DAY-29

(1). Rotate a linked list :-
(rotate clockwise by k nodes).

Approach :

→ need to change the kth node to NULL,

→ next of the last node to the previous head node,

→ change head to (k+1)th node.

* need to hold three nodes :- kth node, (k+1)th node and last node.

→ Traverse the linked list

→ store pointer to kth node.

→ We can get k+1 node by using kth node → next,

→ Get and store the last node after traversing.

* then, finally change the pointers.

Rotate part Implementation :-

```
def rotate(self, k):
    if k == 0:
        return
    current = self.head
    count = 1
    while (count < k and current is not None):
        current = current.next
        count += 1
    if count is None:
        return
    kthNode = current
```

```
while (current.next is not None):
    current = current.next
current.next = self.head
self.head = kthNode.next
kthNode.next = None
```

② Merge two linked lists (sorted)

Recursion implementation:

```
def mergeLists(head1, head2):
    temp = ~~Null~~ None
    if head1 is None:
        return head2
    if head2 is None:
        return head1

    if head1.data <= head2.data:
        temp = head1
        temp.next = mergeLists(head1.next,
                               head2)
    else:
        temp = head2
        temp.next = mergeLists(head1,
                               head2.next).
    return temp.
```