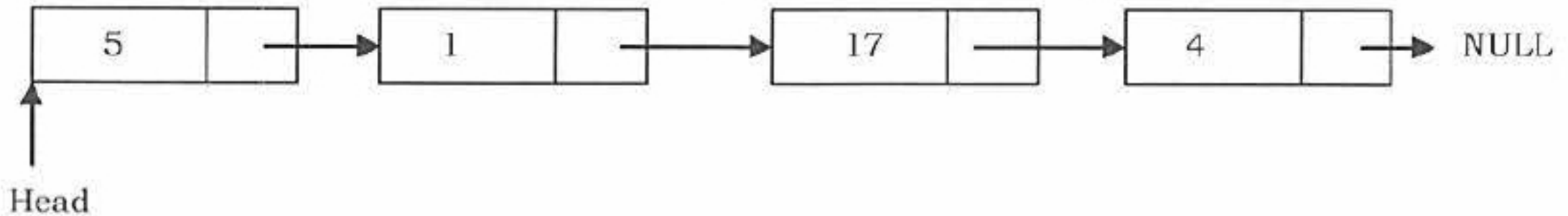
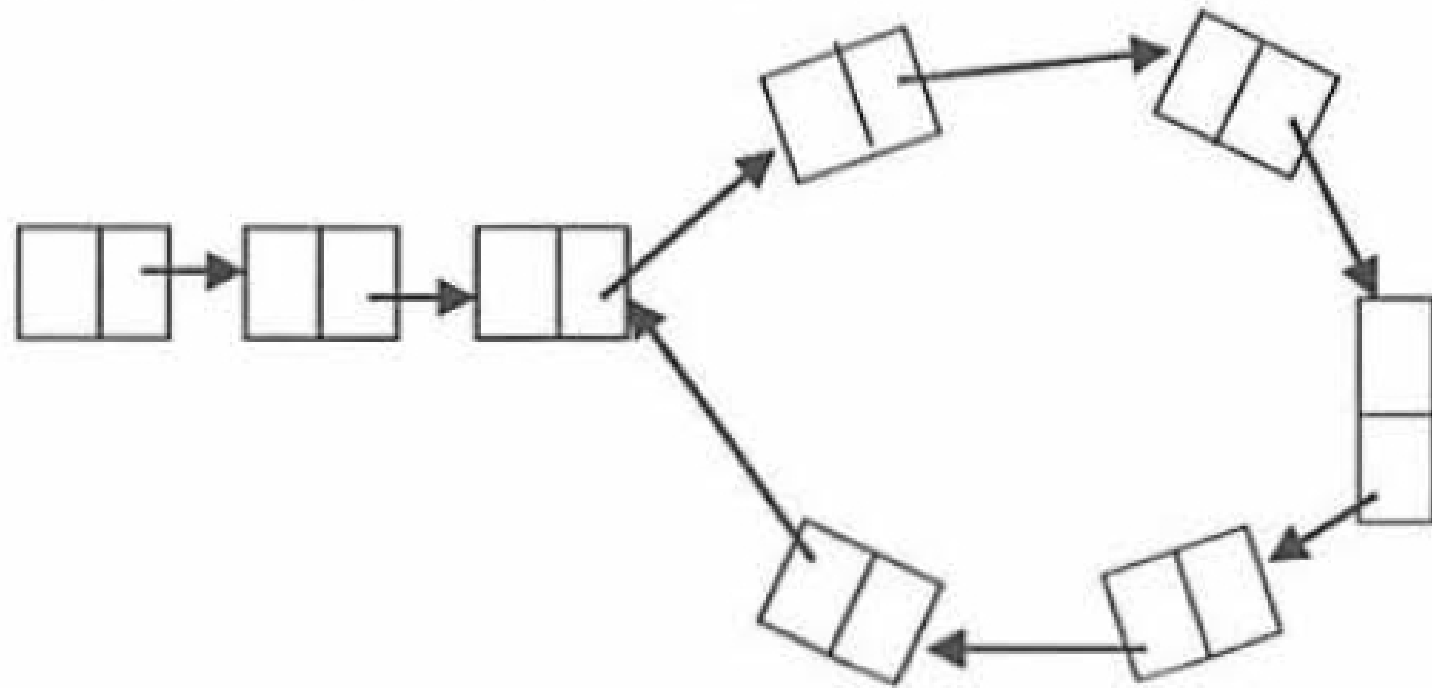


Problem 1: Find the n^{th} node from the end of single linked list. In the example given below, for $n = 3$, 3^{rd} node from end is 1.



Problem 2: Check whether a given linked list has a cycle or not. In the example given below, there is a cycle and it returns True.



Problem 3: Find modular node

- Given a singly linked list, write a function to find the last element from the beginning whose $n \% k == 0$, where n is the number of elements in the list and k is an integer constant.
- For example, if $n = 19$ and $k = 3$, then we should return 18th node.

Problem 4:

Given a sorted linked list, write a program to remove duplicates from it.

Problem 5:

Suppose there are two singly linked lists both of which intersect at some point and become a single linked list. The head or start pointers of both the lists are known, but the intersecting node is not known. Also, the number of nodes in each of the lists before they intersect is unknown and may be different in each list.

List1 may have n nodes before it reaches the intersection point, and List2 might have m nodes before it reaches the intersection point where m and n may be $m = n$, $m < n$ or $m > n$.

Given an algorithm for finding the merging point.

