

Prompt	Scratchpad	Our Solution(s)	Video Explanation	Run Code
--------	------------	-----------------	-------------------	----------

Solution 1

```
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 package main
4
5 ▼ type LinkedList struct {
6     Value int
7     Next *LinkedList
8 }
9
10 // O(n) time | O(1) space - where n is the number of nodes in the Linked List
11 ▼ func ShiftLinkedList(head *LinkedList, k int) *LinkedList {
12     listLength := 1
13     listTail := head
14     ▼ for listTail.Next != nil {
15         listTail = listTail.Next
16         listLength += 1
17     }
18
19     offset := abs(k) % listLength
20     ▼ if offset == 0 {
21         return head
22     }
23
24     newTailPosition := listLength - offset
25     ▼ if k <= 0 {
26         newTailPosition = offset
27     }
28
29     newTail := head
30     ▼ for i := 1; i < newTailPosition; i++ {
31         newTail = newTail.Next
32     }
33
34     newHead := newTail.Next
35     newTail.Next = nil
36     listTail.Next = head
37     return newHead
38 }
39
40 ▼ func abs(k int) int {
41     ▼ if k > 0 {
42         return k
43     }
44     return -k
45 }
46
```

