class Solution {

public int romanToInt(String S) {

int ans = 0, num = 0;

for (int i = S.length()-1; i >= 0; i--) {

switch(S.charAt(i)) {

case 'I': num = 1; break;

case 'V': num = 5; break;

case 'X': num = 10; break;

case 'L': num = 50; break;

case 'C': num = 100; break;

case 'D': num = 500; break;

case 'M': num = 1000; break;

}

if (4 \* num < ans) ans -= num;

else ans += num;

}

return ans;

}

}

—---------------

import java.util.HashMap;

class Solution {

public ListNode addTwoNumbers(ListNode l1, ListNode l2) {

ListNode head = new ListNode(0);

head.val = generateSumList(l1.next, l2.next, head.next);

return head;

}

public int generateSumList(ListNode l1, ListNode l2, ListNode res) {

int rest, sum;

if (l1.next == null && l2.next != null) {

return generateSumList(l1, l2.next, res.next);

}

if (l1.next != null && l1.next == null) {

return generateSumList(l1.next, l2, res.next);

}

if (l1.next == null && l2.next == null) {

sum = l1.val + l2.val;

if (sum > 9) {

ListNode n = new ListNode(sum % 10, null);

res = n;

return 1;

}

else {

ListNode n = new ListNode(sum, null);

res = n;

return 0;

}

}

rest = generateSumList(l1.next, l2.next, res.next);

sum = l1.val + l2.val + rest;

if (sum > 9) {

res.val = sum % 10;

return 1;

}

else {

res.val = sum;

return 0;

}

}

}