① ② → ④ → ③   trinath

⑤ → ⑥ → ④   thought

⑦ → ⑥ → ⑦   process

___  ___  ___  ___

case ② :)

⑨ → ⑨ → ⑨ → ⑨ → ⑨ ⑨ ⑨

⑨ → ⑨ → ⑨ → ⑨

⑧ → ⑨ → ⑨ → ⑨ → 0 → ⑥ → ⑥

8, 9, 9, 9, 0, 0, 6, 1

5 → 6 → 7

3 → 7 → 9

0 → 9 → 0 → 8

13 2) $n/10 = 3$

$no\%10 = 1$

2 → 4 → 3

5 → 6 → 4

new Node

t3 = node

Thought Process ⑦

| 7 | null |

↳ temp

newNode • next = temp
temp = newNode

Sum = 0;

while ( l1 != null & l2 != null )

Sum = l1 • val + l2 • val + Sum;

if (Sum < 9) {
  newNode = Node but (Sum)
  3 Sum = 0;

```
Else {
    new Node = node (sum%(0)
    sum = sum / (0;
}
```

```
IF ( AnsNode == node ) {
    AnsNode = newNode;
} Else {
```

```
        newNode.next = AnsNode
        AnsNode = newNode;
    }

    l1 = l1.next;

    l2 = l2.next;


3 while end.

while ( lc != null ) {
Sum = Sum + l1. val
if (Sum > 9) {
newNode = new val (Sum%10
```

```
            sum = sum / 10;
    }
    else{ new node = new N (sum)
            sum = 0;
    }
    newNode.next = AnsNode
        AnsNode = new Node;

        l1 = l1.next.

}
    &while( l2 != null){

    sum = sum + l2.val;

    If (sum > 9) {
    newNode = new (sum/1f)
```

```
        Sum = Sum/10;
}   else{
        newNode = new (Sum)
            Sum =0;
}

    newNode.next = Ansnode
    AnsNod = newNode;

    l2 = l2.next;

}
}
```