

LAB EXERCISE 6

- 1.** Write a java function that moves last element to 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.
- 2:** 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.
- 3.** 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.
- 4:** Given a singly linked list, remove all the nodes which have a greater value on right side.
Examples:
a) The list 12->15->10->11->5->6->2->3->NULL should be changed to 15->11->6->3->NULL. Note that 12, 10, 5 and 2 have been deleted because there is a greater value on the right side.
When we examine 12, we see that after 12 there is one node with value greater than 12 (i.e. 15), so we delete 12.
When we examine 15, we find no node after 15 that has value greater than 15 so we keep this node.
When we go like this, we get 15->6->3
b) The list 10->20->30->40->50->60->NULL should be changed to 60->NULL.
Note that 10, 20, 30, 40 and 50 have been deleted because they all have a greater value on the right side.
c) The list 60->50->40->30->20->10->NULL should not be changed.