Reverse a singly linked list.

Example:

```
Input: 1->2->3->4->5->NULL
Output: 5->4->3->2->1->NULL
```

Reverse a linked list from position m to n. Do it in one-pass.

Note: 1 ≤ m ≤ n ≤ length of list.

Example:

```
Input: 1->2->3->4->5->NULL, m = 2, n = 4
Output: 1->4->3->2->5->NULL
```

Implement and test your code in the SinglyLinkedList.java as flowing:

```
public class SinglyLinkedList {
	// reference that points to the list head
	public ListNode head;
	// nested class for singly-list node
	private static class ListNode {
		int val;
		ListNode next;
		ListNode(int x) {val = x;}
		ListNode(int x, ListNode nextIn) {
			this.val = x;
			this.next = nextIn;
		}
	}
	public SinglyLinkedList() {head = null;}

	// add node to the end of list
	private void add(int val) {
		ListNode e = new ListNode(val, head);
		head = e;
	}
	public String toString() {
		String mylist = "";
		ListNode e = head;
		while(e != null) {
			mylist = mylist + e.val + " ";
			e = e.next;
		 }
		 return mylist;
	}

	private void reverseList() {
		 // place your code here
```

```
	}

	private void reverseBetween(int m, int n) {
		// place your code here
	}

	public static void main(String args[]) {
		SinglyLinkedList list1 = new SinglyLinkedList();
		for(int i = 10; i > 0; i--) {
			list1.add(i);
		}
		System.out.println(list1);
		list1.reverseList();
		System.out.println(list1);
		list1.reverseList();
		System.out.println(list1);
		list1.reverseBetween(3,7);
		System.out.println(list1);
	}
}
```

The expected output of the code is as follows:

```
1 2 3 4 5 6 7 8 9 10
10 9 8 7 6 5 4 3 2 1
1 2 3 4 5 6 7 8 9 10
1 2 7 6 5 4 3 8 9 10
```