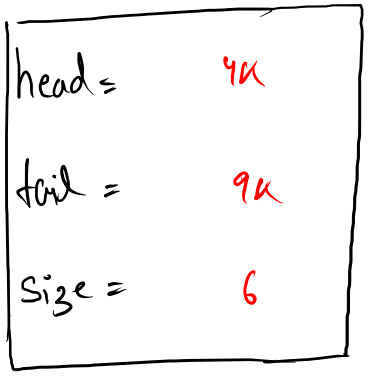
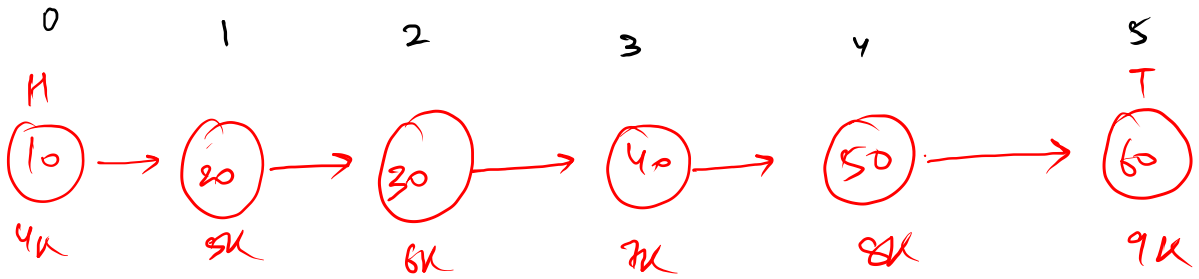
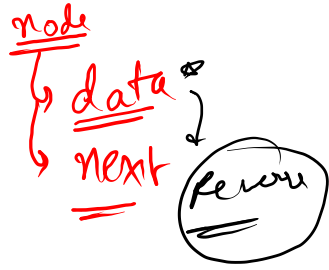
- </> Reverse A Linked List (data Iterative)
- </> Reverse Linked List (pointer Iterative)
- </> Display Reverse (recursive) - Linked List
- </> Reverse Linked List (pointer - Recursive)
- </> Add Two Linked Lists





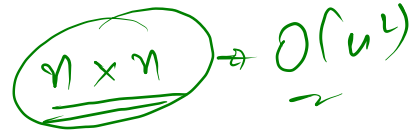


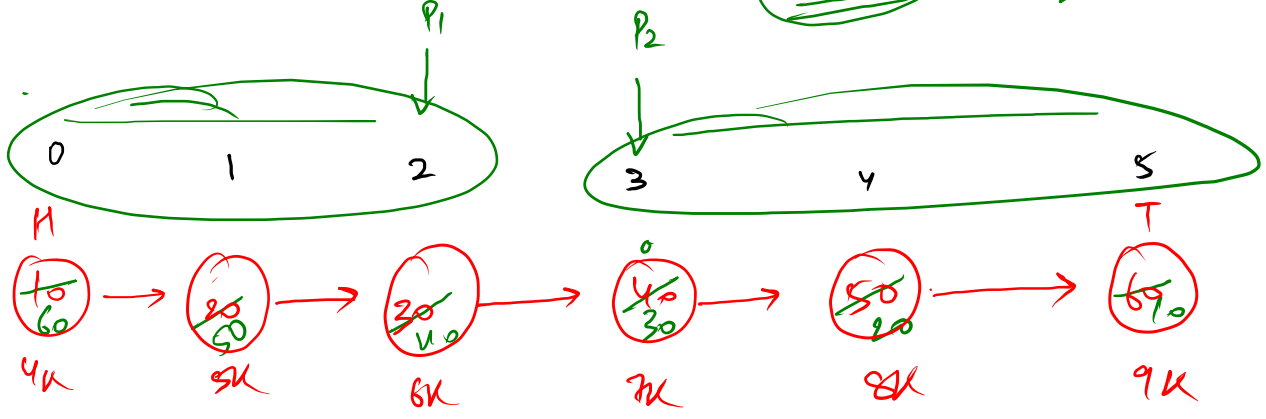
$P_1 \Rightarrow 0$

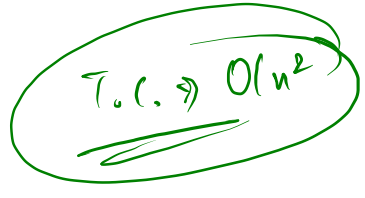
$P_2 \Rightarrow Size - 1$

while ($P_1 < P_2$){

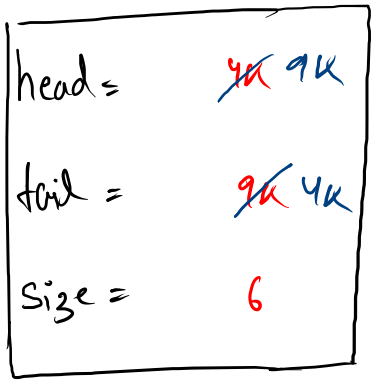
}

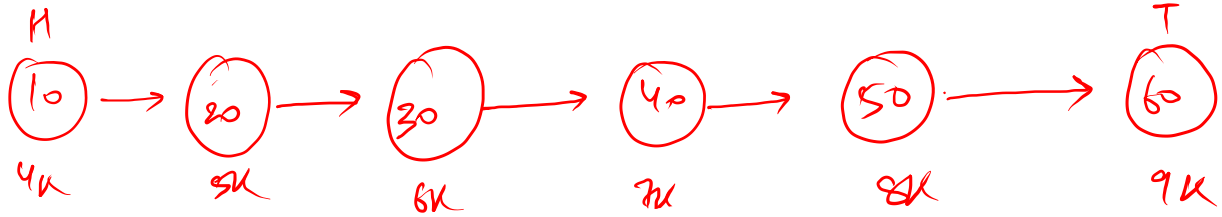







```java
public Node getNodeAt(int idx) {
  if (size == 0) {
    System.out.println("List is empty");
    return null;
  } else if (idx < 0 || idx >= size) {
    System.out.println("Invalid arguments");
    return null;
  } else {
    Node temp = head;
    for (int i = 0; i < idx; i++) {
      temp = temp.next;
    }
    return temp;
  }
}
public void reverseDI() {
  int p1 = 0 , p2 = this.size-1;

  while(p1 < p2){
      Node n1 = getNodeAt(p1);
      Node n2 = getNodeAt(p2);

      int tmp = n1.data;
      n1.data = n2.data;
      n2.data = tmp;

      p1++;
      p2--;
  }
}
```

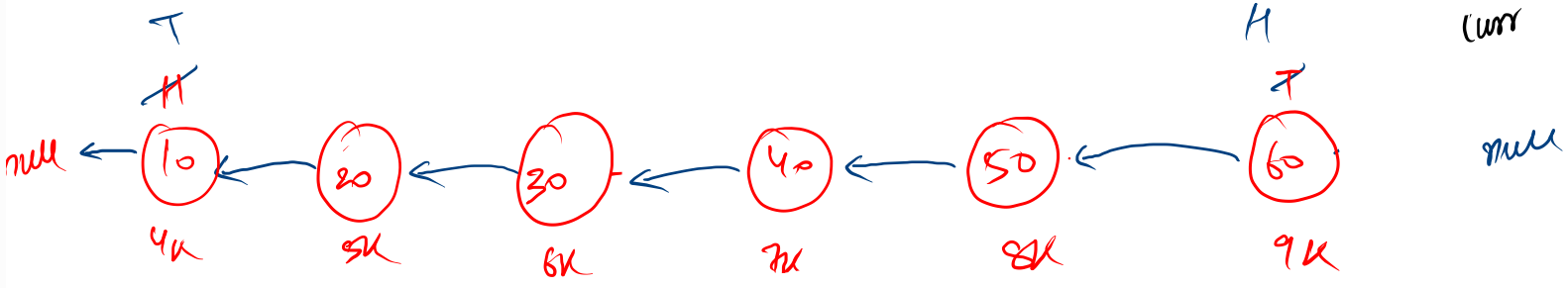





```
Node fwd = curr.next;
curr.next = prev
prev = curr
curr = fwd
```

prev

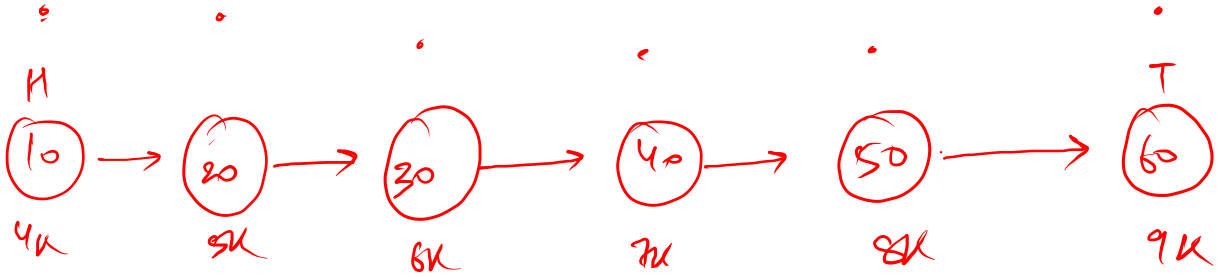
```
public void reversePI(){
  Node curr = this.head , prev = null;

  while(curr != null){
      Node fwd = curr.next;
      curr.next = prev;
      prev = curr;
      curr = fwd;
  }

  Node tmp = this.head;
  this.head = this.tail;
  this.tail = tmp;
}
```



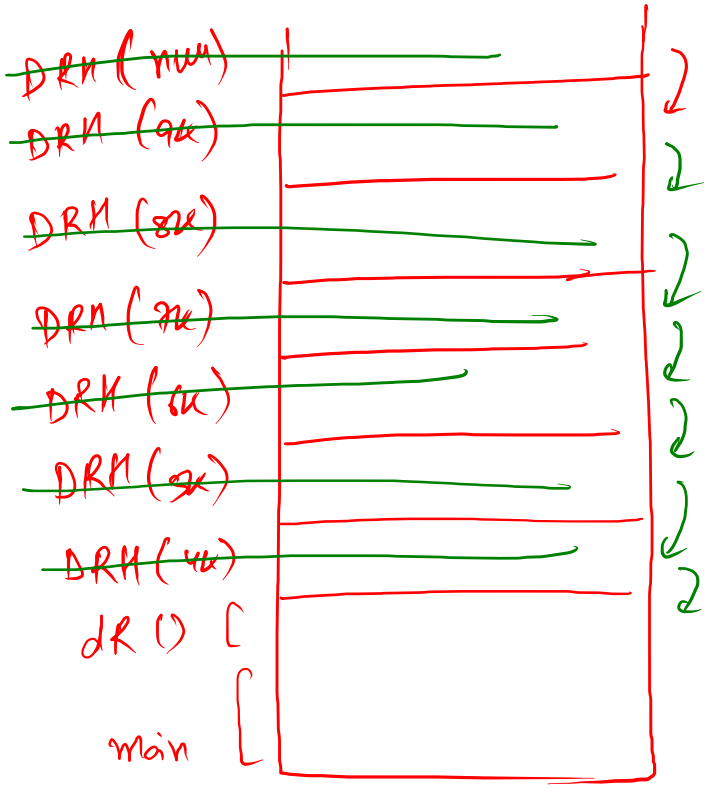
Display Reverse

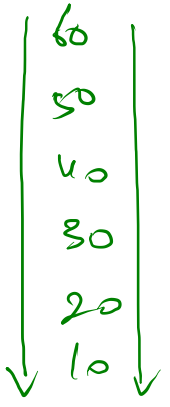
Implement

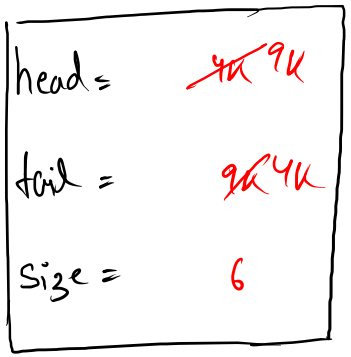
head =	4K
tail =	9K
Size =	6

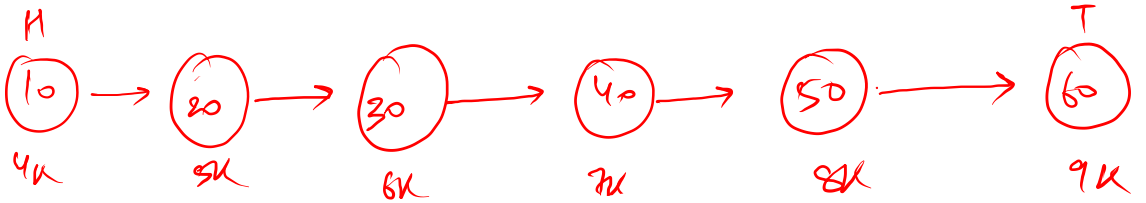


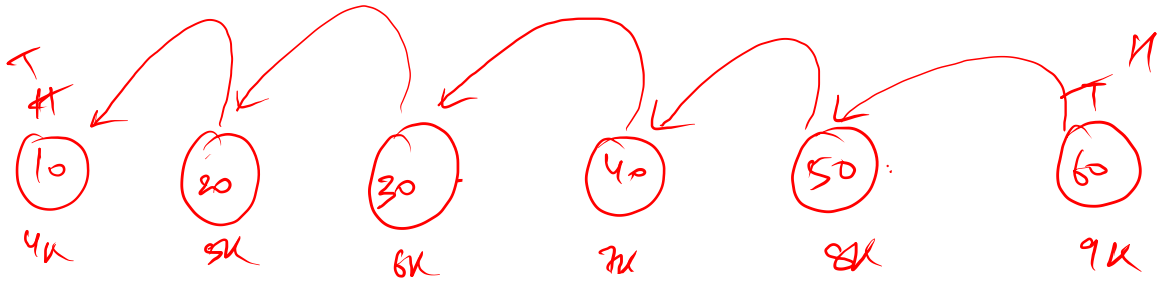

```java
private void displayReverseHelper(Node node){
  // write your code here
}

public void displayReverse(){
  displayReverseHelper(head);
  System.out.println();
}
```



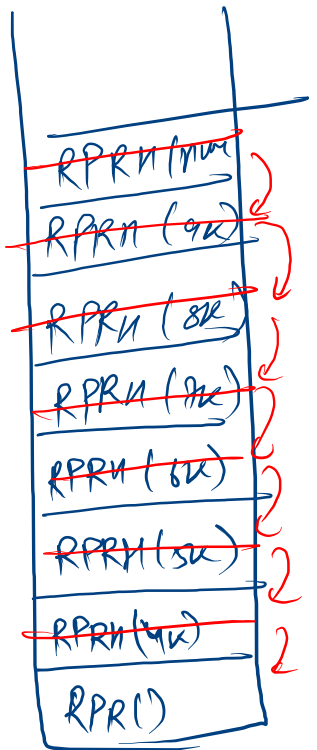


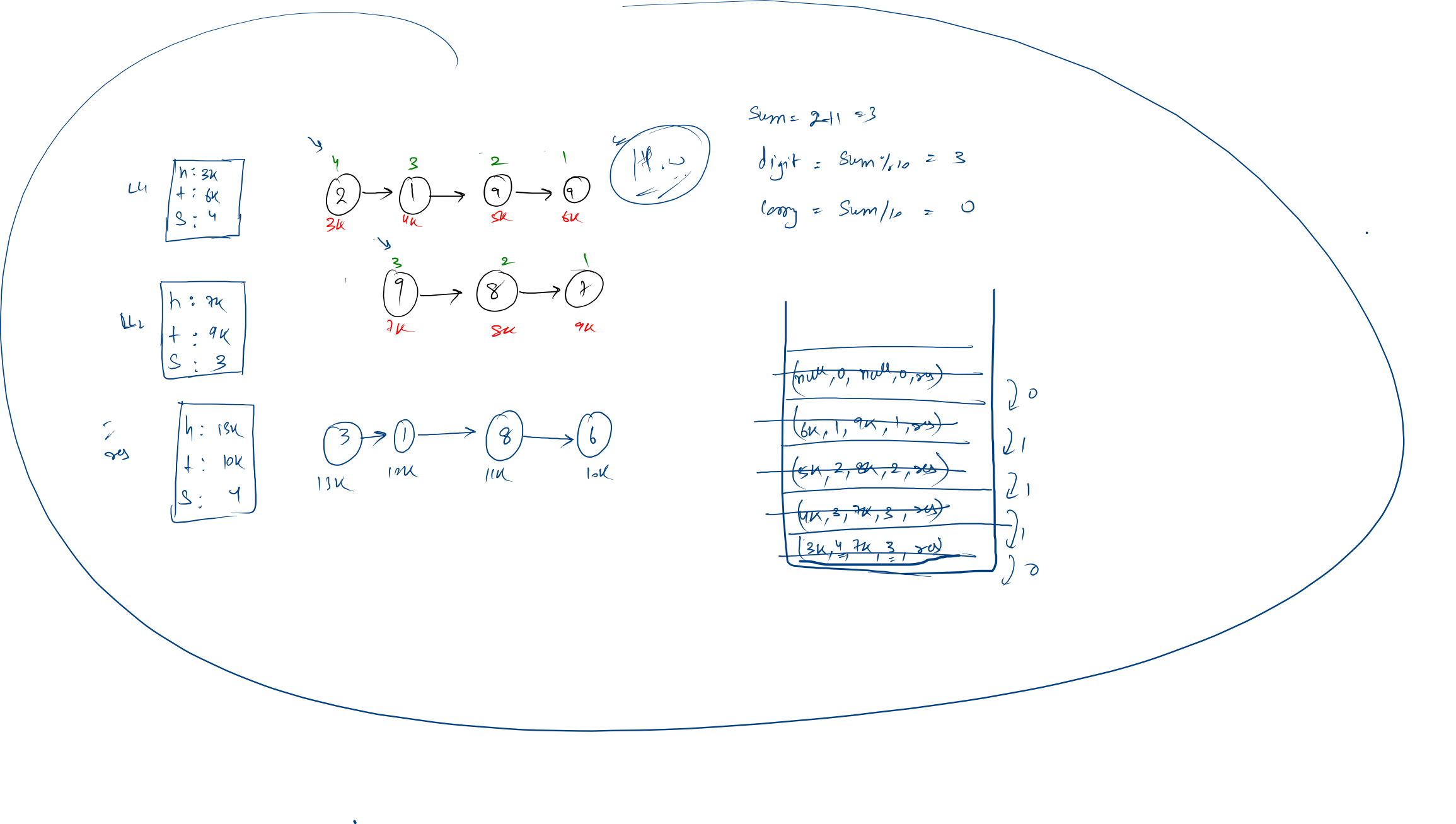







```java
private void reversePRHelper(Node node){
  if(node == null){
    return;
  }
  reversePRHelper(node.next);
  if(node != tail){
    node.next.next = node;
  }
}

public void reversePR(){
  reversePRHelper(this.head);

  this.head.next = null;

  Node tmp = this.head;
  this.head = this.tail;
  this.tail = tmp;
}
```




LL1
h: 3K
t: 6K
S: 4
LL2
h: 7K
t: 9K
S: 3
LL3
h: 13K
t: 10K
S: 4
2 → 1 → 9 → 9
3K 4K 5K 6K
9 → 8 → 7
7K 8K 9K
3 → 1 → 8 → 6
13K 10K 11K 10K
Sum = 2+1 = 3
digit = Sum % 10 = 3
carry = Sum / 10 = 0
(null, 0, null, 0, 0)
(6K, 1, 9K, 1, 0)
(5K, 2, 8K, 2, 0)
(4K, 3, 7K, 3, 0)
(3K, 4, 7K, 3, 0)
0
1
1
1
0