



# 2022.12.16 스터디 자료

## 내가 푼 문제

- LeetCode 002. Add Two Numbers

## 예시

Input: l1 = [2,4,3], l2 = [5,6,4]

Output: [7,0,8]

Explanation: 342 + 465 = 807.

## 방법

### 1 레퍼런스

```
class Solution {
    // Add Two Numbers (Java improved)
    public ListNode addTwoNumbers(ListNode l1, ListNode l2) {
        ListNode dummyHead = new ListNode(0);
        ListNode curr = dummyHead;
        int carry = 0;
        while (l1 != null || l2 != null || carry != 0) {
            int x = (l1 != null) ? l1.val : 0;
            int y = (l2 != null) ? l2.val : 0;
            int sum = carry + x + y;
            carry = sum / 10;
            curr.next = new ListNode(sum % 10);
            curr = curr.next;
            if (l1 != null)
                l1 = l1.next;
            if (l2 != null)
                l2 = l2.next;
        }
        return dummyHead.next;
    }
}
```

## 2 내가 푼 방법

```
/**
 * Definition for singly-linked list.
 * public class ListNode {
 *     int val;
 *     ListNode next;
 *     ListNode() {}
 *     ListNode(int val) { this.val = val; }
 *     ListNode(int val, ListNode next) { this.val = val; this.next = next; }
 * }
 */

class Solution {
    public ListNode addTwoNumbers(ListNode l1, ListNode l2) {
        /*
         * l1 = [2,4,3]
         * l2 = [5,6,4]
         * result = [7,0,8]
         */

        /* ListNode를 0을 넣어 초기화 시킨다. */
        ListNode node = new ListNode(0);

        /* 결과를 반환할 ListNode를 만들고 현재 노드를 넣어준다. */
        ListNode result = node;

        /* 더할 값을 0으로 초기화 시켜준다. */
        int sum = 0;

        /* 반복문을 통해 두 ListNode의 값을 더한다. */
        while(l1 != null || l2 != null || sum > 0) {
            if (l1 != null) {
                sum += l1.val; /* 2
                               * 4
                               * 1 + 3
                               */
                l1 = l1.next; /* 4
                               * 3
                               * null
                               */
            }

            if (l2 != null) {
                sum += l2.val; /* 2 + 5 = 7
                               * 4 + 6 = 10
                               * 1 + 3 + 4 = 8
                               */
                l2 = l2.next; /* 6
                               * 4
                               * null
                               */
            }

            result.val = sum % 10;
            sum = sum / 10;
            result.next = new ListNode(0);
            result = result.next;
        }

        result.next = null;
        return result;
    }
}
```

```

/* 나머지와 나누는 이유 = 합이 두 자리수가 될 수 있기 때문
 * 한 자리수 이상이면, 일의 자리 수가 값이 되고 십의 자리수의 값 만큼 다음에 더해줘야 함.
 * 나머지 == 일의 자리 수 값
 * 나누기 == 십의 자리 수 값
 */
node.next = new ListNode(sum % 10); /* node = 0, node.next = 7
                                     * node = 7, node.next = 0
                                     * node = 0, node.next = 8
                                     */

sum /= 10; /* sum = 0
           * sum = 1
           * sum = 0
           */

node = node.next; /* node = 7
                  * node = 0
                  * node = 8
                  */
}

/* result = node로 선언
 * result = [7,0,8]
 * result.next = null
 * 위의 값 반환
 */
return result.next;
}
}

```