DSA QUIZ 2 AND 3

**QUIZ 2**

public class ListNode {

int val;

ListNode next;

ListNode(int x) {

val = x;

next = null;

}

}

public class LinkedListCycle {

public ListNode detectCycle(ListNode head) {

if (head == null || head.next == null)

return null;

ListNode slow = head;

ListNode fast = head;

// Move slow and fast pointers until they meet

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

slow = slow.next;

fast = fast.next.next;

if (slow == fast)

break;

}

// If there's no cycle, return null

if (fast == null || fast.next == null)

return null;

// Move one pointer back to the head

slow = head;

// Move both pointers one step at a time until they meet again

while (slow != fast) {

slow = slow.next;

fast = fast.next;

}

// Return the node where they meet (starting point of the cycle)

return slow;

}

}

**QUIZ 3**

public class ListNode {

int val;

ListNode next;

ListNode(int x) {

val = x;

next = null;

}

}

public class ReverseLinkedList {

public ListNode reverseList(ListNode head) {

ListNode prev = null;

ListNode current = head;

while (current != null) {

ListNode nextNode = current.next;

current.next = prev;

prev = current;

current = nextNode;

}

// At the end of the loop, prev will be the new head of the reversed list

return prev;

}

}