Partition a Linked List around a given value

Given a linked list and a value x, partition it such that all nodes less than x come first, then all nodes with value equal to x and finally nodes with value greater than or equal to x. The original relative order of the nodes in each of the three partitions should be preserved. The partition must work in-place.

Example 1:

```
Input:
1->4->3->2->5->2->3,
x = 3
Output:
1->2->2->3->3->4->5
Explanation:
Nodes with value less than 3 come first,
then equal to 3 and then greater than 3.
```

Example 2:

```
Input:
1->4->2->10
x = 3
Output:
1->2->4->10
Explanation:
Nodes with value less than 3 come first,
then equal to 3 and then greater than 3.
```

Your task:
You don't need to read input or print anything. Your task is to complete the function **partition()** which takes the head of the inked list and an integer x as input, and returns the head of the modified linked list after arranging the values according to x.

Expected time complexity : O(n)
Expected Auxiliary Space: O(n)

Constraints:
$1 <= N <= 10^5$
$1 <= k <= 10^5$

Code : https://sapphireengine.com/@/5xxd20

https://practice.geeksforgeeks.org/problems/partition-a-linked-list-around-a-given-value/1/?category[]=Linked%20List&category[]=Linked%20List&difficulty[]=1&difficulty[]=2&page=1&query=category[]Linked%20Listdifficulty[]1difficulty[]2page1category[]Linked%20List#