



Hard **Codewriting** **4000**



Note: Try to solve this task in $O(\text{list size})$ time using $O(1)$ additional space, since this is what you'll be asked during an interview.



Given a singly linked list of integers `l` and a non-negative integer `n`, move the last `n` list nodes to the beginning of the linked list.



Example

- For `l = [1, 2, 3, 4, 5]` and `n = 3`, the output should be
`rearrangeLastN(l, n) = [3, 4, 5, 1, 2]`;
- For `l = [1, 2, 3, 4, 5, 6, 7]` and `n = 1`, the output should be
`rearrangeLastN(l, n) = [7, 1, 2, 3, 4, 5, 6]`.



Input/Output

- [execution time limit] 20 seconds (scala)

- [input] `linkedList.integer l`

A singly linked list of integers.

Guaranteed constraints:

$0 \leq \text{list size} \leq 10^5$,
 $-1000 \leq \text{element value} \leq 1000$.

- [input] `integer n`

A non-negative integer.

Guaranteed constraints:

$0 \leq n \leq \text{list size}$.

- [output] `linkedList.integer`

Return `l` with the `n` last elements moved to the beginning.

[Scala] Syntax Tips

```
def helloWorld(name: String): String = {
  println("This prints to the console when you Run Tests")
  "Hello, " + name
}
```