```java
/**
 * Definition for singly-linked list.
 * class ListNode {
 *     public int val;
 *     public ListNode next;
 *     ListNode(int x) { val = x; next = null; }
 * }
 */
public class Solution {
    public ListNode reverseBetween(ListNode head, int a, int b) {
        int size=getSize(head);

        if(a==1 && b!=size){
            ListNode s=head;
            ListNode e=head;
            for(int i=1;i<b;i++) e=e.next;
             ListNode afb=e.next;
             e.next=null;
             head=reverseList(s);
             s.next=afb;

             return head;
        }
        else if(b==size && a!=1){
            ListNode bFa=head;
            for(int i=1;i<a-1;i++) bFa=bFa.next;
            ListNode s=bFa.next;
            bFa.next=reverseList(s);

            return head;
        }
        else if(a>1 && b<size){
            ListNode bFa=head;
            for(int i=1;i<a-1;i++) bFa=bFa.next;
            ListNode s=bFa.next;

            ListNode e=head;
           for(int i=1;i<b;i++) e=e.next;
            ListNode afb=e.next;
            e.next=null;

            bFa.next=reverseList(s);
            s.next=afb;

            return head;

        }
        else{
            return reverseList(head);
        }
    }

```

```
53.
54.
55.
56.
57.
58.     public int getSize(ListNode h){
59.         ListNode t=h;
60.         int c=0;
61.         while(t!=null){
62.             c++;
63.             t=t.next;
64.         }
65.         return c;
66.     }
67.     public ListNode reverseList(ListNode A) {
68.         ListNode curr=A;
69.         ListNode previous=null;
70.         ListNode nex=null;
71.         while(curr!=null) {
72.             nex=curr.next;
73.             curr.next=previous;
74.             previous=curr;
75.             curr=nex;
76.         }
77.         return previous;
78.     }
79. }
```

Problem Link: [reverse-link-list-ii InterviewBit]