Reverse a Doubly Linked List

Given a doubly linked list of n elements. The task is to **reverse** the doubly linked list.

Example 1:

```
Input: LinkedList: 3 <--> 4 <--> 5
Output: 5 4 3
```

Example 2:

```
Input: LinkedList: 75 <--> 122 <--> 59 <--> 196
Output: 196 59 122 75
```

Your Task:
Your task is to complete the given function **reverseDLL**(), which takes **head** reference as argument and should **reverse** the elements so that the tail becomes the new head and all pointers are correctly pointed. You need to **return** the **new head** of the reversed list. The **printing** and **verification** is done by the **driver** code.

Expected Time Complexity: O(n).
Expected Auxiliary Space: O(1).

Constraints:

1 <= number of nodes <= 10^3

0 <= value of nodes <= 10^3

```python
def reverseDLL(head):
    #return head after reversing
    if head is None or head.next is None:
        return head
    curr = head
    last = None
    forward = None

    while curr!=None:
        forward = curr.next
        curr.next = last
        curr.prev = curr.next
        last = curr
        curr = forward
    return last
```