

6. Merge in between linked lists (LeetCode)

```
class Solution {  
public:  
    ListNode* mergeInBetween(ListNode* list1, int a, int b, ListNode* list2) {  
        ListNode* prevA = list1;  
        for (int i = 0; i < a - 1; i++)  
            prevA = prevA->next;  
  
        ListNode* afterB = prevA;  
        for (int i = 0; i <= b - a + 1; i++)  
            afterB = afterB->next;  
  
        prevA->next = list2;  
  
        ListNode* tail = list2;  
        while (tail->next != nullptr)  
            tail = tail->next;  
  
        tail->next = afterB;  
  
        return list1;  
    }  
};
```

OUTPUT:

The screenshot shows a software interface for testing code. At the top, there are two tabs: "Testcase" (with a checked checkbox) and "Test Result".

Under "Testcase", the following variables are defined:

- list1 = [10, 1, 13, 6, 9, 5]
- a = 3
- b = 4
- list2 = [1000000, 1000001, 1000002]

Below these, under the "Output" section, is the resulting list: [10, 1, 13, 1000000, 1000001, 1000002, 5].

Under the "Expected" section, the target list is shown: [10, 1, 13, 1000000, 1000001, 1000002, 5].

Testcase | [Test Result](#)

```
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]
```