

## 7B. Leetcode 1669

### Merge in Between Linked List

```
struct ListNode* mergeInBetween(struct ListNode* list1, int a, int b, struct ListNode* list2) {  
    struct ListNode *prevA = list1;  
    int i;  
  
    // Find node at index a-1  
    for (i = 0; i < a - 1; i++) {  
        prevA = prevA->next;  
    }  
  
    // Find node at index b  
    struct ListNode *nodeB = prevA->next;  
    for (i = a; i <= b; i++) {  
        nodeB = nodeB->next;  
    }  
  
    // Connect list2 to list1  
    prevA->next = list2;  
  
    // Find last node of list2  
    struct ListNode *tail2 = list2;  
    while (tail2->next != NULL) {  
        tail2 = tail2->next;  
    }  
}
```

```
// Connect last node of list2 to nodeB->next  
tail2->next = nodeB;  
  
return list1;  
}
```

OUTPUT:

**Accepted** Runtime: 0 ms

Case 1     Case 2

Input

```
list1 =  
[10,1,13,6,9,5]
```

```
a =
```

```
3
```

```
b =
```

```
4
```

```
list2 =  
[1000000,1000001,1000002]
```

Output

```
[10,1,13,1000000,1000001,1000002,5]
```

Expected

```
[10,1,13,1000000,1000001,1000002,5]
```

**Accepted** Runtime: 0 ms

Case 1

Case 2

Input

```
list1 =  
[0,1,2,3,4,5,6]
```

```
a =  
2
```

```
b =  
5
```

```
list2 =  
[1000000,1000001,1000002,1000003,1000004]
```

Output

```
[0,1,1000000,1000001,1000002,1000003,1000004,6]
```

Expected

```
[0,1,1000000,1000001,1000002,1000003,1000004,6]
```