

1669. Merge in Between Linked Lists

```
#include <stdio.h>
#include <stdlib.h>

/* Definition for singly-linked list */
struct ListNode {
    int val;
    struct ListNode *next;
};

/* Function to create a new node */
struct ListNode* createNode(int val) {
    struct ListNode* newNode = (struct ListNode*)malloc(sizeof(struct ListNode));
    newNode->val = val;
    newNode->next = NULL;
    return newNode;
}

/* Function to display linked list */
void displayList(struct ListNode* head) {
    struct ListNode* temp = head;
    while (temp != NULL) {
        printf("%d -> ", temp->val);
        temp = temp->next;
    }
    printf("NULL\n");
}

/* Function to merge list2 into list1 between indices a and b */
struct ListNode* mergeInBetween(struct ListNode* list1, int a, int b, struct ListNode* list2) {
    struct ListNode *prevA = list1;
```

```
struct ListNode *afterB = list1;

/* Move prevA to node just before index a */
for (int i = 0; i < a - 1; i++) {
    prevA = prevA->next;
}

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

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

/* Move to the end of list2 */
struct ListNode *tail = list2;
while (tail->next != NULL) {
    tail = tail->next;
}

/* Connect tail of list2 to afterB */
tail->next = afterB;

return list1;
}

int main() {
    /* Create list1: 0 -> 1 -> 2 -> 3 -> 4 -> 5 */
    struct ListNode *list1 = createNode(0);
    list1->next = createNode(1);
```

```
list1->next->next = createNode(2);

list1->next->next->next = createNode(3);

list1->next->next->next->next = createNode(4);

list1->next->next->next->next->next = createNode(5);

/* Create list2: 100 -> 101 -> 102 */

struct ListNode *list2 = createNode(100);

list2->next = createNode(101);

list2->next->next = createNode(102);

int a = 2, b = 4; // Merge list2 between indices 2 and 4 of list1

printf("List1 before merge:\n");

displayList(list1);

printf("List2:\n");

displayList(list2);

/* Merge lists */

list1 = mergeInBetween(list1, a, b, list2);

printf("List1 after merge:\n");

displayList(list1);

return 0;

}
```

OUTPUT:

The screenshot shows a terminal window with the following output:

```
c:\Users\Mohammed Javeed\OneDrive\Desktop\Javeed\Leetcode_Problems>cd "c:\Users\Mohammed Javeed\OneDrive\Desktop\Javeed\Leetcode_Problems\"&& gcc 109_ConvertListToBST.c -o 109_ConvertListToBST && "c:\Users\Mohammed Javeed\OneDrive\Desktop\Javeed\Leetcode_Problems\"109_ConvertListToBST
Enter number of elements in list1: 6
Enter elements of list1 in order:
Enter element 1: 0
Enter element 2: 1
Enter element 3: 2
Enter element 4: 3
Enter element 5: 4
Enter element 6: 5
Enter number of elements in list2: 3
Enter elements of list2 in order:
Enter element 1: 100
Enter element 2: 101
Enter element 3: 102
Enter index a (start of deletion in list1): 2
Enter index b (end of deletion in list1): 4

List1 before merge:
0 -> 1 -> 2 -> 3 -> 4 -> 5 -> NULL
List2:
100 -> 101 -> 102 -> NULL

List1 after merge:
0 -> 1 -> 100 -> 101 -> 102 -> 5 -> NULL
c:\Users\Mohammed Javeed\OneDrive\Desktop\Javeed\Leetcode_Problems>
```