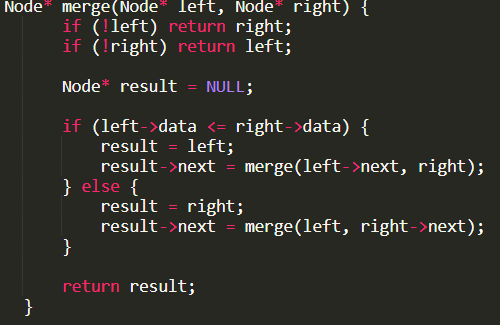
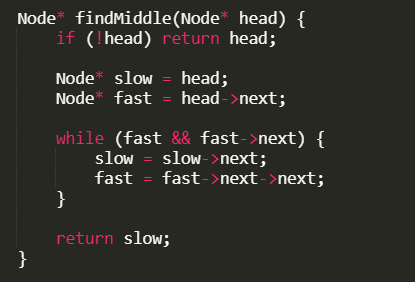
# LinkedList

* **Merging of two Linked List:**

Recursive Approach:-

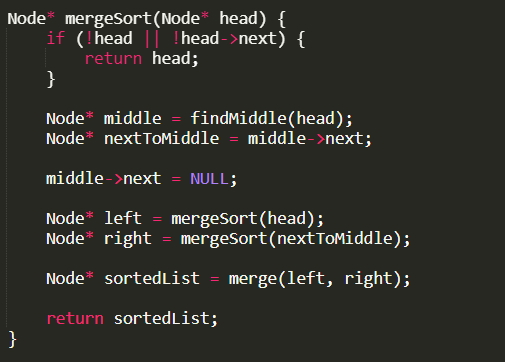


* **Find the Middle of the Linked List:**

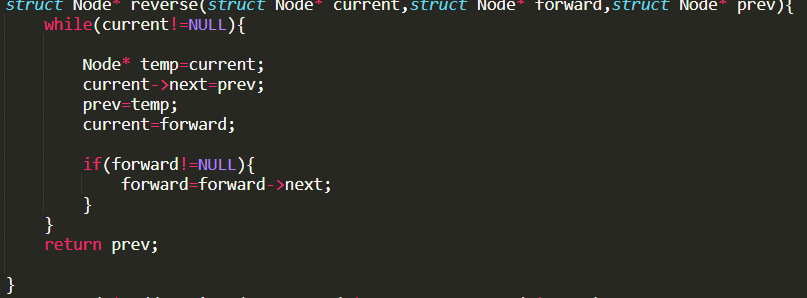


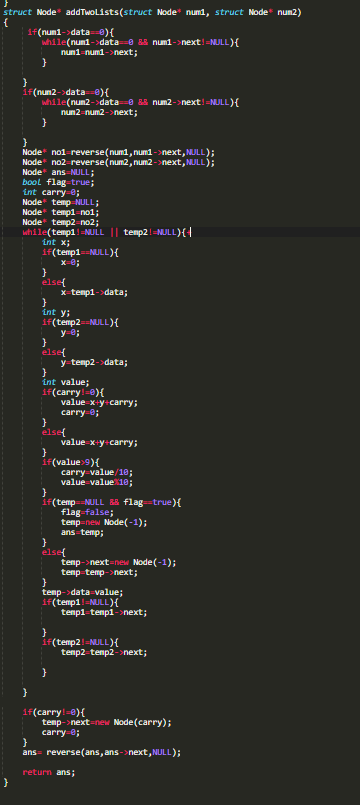
* **LinkedList X Sorting:-**

**1)Merge Sort in Linked list**



Above three images are the part of this combined merge sort code. Here directly breaking of the linked list is carried out which we do not do in case of the array.

* **Reversing the Linked List**
* **Add Two Nos**



**General Algo:-**

**1)Function Definition:**

Define a function addTwoLists that takes two linked list nodes num1 and num2 as arguments.

**2)Remove Leading Zeros:**

If num1's data is 0, iterate through num1 until a non-zero node is found.

If num2's data is 0, iterate through num2 until a non-zero node is found.

**3)Reverse the Linked Lists:**

Reverse num1 and store the result in nol.

Reverse num2 and store the result in no2.

**4)Initialize Variables:**

Initialize a new node ans to NULL.

Initialize carry to 0 and flag to true.

Initialize temp, temp1, and temp2 to point to nol, no2, and NULL respectively.

**5)Iterate and Add:**

While either temp1 or temp2 is not NULL, perform the following steps:

Initialize x and y to 0.

If temp1 is not NULL, set x to temp1's data.

If temp2 is not NULL, set y to temp2's data.

Calculate value as the sum of x, y, and carry.

Update carry based on the value of value (if value is greater than 9, set carry to 1 and adjust value to be within 0-9).

If temp is NULL and flag is true:

Set flag to false.

Create a new node with data value and assign it to temp.

Set ans to point to temp.

Else, create a new node with data value and link it to temp.

Move temp1 and temp2 to their respective next nodes.

**6)Handle Carry:**

If carry is greater than 0, create a new node with data carry and link it to temp.

**7)Reverse the Resultant List:**

Reverse the ans list to get the final result.

**8)Return Result:**

Return the ans node.