N^{th} NODE FROM BACK

☆ In this problem, we are given a singly linked list and a value K. Our job is to return the linked list after deleting the N^{th} node from the back.

Bruteforce solution is to calculate the length of the linked lists Then just iterate to N - K and delete that node. Time Complexity is O(N + K).

Optimal Solution is to use 2 pointers, a fast one and a slow one. Fast one will get a headstart of K steps, after which both the slow and fast pointers move together. In this manner, when the fast pointer is at the last node, slow pointer will reach the $N-K^{th}$ Node.

C++ :

```
Node* nthNodeFromBack(Node* head, int K)
    Node* fast = head;
    Node* slow = head;
    for(int i = 0; i < K; i++) {
        fast = fast -> next;
    }
```

```
    while (fast->next != nullptr) {
        slow = slow->next;
        fast = fast->next;
    }
    Node* delNode = slow->next;
    slow->next = slow->next->next;
    delete delNode;
    return head;
}
```