

Given a singly linked list, remove all the nodes in the list which have any node on their right whose value is greater. (Not just immediate Right , but entire List on the Right)

Example 1:

Input:

LinkedList = 12->15->10->11->5->6->2->3

Output: 15 11 6 3

Explanation: Since, 12, 10, 5 and 2 are the elements which have greater elements on the following nodes. So, after deleting them, the linked list would like be 15, 11, 6, 3.

Example 2:

Input:

LinkedList = 10->20->30->40->50->60

Output: 60

Explanation: All the nodes except the last node has a greater value node on its right, so all the nodes except the last node must be removed.

Your Task:

The task is to complete the function compute() which should modify the list as required and return the head of the modified linked list.

The printing is done by the driver code,

Expected Time Complexity:  $O(N)$

Expected Auxiliary Space:  $O(1)$

Constraints:

$1 \leq \text{size of linked list} \leq 105$

$1 \leq \text{element of linked list} \leq 105$

## Solution:

```
class Solution
{
    Node compute(Node head)
    {
        if (head == null || head.next == null) {
            return head;
        }

        Node current = head;
        Node prev = null;

        while (current != null && current.next != null) {
            boolean shouldDelete = false;
            Node runner = current.next;

            while (runner != null) {
                if (runner.data > current.data) {
                    shouldDelete = true;
                    break;
                }
                runner = runner.next;
            }

            if (shouldDelete) {
                if (prev == null) {
                    head = current.next;
                } else {
                    prev.next = current.next;
                }
            } else {
                prev = current;
            }

            current = current.next;
        }

        return head;
    }
}
```