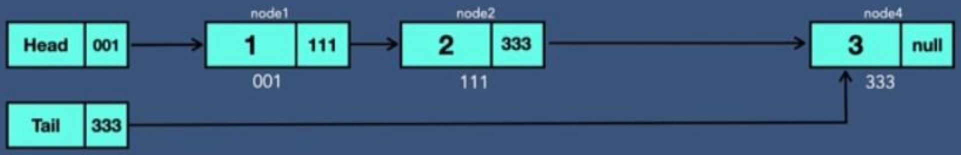
Remove Duplicates

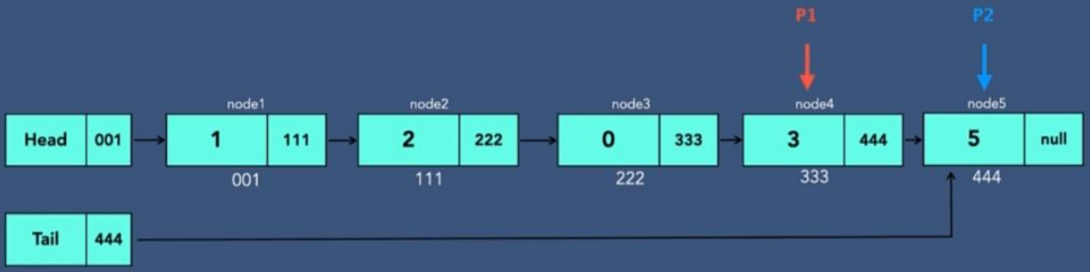
Write a method to remove duplicates from an unsorted linked list.



```
currentNode = node1
hashSet =  {1} → {1,2} → {1,2,3}
while currentNode.next is not Null
    If next node's value is in hashSet
        Delete next node
    Otherwise add it to hashSet
```

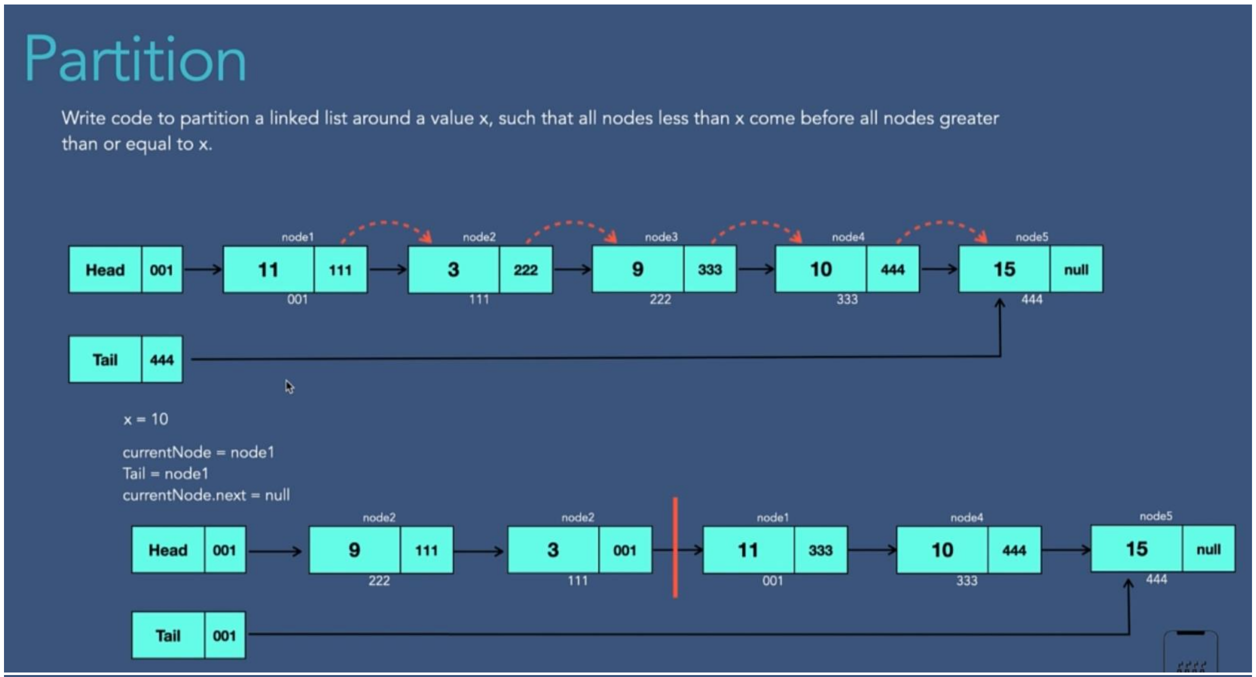
Return Nth to Last

Implement and algorithm to find the nth to last element of a singly linked list.



N = 2 ⟶ Node4

```
pointer1 = node1                pointer2 = node2
         = node2                         = node3
         = node3                         = node4
         = node4                         = node5
```

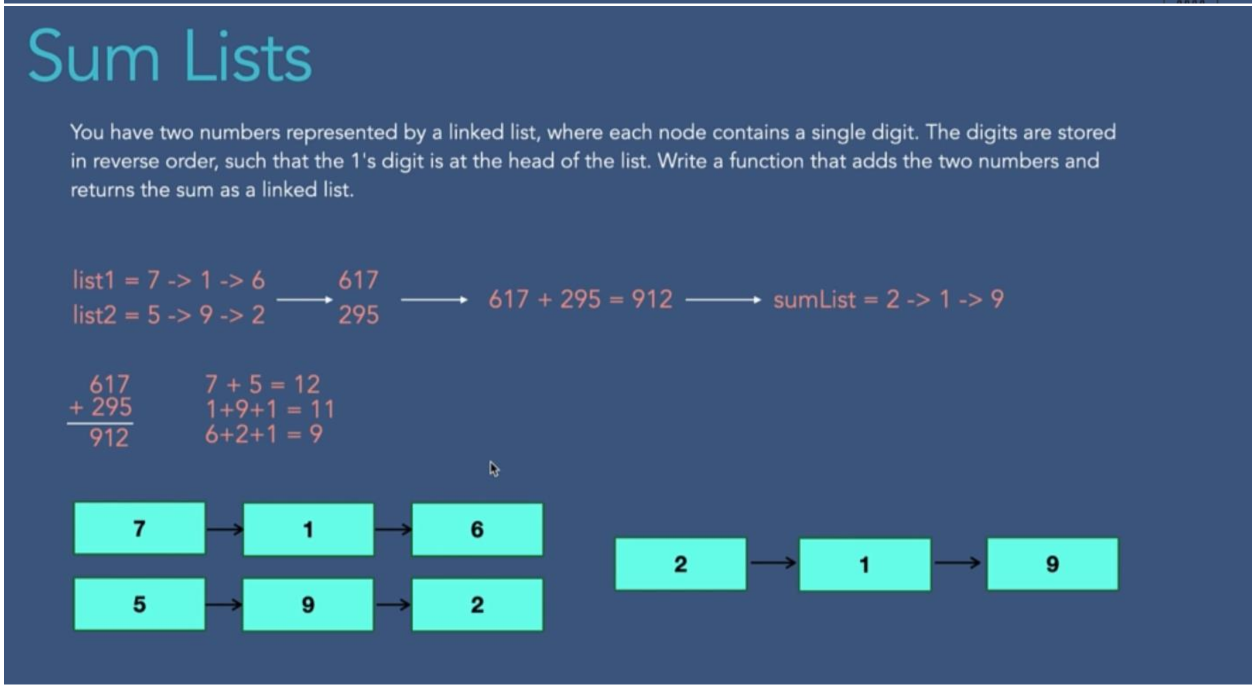


Partition

Write code to partition a linked list around a value x, such that all nodes less than x come before all nodes greater than or equal to x.

x = 10

currentNode = node1
Tail = node1
currentNode.next = null



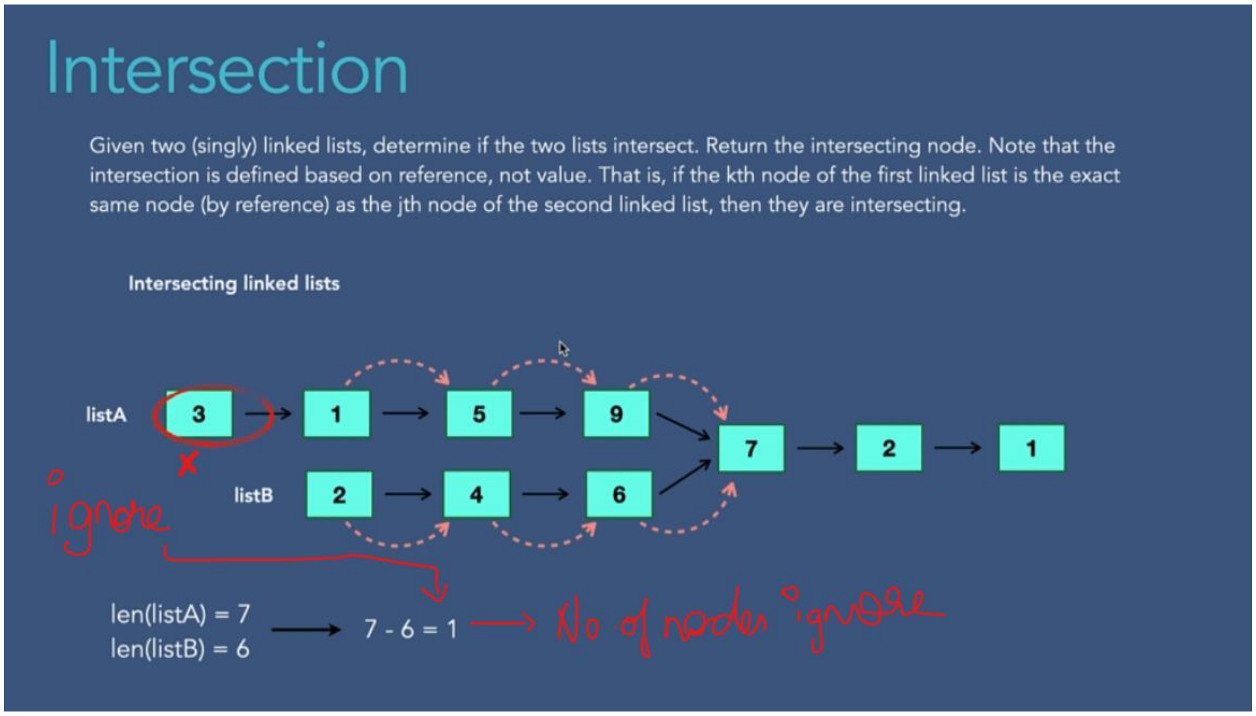
Sum Lists

You have two numbers represented by a linked list, where each node contains a single digit. The digits are stored in reverse order, such that the 1's digit is at the head of the list. Write a function that adds the two numbers and returns the sum as a linked list.

list1 = 7 -> 1 -> 6
list2 = 5 -> 9 -> 2 $\longrightarrow$ 617 / 295 $\longrightarrow$ 617 + 295 = 912 $\longrightarrow$ sumList = 2 -> 1 -> 9

$$\begin{array}{r} 617 \\ +\,295 \\ \hline 912 \end{array}$$

7 + 5 = 12
1+9+1 = 11
6+2+1 = 9



Intersection

Given two (singly) linked lists, determine if the two lists intersect. Return the intersecting node. Note that the intersection is defined based on reference, not value. That is, if the kth node of the first linked list is the exact same node (by reference) as the jth node of the second linked list, then they are intersecting.

Intersecting linked lists

len(listA) = 7 → 7 - 6 = 1 → No of nodes ignore
len(listB) = 6