# Partition List

**Question**: Given a linked list and a value x, partition it such that all nodes less than x come before nodes greater than or equal to x.

You should preserve the original relative order of the nodes in each of the two partitions.

For example,

Given 1->4->3->2->5->2 and x = 3,

return 1->2->2->4->3->5

**Solutions:**

```
class ListNode:
    def __init__(self, x, next=None):
        self.val = x
        self.next = next


class Solution:
    # @param head, a ListNode
    # @param x, an integer
    # @return a ListNode
    def partition(self, head, x):
        smaller = ListNode(-1)
        others = ListNode(-1)

        smallerLast, othersLast = smaller, others
        while head != None:
```

```
            if head.val < x:
                smallerLast.next = head
                smallerLast = smallerLast.next
            else:
                othersLast.next = head
                othersLast = othersLast.next
            head = head.next

        smallerLast.next = others.next
        othersLast.next = None
        return smaller.next

    def printll(self, node):
        while node:
            print ( node.val )
            node = node.next

if __name__ == '__main__':
    node6 = ListNode(2)
    node5 = ListNode(5, node6)
    node4 = ListNode(2, node5)
    node3 = ListNode(3, node4)
    node2 = ListNode(4, node3)
    ll1 = ListNode(1, node2)
    Solution().printll( Solution().partition(ll1,3) )
```