1. Append a node in a singly link list.

2. Insert a node in Nth position in a singly link list.

3. Delete of a node from a singly link list.

3. Delete a node from Nth position in a singly link list.

4. Insert a node in a sorted singly link list.

5. Front back split {1,2,3,4,5,6} ==>> {1,2,3} && {4,5,6} (Do it in O(n))

6. Remove duplicate node from a singly link list.

7. Alternating split {1,2,3,4,5,6} ==>> {1,3,5} && {2,4,6}

8. Reverse of single link list.(With iteration and Recursion)

9. Delete max and 2nd max element from a single link list.

10. Remove duplicate from a single link list.

11. Shuffle merge of two given link list, {1,2,3} && {4,5,6} ==>> {1,4,2,5,3,6}

12. Sorted merge {1,1,3,4} && {2,3,5}   ==>>> {1,1,2,3,3,4,5}.

13. Check any link list contains cycle or not do it in O (n).

14. Remove nth element from last of a single link list. Do it in O (n).

15. Check palindrome of a given single link list. Do it in O (n).

16. How to find the position where the loop begins in a singly linked list.

17. Alternating merge {1,2,3,4} && {5,6,7} ==>>{1,5,2,6,3,7,4}.

18. Break a single link list in to two half (circular link list) and return head of both.

19. Heads of two singly linked lists are given and these two lists are merging somewhere ...
how can we find node where two lists are merging using method other then brute force method.... that is in nlogn time????

20. We have given two-link list; make a adder (link list) from these two Link list,

L1=4->6->8->9
L2=5->7->2->4
Output=➔L= 1->0->4-> 1 - > 3

21. WAP Intersection, Union, Set diff of two given Link Lists.

22. What will be the code to print the alternate nodes of a linked list in reverse order.....
suppose we have 1->2->3->4->5->6->NULL

print
6 4 2

23. There is linked list where the each node contains 2 pointers. One pointer points to d next node in d list .the other pointer points randomly to another node..U hav to create a clone of that linked list...
they said i can use as much memory as i want. But it has be very efficient..

24. We are given with two link list...at some node they merge...find that node...
E.g.
first list contain: 4 5 7 8 9 10
Second list contain: 3 6 8 9 10
and return address of 8.

25 . Deleting last node in circular singly-linked list

->A->B->C->D->E->F->|

|__________________v

26. You are given a singly link-list such that each node of this list is also a head of another link list of the same type. So, how does one flatten the linked-list.

27.You are given a Linked List and a function declaration as

node* Reverse (node* head, int k);

Reverse is a function that reverses the nodes of a Linked List k at a time and then returns the modified Linked List.

For e.g.

Linked List: 1->2->3->4->5->6->7->8->9->10->11

for k = 2

Return Value: 2->1->4->3->6->5->8->7->10->9->11

for k = 3

Return value: 3->2->1->6->5->4->9->8->7->10->11

28. You are given a Linked List and alternate the position of each node by one.

Given linked list=>>1->2->3->4->5->6

After alternating the position =>> 2->1->4->3->6->5

29. Given a node-A of linked list and a second node-B is given,

1.Then insert node-B just behind the node –A