

Unfold a LL

```
class Node:
    def __init__(self, val):
        self.data = val
        self.next = None

def unfoldLL(head):
    if head is None or head.next is None:
        return head
    head1 = head
    head2 = head.next
    tempH1 = head1
    tempH2 = head2

    while tempH2 != None and tempH2.next != None:
        tempH1.next = tempH2.next
        tempH1 = tempH1.next
        tempH2.next = tempH1.next
        tempH2 = tempH2.next
    if tempH2 != None:
        tempH1.next = None
    head2 = reverseLL(head2)
    tempH1.next = head2
    return head1

def reverseLL(head):
    if head is None or head.next is None:
        return head
    curr = head
    prev = None
    forward = head

    while curr!=None:
        forward = curr.next
        curr.next = prev
        prev = curr
        curr = forward
    return prev
```

```
head = Node(1)
# head.next = Node(8)
# head.next.next = Node(2)
# head.next.next.next = Node(7)
# head.next.next.next.next = Node(3)
# head.next.next.next.next.next = Node(6)
# head.next.next.next.next.next.next = Node(4)
# head.next.next.next.next.next.next.next = Node(5)
head1= unfoldLL(head)
while head1 != None:
    print(head1.data, end=' ')
    head1 = head1.next
# head.next = Node(7)
```