Delete nodes having greater value on right

Given a singly linked list, remove all the nodes which have a greater value on its following nodes.

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
```

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 ≤ size of linked list ≤ 1000
1 ≤ element of linked list ≤ 1000
Note: Try to solve the problem without using any extra space.

```
def compute(self,head):
        #Your code here
        if head is None or head.next is None:
            return head
        dummyHead = Node(-1)
        prev = dummyHead
        head = self.reverse(head)
        curr = head
        maxEle = 0
        while curr!=None:
            if curr.data>=maxEle:
                prev.next = curr
                prev = curr
```

```
                maxEle = curr.data
            curr = curr.next
        if prev.next!=None:
            prev.next = None
        dummyHead.next = self.reverse(dummyHead.next)
        return dummyHead.next
```