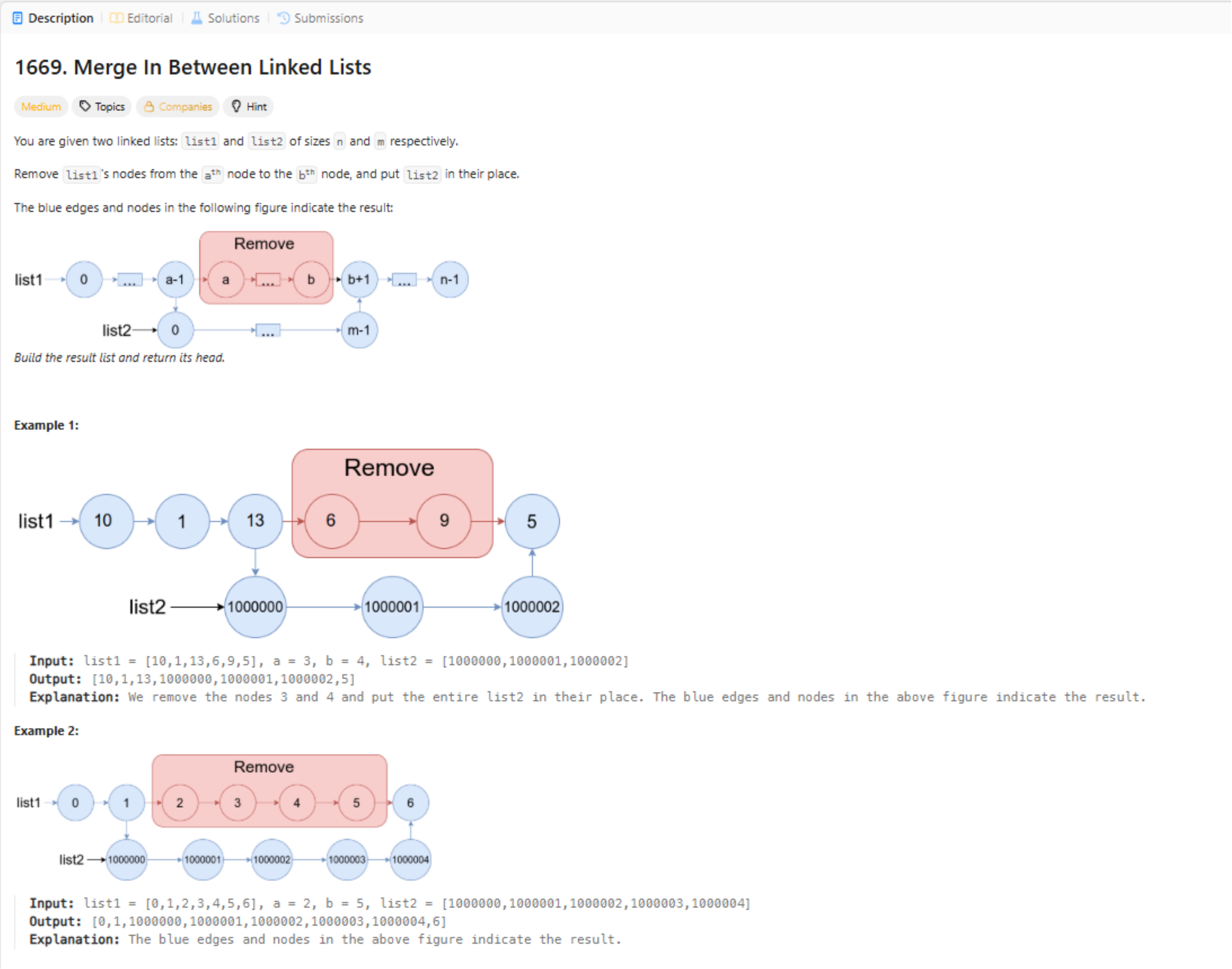


1669. Merge In Between Linked Lists

You are given two linked lists: `list1` and `list2` of sizes `n` and `m` respectively.

Remove `list1`'s nodes from the a^{th} node to the b^{th} node, and put `list2` in their place.

The blue edges and nodes in the following figure indicate the result:

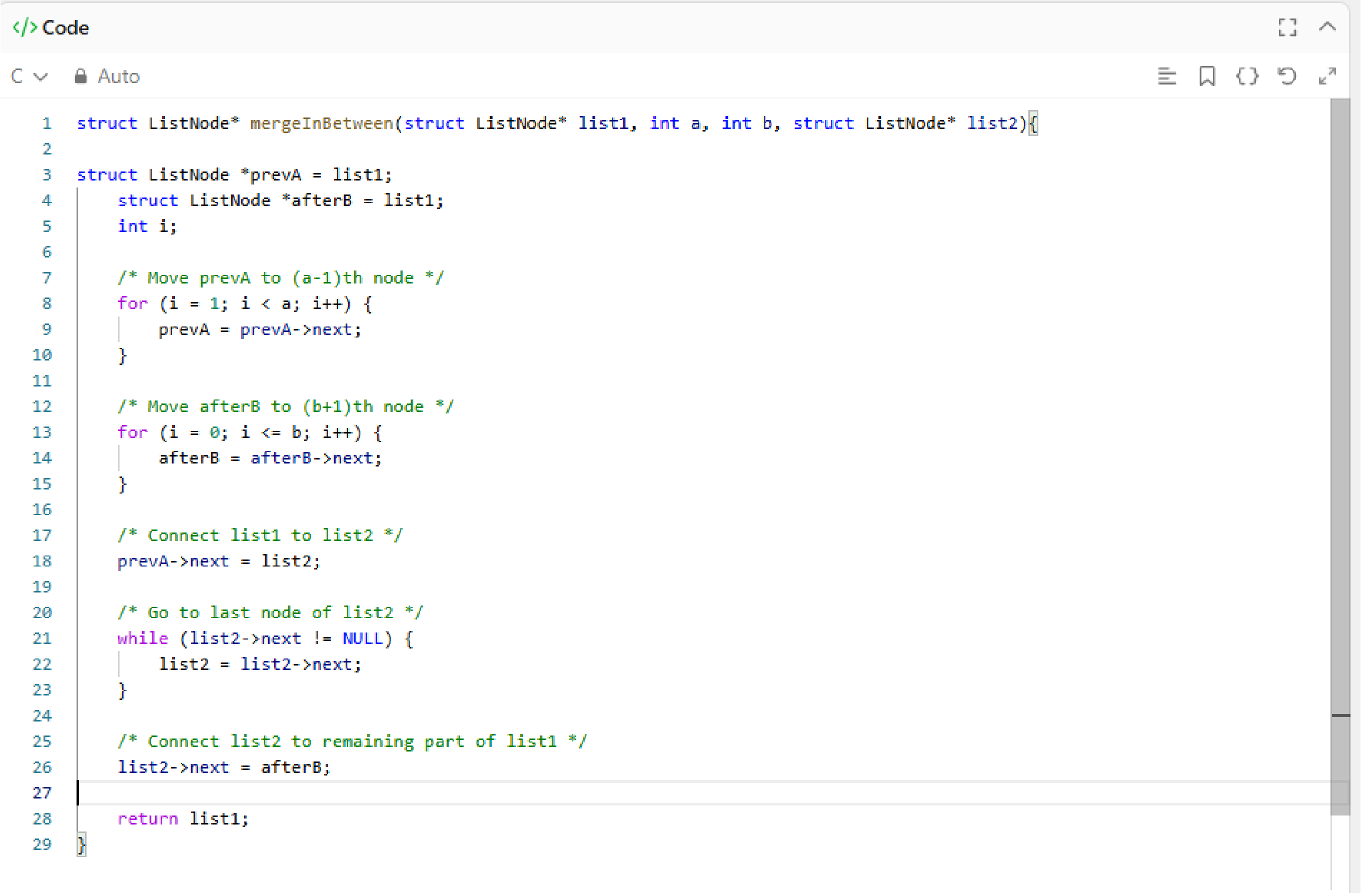
Build the result list and return its head.

Example 1:

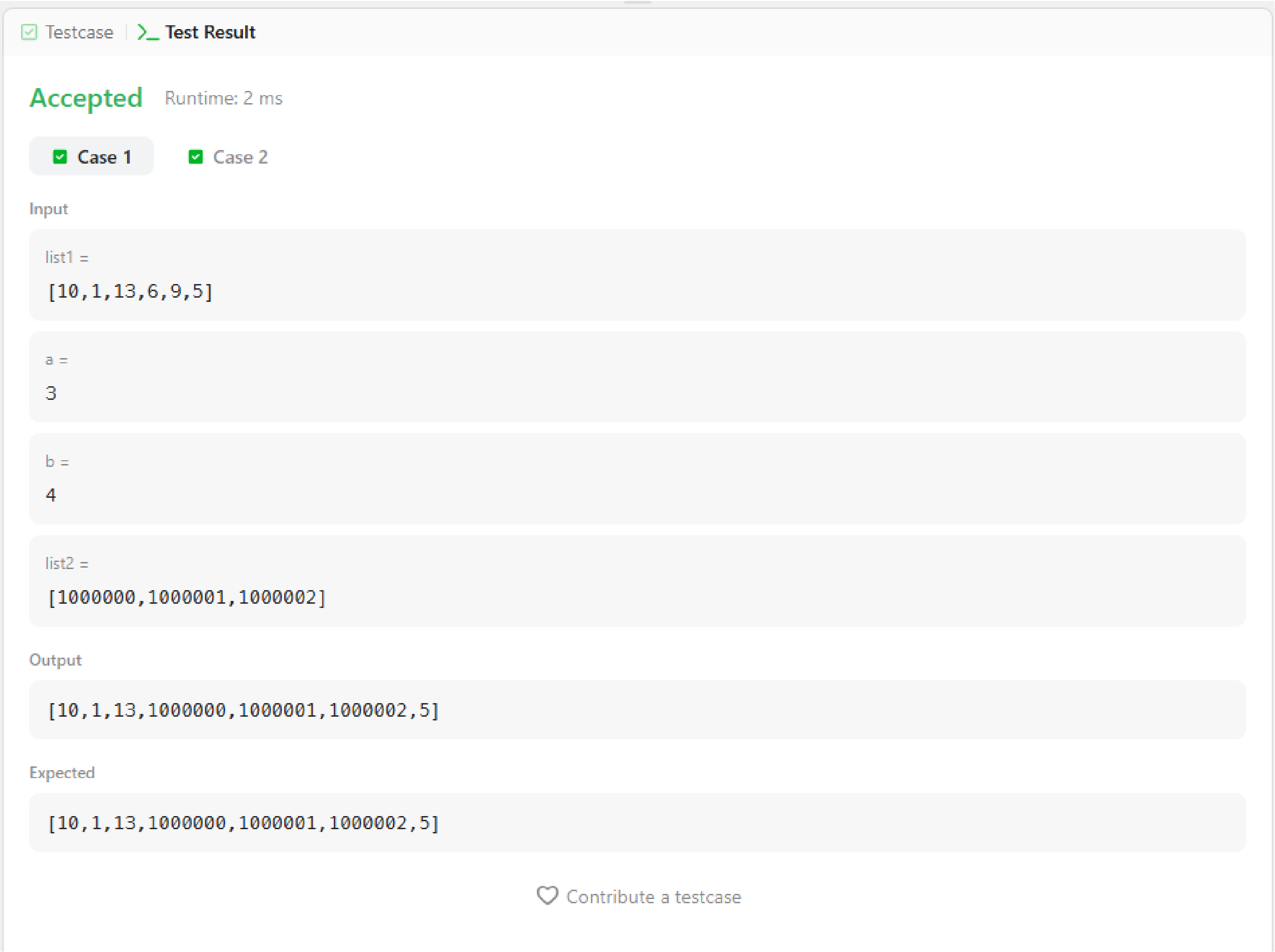
```
Input: list1 = [10,1,13,6,9,5], a = 3, b = 4, list2 = [1000000,1000001,1000002]
Output: [10,1,13,1000000,1000001,1000002,5]
Explanation: We remove the nodes 3 and 4 and put the entire list2 in their place. The blue edges and nodes in the above figure indicate the result.
```

Example 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]
Explanation: The blue edges and nodes in the above figure indicate the result.
```



```c
struct ListNode* mergeInBetween(struct ListNode* list1, int a, int b, struct ListNode* list2){

struct ListNode *prevA = list1;
    struct ListNode *afterB = list1;
    int i;

    /* Move prevA to (a-1)th node */
    for (i = 1; i < a; i++) {
        prevA = prevA->next;
    }

    /* Move afterB to (b+1)th node */
    for (i = 0; i <= b; i++) {
        afterB = afterB->next;
    }

    /* Connect list1 to list2 */
    prevA->next = list2;

    /* Go to last node of list2 */
    while (list2->next != NULL) {
        list2 = list2->next;
    }

    /* Connect list2 to remaining part of list1 */
    list2->next = afterB;

    return list1;
}
```



Accepted Runtime: 2 ms

Input

list1 =

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

a =

```
3
```

b =

```
4
```

list2 =

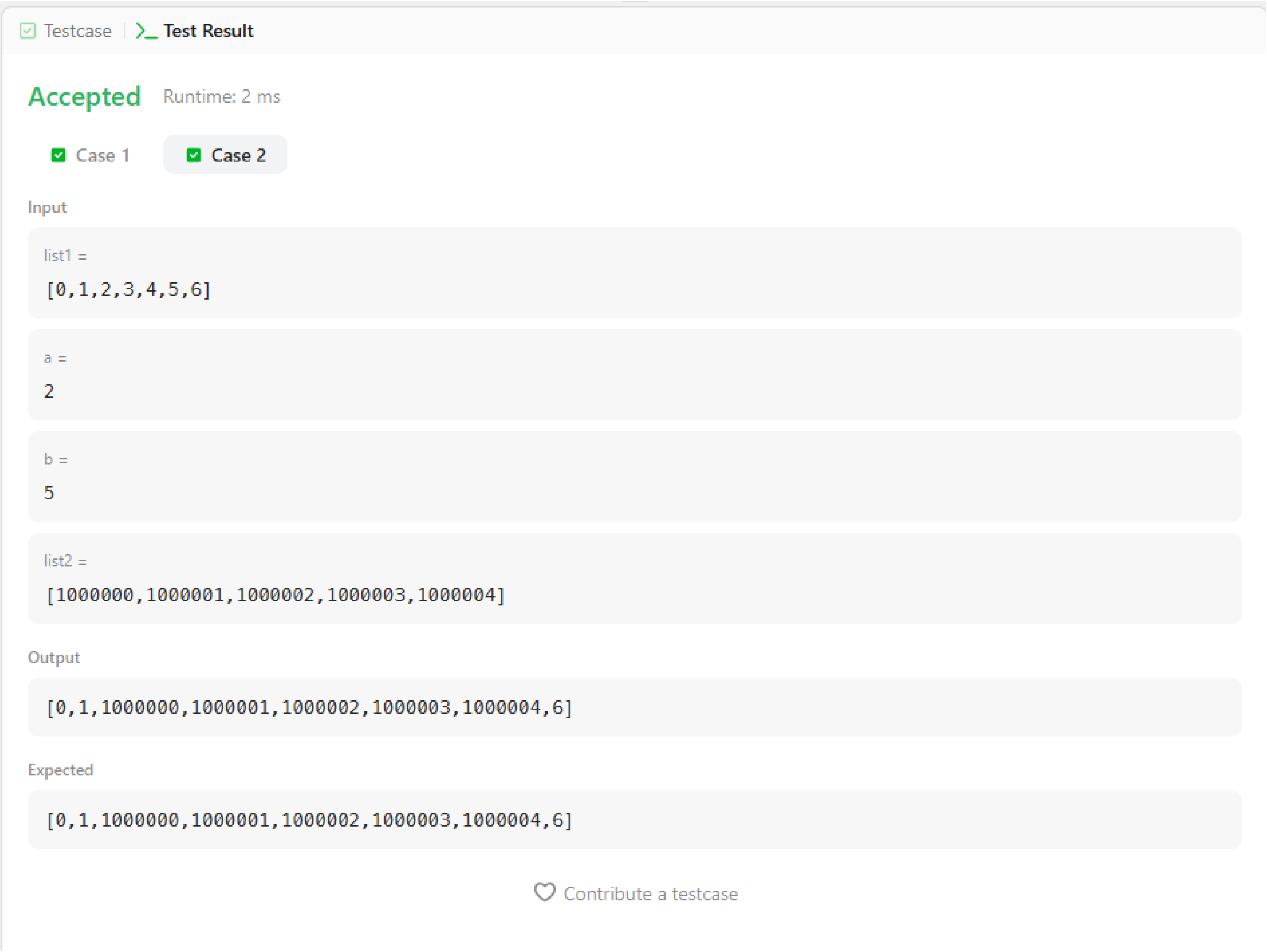
```
[1000000,1000001,1000002]
```

Output

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

Expected

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



Accepted Runtime: 2 ms

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]
```