```
package Love_Babbar;

public class LinkedList {
    class LinkedList {
        // Write a Program to reverse the Linked List. (Both Iterative and recursive)
        /// Reverse a Linked List in group of Given Size. [Very Imp]
        // Write a program to Detect loop in a linked list.
        // Write a program to Delete loop in a linked list.
        // Find the starting point of the loop.
        // Remove Duplicates in a sorted Linked List.
        // Remove Duplicates in a Un-sorted Linked List.
        // Write a Program to Move the last element to Front in a Linked List.
        // Add “1” to a number represented as a Linked List.
        // Add two numbers represented by linked lists.
        // Intersection of two Sorted Linked List.
        // Intersection Point of two Linked Lists.
        // Merge Sort For Linked lists.[Very Important]
        // Quicksort for Linked Lists.[Very Important]
        // Find the middle Element of a linked list.
        // Check if a linked list is a circular linked list.
        // Split a Circular linked list into two halves.
        // Write a Program to check whether the Singly Linked list is a palindrome or
        /// not.
        // Deletion from a Circular Linked List.
        // Reverse a Doubly Linked list.
        // Find pairs with a given sum in a DLL.
        // Count triplets in a sorted DLL whose sum is equal to given value “X”.
        // Sort a “k”sorted Doubly Linked list.[Very IMP]
        // Rotate Doubly Linked list by N nodes.
        // Rotate a Doubly Linked list in group of Given Size.[Very IMP]
        // Can we reverse a linked list in less than O(n) ?
        // Why Quicksort is preferred for. Arrays and Merge Sort for Linked Lists ?
        // Flatten a Linked List
        // Sort a LL of 0’s, 1’s and 2’s
        // Clone a linked list with next and random pointer
        // Merge K sorted Linked list
        // Multiply 2 no. represented by LL
        // Delete nodes which have a greater value on right side
        // Segregate even and odd nodes in a Linked List
        // Program for n’th node from the end of a Linked List
Bit Manipulation:

    }

}
```