Question 1: Given a 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 the links between nodes.
Examples:
Input: Head of following linked list
1->2->3->4->NULL
Output: Linked list should be changed to,
4->3->2->1->NULL
Input: Head of following linked list
1->2->3->4->5->NULL
Output: Linked list should be changed to,
5->4->3->2->1->NULL
Input: NULL
Output: NULL

Question 2: Given a linked list of size N. The task is to reverse every k node (where k is an input to the function) in the linked list.
Example 1:
Input:
LinkedList: 1->2->2->4->5->6->7->8
K = 4
Output: 4 2 2 1 8 7 6 5
Explanation:
The first 4 elements 1,2,2,4 are reversed first
and then the next 4 elements 5,6,7,8. Hence, the
resultant linked list is 4->2->2->1->8->7->6->5.

Question 3: Given a linked list of N nodes. The task is to check if the linked list has a loop. Linked list can contain a self loop.
Example 1:
Input:
N = 3
value[] = {1,3,4}
x = 2
Output: True
Explanation: In above test case N = 3.
The linked list with nodes N = 3 is
given. Then value of x=2 is given which
means last node is connected with xth
node of linked list. Therefore, there
exists a loop.

Question 4: You are given a linked list of N nodes. Remove the loop from the linked list, if present.
Note: X is the position of the node to which the last node is connected to. If it is 0, then there is no loop.
Example 1:
Input:
N = 3
value[] = {1,3,4}
X = 2
Output: 1
Explanation: The link list looks like
1 -> 3 -> 4
^ |
|____|
A loop is present. If you remove it
successfully, the answer will be 1.

Question 5: Given a singly linked list consisting of N nodes. The task is to remove duplicates (nodes with duplicate values) from the given list (if exists).
Note: Try not to use extra space. Expected time complexity is O(N). The nodes are arranged in a sorted way.
Example 1:
Input:
LinkedList: 2->2->4->5
Output: 2 4 5
Explanation: In the given linked list
2 ->2 -> 4-> 5, only 2 occurs more
than 1 time.

Question 6: 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.