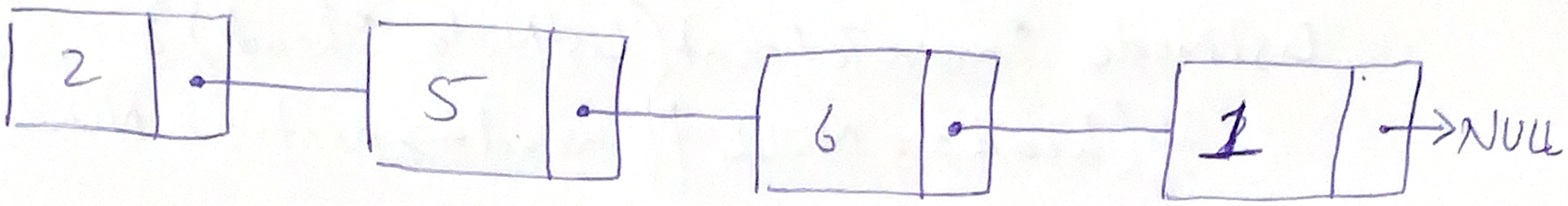
Move last Element to Front of a LinkedList



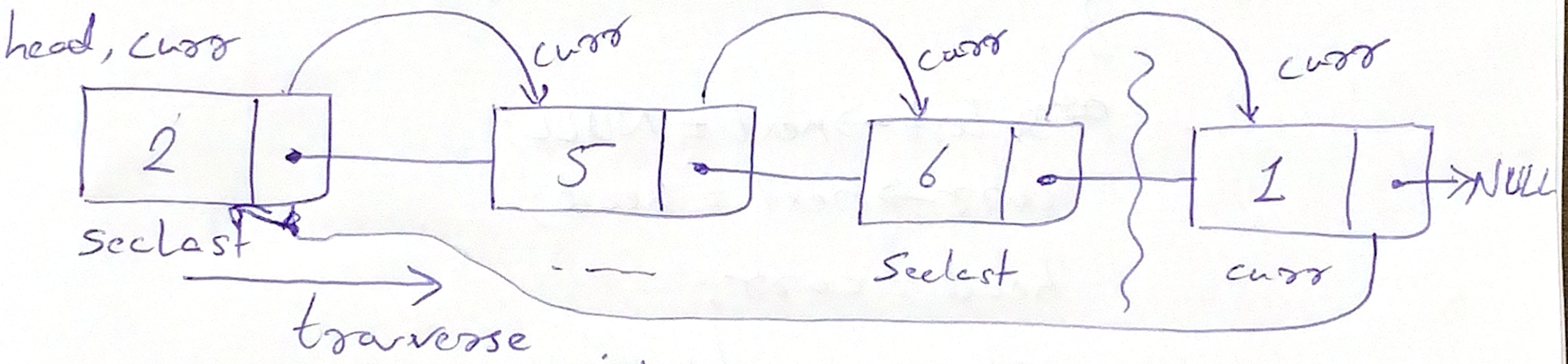
What the question is asking?

→ We have to break the linkage between n^{th} and $n^{th}-1$ node.

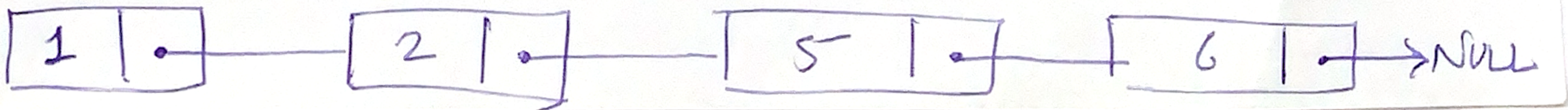
→ Assign $n^{th}-1$, address ~~pos~~ as Null

→ n^{th} node will be merge at the head position

→ n^{th} node will be the head node then.



- create two ~~pointers~~ * curr = head, *seclast = NULL
- traverse till last
- if n^{th} node address is NULL (n^{th} node = curr)
- Assign seclast (which is $n^{th}-1$ node) address as NULL.
- provide curr → next = head.
  curr = head
- Now the list will look like this



Code:

```
class LinkedListMovement {
public:
    ListNode *moveToFront(ListNode *head) {
        if(head == NULL || head -> next == NULL){
            return head;
    }
    else {
        struct ListNode *curr = head;
        struct ListNode *seclast = NULL;
        while(curr -> next != NULL){
            seclast = curr;
            curr = curr -> next;
        }
        seclast -> next = NULL;
        curr -> next = head;
        head = curr;
        return head;
    }
}
```