**Assignment12**

**1.**

class ListNode {

constructor(val = 0, next = null) {

this.val = val;

this.next = next;

}

}

function deleteMiddleNode(head) {

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

return null;

}

let slowPtr = head;

let fastPtr = head;

let prev = null;

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

fastPtr = fastPtr.next.next;

prev = slowPtr;

slowPtr = slowPtr.next;

}

if (slowPtr === null) {

return null;

}

if (prev === null) {

head = slowPtr.next;

} else {

prev.next = slowPtr.next;

}

return head;

}

// Helper function to print the linked list

function printLinkedList(head) {

let result = "";

while (head !== null) {

result += head.val + " ";

head = head.next;

}

console.log(result);

}

// Example 1

let head1 = new ListNode(1);

head1.next = new ListNode(2);

head1.next.next = new ListNode(3);

head1.next.next.next = new ListNode(4);

head1.next.next.next.next = new ListNode(5);

printLinkedList(head1);

head1 = deleteMiddleNode(head1);

printLinkedList(head1);

// Example 2

let head2 = new ListNode(2);

head2.next = new ListNode(4);

head2.next.next = new ListNode(6);

head2.next.next.next = new ListNode(7);

head2.next.next.next.next = new ListNode(5);

head2.next.next.next.next.next = new ListNode(1);

console.log("Example 2:");

console.log("Input: ");

printLinkedList(head2);

head2 = deleteMiddleNode(head2);

console.log("Output: ");

printLinkedList(head2);