

Linked List – Set 1

1. Search an element in a Linked List (Iterative and Recursive)

2. Write a function to get Nth node in a Linked List

3. Given a Linked List and a number n, write a function that returns the value at the n'th node from the end of the Linked List.

4. Given a singly linked list, find middle of the linked list.

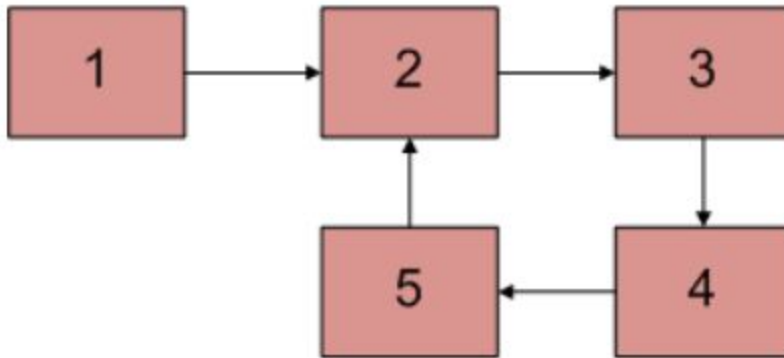
For example, if given linked list is 1->2->3->4->5 then output should be 3.

If there are even nodes, then there would be two middle nodes, we need to print second middle element. For example, if given linked list is 1->2->3->4->5->6 then output should be 4

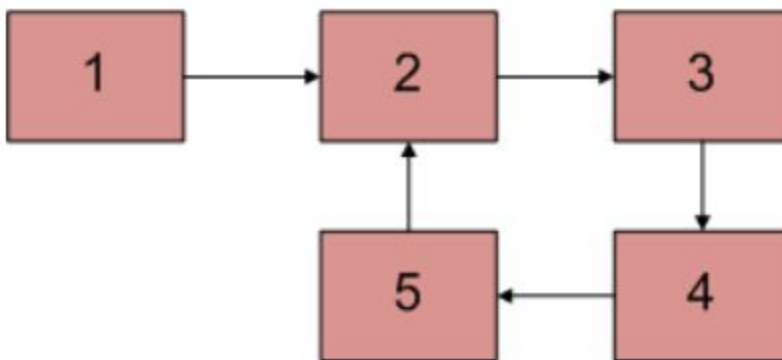
5. Given a singly linked list and a key, count number of occurrences of given key in linked list. For example, if given linked list is 1->2->1->2->1->3->1 and given key is 1, then output

should be 4.

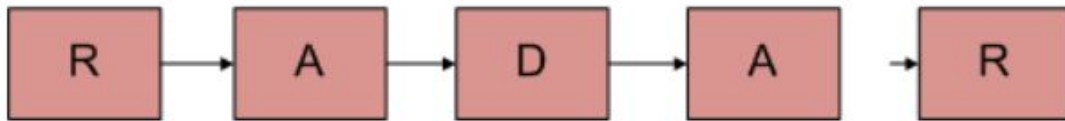
6. Given a linked list, check if the linked list has a loop or not. Below diagram shows a linked list with a loop.



7. Write a function *detectAndCountLoop()* that checks whether a given Linked List contains a loop and if a loop is present then returns the count of nodes in the loop. For example, a loop is present in the below linked list and the length of the loop is 4. If a loop is not present, then the function should return 0.



8. Given a singly linked list of characters, write a function that returns true if the given list is a palindrome, else false.



9. Write a function which takes a list sorted in non-decreasing order and deletes any duplicate nodes from the list. The list should only be traversed once.

For example if the linked list is 11->11->11->21->43->43->60 then `removeDuplicates()` should convert the list to 11->21->43->60.

10. Write a `removeDuplicates()` function which takes a list and deletes any duplicate nodes from the list. The list is not sorted.

For example if the linked list is 12->11->12->21->41->43->21 then `removeDuplicates()` should convert the list to 12->11->21->41->43.

11. Given a linked list and two keys in it, swap nodes for two given keys. Nodes should be swapped by changing links. Swapping data of nodes may be expensive in many situations when data contains many fields.

It may be assumed that all keys in linked list are distinct.

Examples: Input: 10->15->12->13->20->14, x = 12, y = 20

Output: 10->15->20->13->12->14

12. Given a singly linked list, write a function to swap elements pairwise.

For example, if the linked list is 1->2->3->4->5 then the function should change it to 2->1->4->3->5, and if the linked list is 1->2->3->4->5->6 then the function should change it to 2->1->4->3->6->5.

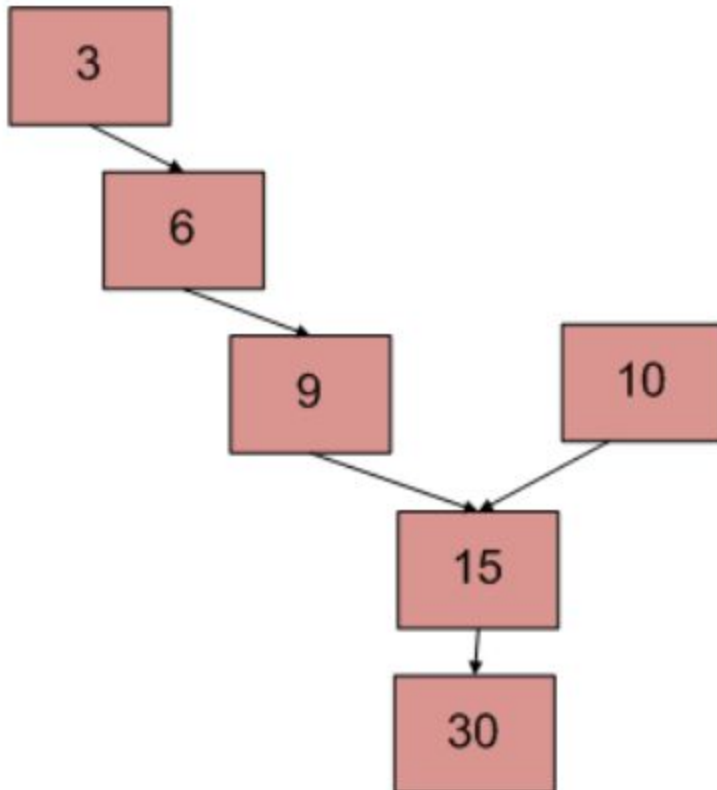
13. Write a function that moves the last element to the front in a given Singly Linked List. For example, if the given Linked List is 1->2->3->4->5, then the function should change the list to 5->1->2->3->4.

14. Given two lists sorted in increasing order, create and return a new list representing the intersection of the two lists. The new list should be made with its own memory — the original lists should not be changed.

For example, let the first linked list be 1->2->3->4->6 and second linked list be 2->4->6->8, then your function should create and

return a third list as 2->4->6.

15. There are two singly linked lists in a system. By some programming error, the end node of one of the linked list got linked to the second list, forming an inverted Y shaped list. Write a program to get the point where two linked list merge.



Above diagram shows an example with two linked list having 15 as intersection point.

16. QuickSort on Singly Linked List

17. Given a Linked List of integers, write a function to modify the

linked list such that all even numbers appear before all the odd numbers in the modified linked list. Also, keep the order of even and odd numbers same.

18. Given pointer to the head node of a linked list, the task is to reverse the linked list. We need to reverse the list by changing links between nodes.

19. Given a linked list, print reverse of it using a recursive function. For example, if the given linked list is 1->2->3->4, then output should be 4->3->2->1.

Note that the question is only about printing the reverse.

20. Given two linked lists sorted in increasing order. Merge them such a way that the result list is in decreasing order (reverse order). Input: a: 5->10->15->40

b: 2->3->20

Output: res: 40->20->15->10->5->3->2

21. Given a linked list, write a function to reverse every k nodes (where k is an input to the function).

Example: Input: 1->2->3->4->5->6->7->8->NULL, K = 3

Output: 3->2->1->6->5->4->8->7->NULL **Input:**

1->2->3->4->5->6->7->8->NULL, K = 5 **Output:**

5->4->3->2->1->8->7->6->NULL

22. Given a linked list, write a function to reverse every alternate k nodes (where k is an input to the function) in an efficient way. Give the complexity of your algorithm.

Example: Inputs: 1->2->3->4->5->6->7->8->9->NULL and k = 3

Output: 3->2->1->4->5->6->9->8->7->NULL.

23. Given a singly linked list, rearrange the list so that even and odd nodes are alternate in the list.

There are two possible forms of this rearrangement. If the first data is odd, then the second node must be even. The third node must be odd and so on. Notice that another arrangement is possible where the first node is even, second odd, third even and so on.

Examples: Input : 11 -> 20 -> 40 -> 55 -> 77 -> 80 -> NULL

Output : 11 -> 20 -> 55 -> 40 -> 77 -> 80 -> NULL

20, 40, 80 occur in even positions and 11, 55, 77 occur in odd positions.

24. Given a Singly Linked List, starting from the second node delete all alternate nodes of it. For example, if the given linked list is 1->2->3->4->5 then your function should convert it to 1->3->5, and if the given linked list is 1->2->3->4 then convert it to 1->3.

25. Write a function AlternatingSplit() that takes one list and divides up its nodes to make two smaller lists 'a' and 'b'. The sublists should be made from alternating elements in the original list. So if the original list is 0->1->0->1->0->1 then one sublist should be 0->0->0 and the other should be 1->1->1.

26. Two Linked Lists are identical when they have same data and arrangement of data is also same. For example Linked lists a (1->2->3) and b(1->2->3) are identical. . Write a function to check if the given two linked lists are identical. a