LeetCode #2

→ Day-5

Adding 2 Numbers.

✡ Given

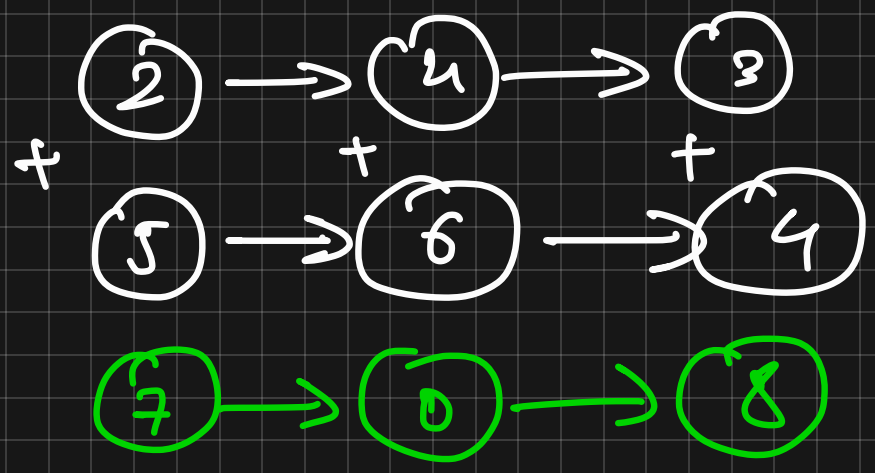
→ 2 Non-empty Linked-List.

→ Each LL represents a non-negative Integer

→ Each Integer is represented in a such way that the digits are reversed.

Task

→ Add Both the LL in place and return a new LL representing the sum of both the integers.



Approach

① To add in-place, we can get the values of each Node and add them together + carry while maintaining a carry

② Calculating Carry

We can Calculate Carry using the total Sum of the digits.

```
Carry = sum/10;
```

This takes the "integer" part of division and assign it to the "Carry" variable.

③ Create a newnode with digit Calculated from the Sum. [digit = Sum % 10;]

```
newNode(digit);
```

④ Attach the NewNode with tail "dummyHead" node of the result list.

⑤ Move the tail pointer to the newly added node.

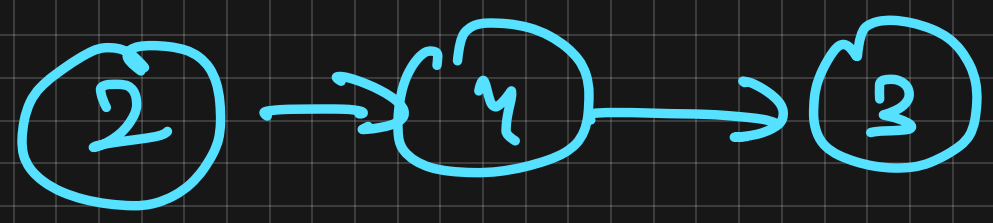
⑥ Move to the next nodes in l1 and l2 if they exist. If either List is exhausted set the list to "null ptr".

⑦ After the loop obtain actual result by skipping the dummy head node.

⑧ Delete Dummy Head.

⑨ Return Result.

DRY RUN

[2, 4, 8] 

[5, 6, 4] 

dummyHead = new Node(0);
tail = dummyHead
int carry = 0

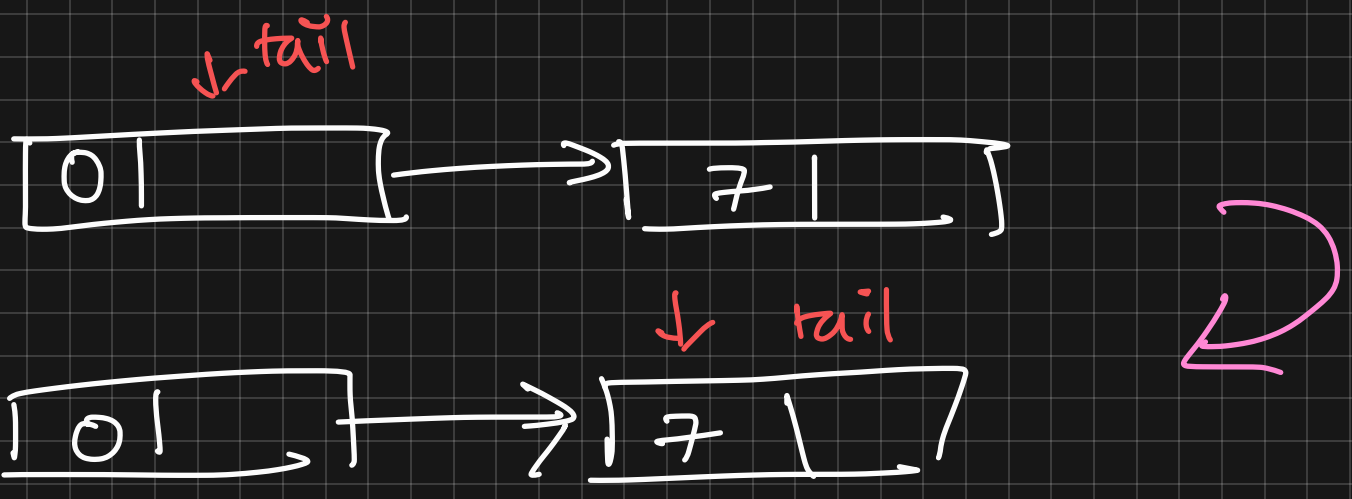
while (l1 != Null OR l2 != NULL OR carry != 0){

int digit 1 = 2
digit 2 = 5
sum = 2+5+0
digit = 7 % 10 (sum % 10)
= 7
carry = 7/10 (sum / 10)
= 0.7
= 0

New Node => 

dummyNode → newNode


tail pointer moves from dummy to newNode.



Iteration #2

Current dummyHead = [0] → [7] [0]

digit 1 = 4

digit 2 = 6

Sum = 4 + 6 + 0

digit = 10 / 10

= 0

Carry = 10 / 10

= 1

newNode = new Node(digit)

tail -> next = newNode.

tail = tail -> next;

Iteration #3

digit 1 = 3

digit 2 = 4

sum = 3+4+ Carry

= 3+4+1

= 8

digit = sum % 10

= 8 % 10

= 8

Carry = 8/10

= 0.8

= 0

newNode = new Node(digit)

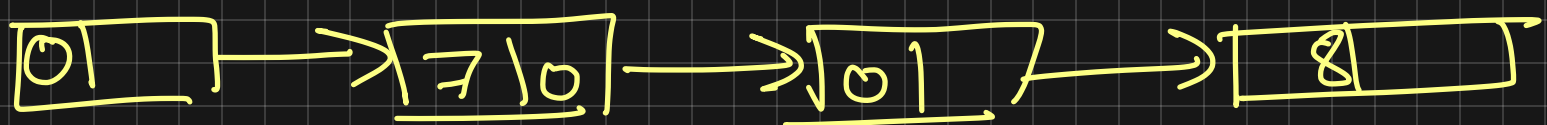
tail -> next = newNode.

tail = tail -> next;

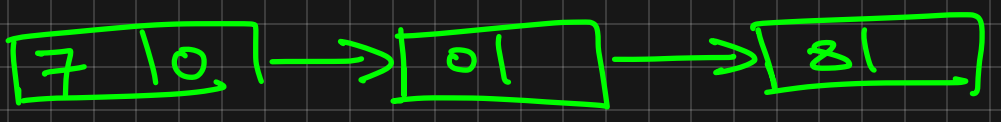
Now Its core Iterating as no nodes are left

and Carry is left to be added.

So resultant dummy Head is



result = dummyHead -> next

{  }

delete (dummyHead);

return result;