Sum Lists

617 + 295 = 912

So, the linked will be given Like this

7 → 1 → 6
5 → 9 → 2

→ output should be → 2 → 1 → 9

and after this try it if it is given in reverse order.

6 → 1 → 7
2 → 9 → 5 → output → 9 → 1 → 2

→ We need to keep this in mind that the length of linked list may be different.

→ lets try with equal length linked List

carry = 0

we need a carry

7 → 1 → 6
5 → 9 → 2

12 → carry   c + 1 + 9 → 11   9

New list → 2 → 1 → 9

carry = ~~0~~ ~~1~~ ~~1~~ 0

carry is 0 ← So, no need to add another element

Stop here

Now, → 3 → 5 → 6
7 → 2 → 1 → 3

10   7+1 = 8   6+1 +0 = 7

carry = ~~0~~ ~~1~~ ~~0~~ 0

Now ~~check~~ which one is not empty. and then keep adding & updating carry to that.

0 → 8 → 7

→ Now lets try this with recursion

7 → 1 → 6
5 → 9 → 2

Recursion function

```
LinkedList SumList(l1, l2, carry);

SumList (7, 5, 0) {
  int Sum = 7 + 5;
   carry = 12 / 10 = 1
   Sum = Sum % 10;
   Linked List = SumList(7.next, 5.next, 1)
    Node temp = new node (Sum);
     Node temp2 = LL.head;
      ll.head = temp.
       ll head.next = temp2;
       return LinkedList
```

→ we will also check if any one is null

→ Now we need to do this if it was in forward order

1 → 2 → 3
5 → 4 → 6

6 → 6 → 9 → like this

→ Here we can do it by Recursion using backtracking. & a wrapper class

→ wrapper class will have a node & carry.

→ Here as the length is not equal we will make length equal by adding 0 to starting → This is a possible way.

or by always passing null.

→ we need to add 0 at starting to equate the lengths