```
/**
 * Definition for singly-linked list.
 * public class ListNode {
 *     int val;
 *     ListNode next;
 *     ListNode(int x) { val = x; }
 * }
 */
class Solution {
    public ListNode reverseBetween(ListNode head, int m, int n) {
        ListNode dh = new ListNode(0);
        dh.next = head;
        ListNode itr = dh;

        ListNode nodem = getPrev(itr, m);
        ListNode noden = getPrev(itr, n+1);
        ListNode nextm = nodem.next;
        ListNode nextn = noden.next;
        nodem.next = null;
        noden.next = null;

        ListNode newHead = reverse(nextm);
        nodem.next = newHead;
        nextm.next = nextn;
        return dh.next;
    }

    private ListNode reverse(ListNode node) {
        ListNode pre = null;
        while (node != null) {
            ListNode tmp = node.next;
            node.next = pre;
            pre = node;
            node = tmp;
        }
        return pre;
    }

    private ListNode getPrev(ListNode node, int idx) {
        while (idx != 1) {
            idx--;
            node = node.next;
        }
        return node;
    }
}
```