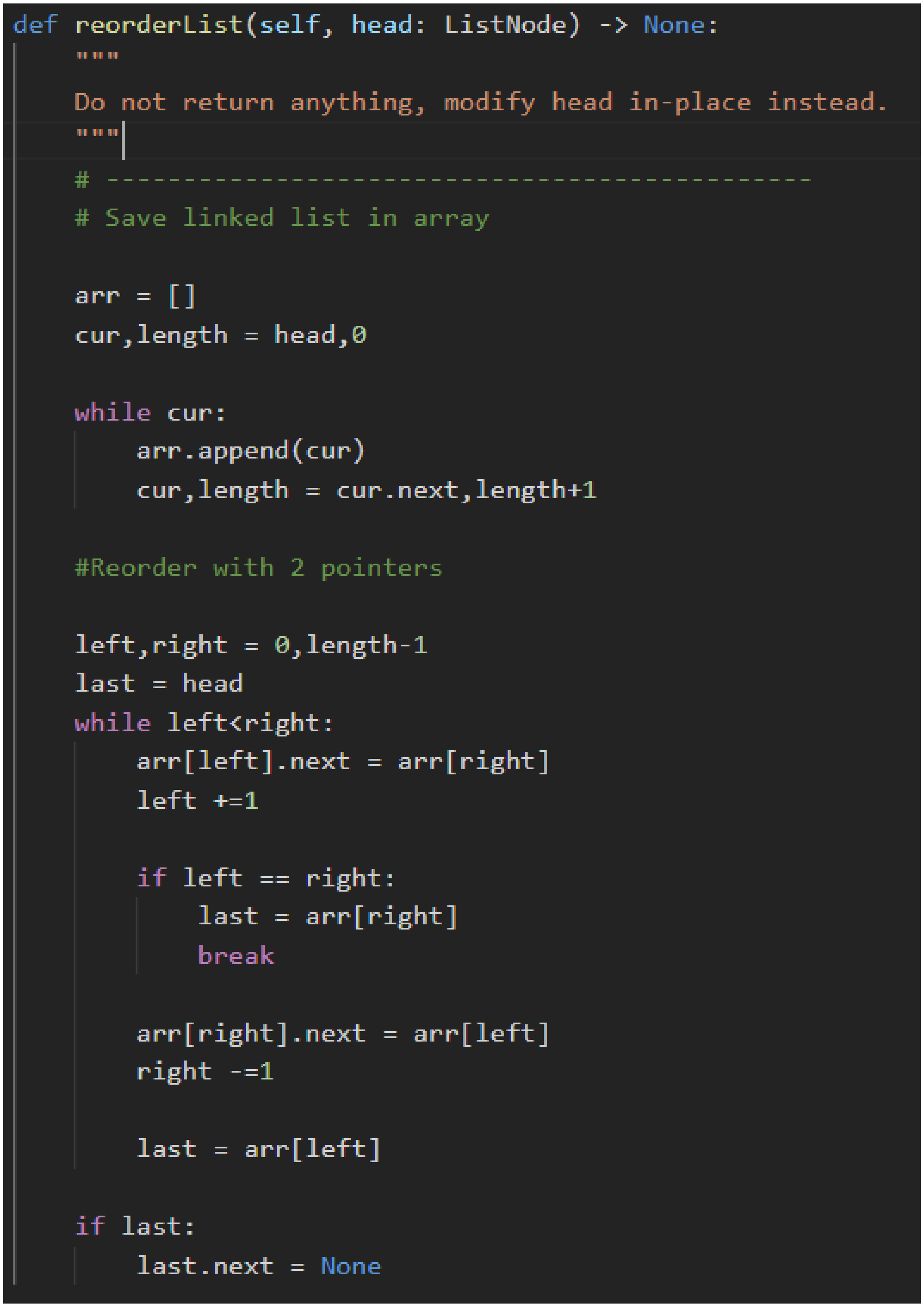
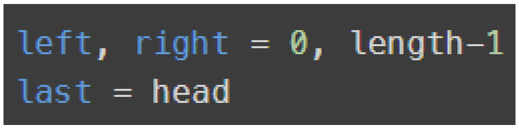
143. Reorder List

Solution using O(N) space:

1. 

```
def reorderList(self, head: ListNode) -> None:
    """
    Do not return anything, modify head in-place instead.
    """
    # ---------------------------------------------
    # Save linked list in array

    arr = []
    cur,length = head,0

    while cur:
        arr.append(cur)
        cur,length = cur.next,length+1

    #Reorder with 2 pointers

    left,right = 0,length-1
    last = head
    while left<right:
        arr[left].next = arr[right]
        left +=1

        if left == right:
            last = arr[right]
            break

        arr[right].next = arr[left]
        right -=1

        last = arr[left]

    if last:
        last.next = None
```

2. Here basically we used an array to store our entire linked list
3. Next we have two pointer left and right and initially assume head is last



```
left, right = 0, length-1
last = head
```

4. Now we iterate our array..until left < right
5. Our array → [1l,2,3,4r] ..
6. Next we link arr[left].next = arr[right] → it is same like 1→4
7. Then we increment the L(L is 2) and we link arr[right].next = L → it is same like 1→4→2 and we decrement the R
8. And we assign last to left (arr[left] .. ideally it is the last node in the LL)
9. Our array → [1,2l,3r,4] ..
10. We do the same steps above until l<r
11. And if l == r: then it says that we have reached our last element in the array and make last = arr[right]