BEST Linked List Questions Java

1. Find the nth node from the end & remove it.

Time complexity - O(n)
Space complexity - O(1)

```
public ListNode removeNthFromEnd(ListNode head, int n) {
      temp = temp.next;
  while(cp != ptf) {
```

2. Check if a Linked List is a palindrome

Time complexity - O(n)

Space complexity - O(1)

```
public ListNode getMiddle(ListNode head) {
public ListNode reverse(ListNode head) {
Apna College
```

```
curr = next;
public boolean isPalindrome(ListNode head) {
  ListNode firstHalfEnd = getMiddle(head);
```

3. Detecting Loop in a Linked List.

Time complexity - O(n)

Space complexity - O(1)

```
public boolean hasCycle(ListNode head) {
   ListNode slow = head;
   ListNode fast = head;

while(fast != null && fast.next != null) {
    slow = slow.next;
    fast = fast.next.next;

   if(fast == slow) {
        return true;
    }
}

return false;
}
```

Homework Problems

1. Removing Loops in a Linked List.

(Please try on your own first. The answer will be updated soon!)