

# Linked List Problems Summary



List of problem in linked list  
i will give you list you have give problem description okay are you ready ?



Yes, I'm ready! Share the list of problems, and I'll provide the descriptions for each. 😊



i start with intro  
arr to list convert  
Reverse DLL  
add 2 Number in linked list  
odd even linked list  
sort a linked list 0,1,2  
remove nth node from end  
check if list is pallindrom or not  
add 1 to a number represent by list  
find the intersection point of y list  
find the middle of element of list  
detect loop of  
find the length of loop  
delte the middle node of list  
find thge startring point of loop and cycle  
delelte all occ of key in dll  
find the all pairs with give sum of DLL  
reveser Node in k groups  
rotate a kist  
merge tow sorted list  
sort a linked list merge sort in linked list  
design brower  
you have to give numbering to each question along with decription of problem i alredy sovled them  
but take for revision what questin i solved



Here's a numbered list with descriptions for each linked list problem you mentioned. Use it for revision!

## 1. Array to List Conversion

**Description:** Convert a given array into a linked list by creating nodes for each element and linking them sequentially.

## 2. Reverse a Doubly Linked List (DLL)

**Description:** Reverse the order of a doubly linked list by swapping the `next` and `prev` pointers of each node.

## 3. Add Two Numbers in Linked List

**Description:** Two linked lists represent two numbers where each node contains a single digit. Add the numbers and return the sum as a linked list.

## 4. Odd-Even Linked List

**Description:** Rearrange a linked list such that all odd-indexed nodes come before even-indexed nodes while maintaining their relative order.

## 5. Sort a Linked List of 0, 1, 2

**Description:** Given a linked list containing only 0s, 1s, and 2s, sort it in ascending order without using extra space.

## 6. Remove Nth Node from End of List

**Description:** Remove the nth node from the end of the linked list in a single traversal.

## 7. Check if a Linked List is Palindrome

**Description:** Determine whether a given singly linked list is a palindrome by comparing its nodes from both ends.

## 8. Add 1 to a Number Represented by Linked List

**Description:** A linked list represents a number where each node is a digit. Add 1 to this number and return the updated linked list.

## 9. Find the Intersection Point of Two Y-Shaped Linked Lists

**Description:** Determine the node where two linked lists intersect, assuming they form a "Y" shape.

## 10. Find the Middle Element of a Linked List

**Description:** Find the middle node of a linked list using the slow and fast pointer technique.

## 11. Detect a Loop in a Linked List

**Description:** Determine if a given linked list contains a cycle using the Floyd's Cycle Detection Algorithm (slow and fast pointers).

## 12. Find the Length of the Loop

**Description:** If a loop is detected in the linked list, calculate the number of nodes in the loop.

### 13. Delete the Middle Node of a Linked List

**Description:** Remove the middle node of a singly linked list in one traversal without knowing the length.

### 14. Find the Starting Point of Loop in a Linked List

**Description:** If a loop exists, find the node where the cycle begins using Floyd's Cycle Detection Algorithm.

### 15. Delete All Occurrences of a Key in DLL

**Description:** In a doubly linked list, delete all nodes containing a given value.

### 16. Find All Pairs with a Given Sum in DLL

**Description:** Find all pairs of nodes in a doubly linked list whose values add up to a given target.

### 17. Reverse Nodes in k Groups

**Description:** Reverse the nodes of a linked list in groups of size  $k$ . If the number of nodes is not a multiple of  $k$ , leave the remaining nodes as is.

### 18. Rotate a Linked List

**Description:** Rotate the linked list to the right by  $k$  places.

### 19. Merge Two Sorted Linked Lists

**Description:** Merge two sorted linked lists into a single sorted linked list.

### 20. Sort a Linked List using Merge Sort

**Description:** Sort a linked list using the merge sort algorithm.

### 21. Design Browser History using Linked List

**Description:** Implement a browser history system where users can navigate back and forward through visited pages using a doubly linked list.

---

These problem descriptions should help refresh your memory! Let me know if you'd like a deeper dive into any of them. 😊