def alldata(self):
    L = []
    cur = self._head
    while cur is not None:
        L.append(cur.data)
        cur = cur.link
    return L

def reverse(self):
    # Add your code here















    return self
```

```
dll = DoublyLinkedList([])

print("Before reverse:")
print(dll)
dll = dll.reverse()
print("After reverse:")
print(dll)

dll = DoublyLinkedList([0])

print("Before reverse:")
print(dll)
dll = dll.reverse()
print(dll)

dll = DoublyLinkedList([i for i in range(10)])

print("Before reverse:")
print(dll)
dll = dll.reverse()
print("After reverse:")
print(dll)
```

This extra credit assignments worth 0.5 points in your final grade.
Please submit your work to Mimir as usual.