Luis Gerardo Bravo Ramones

A01282014

100. Same Tree

class Solution {

public:

bool isSameTree(TreeNode\* p, TreeNode\* q) {

if (p == NULL && q == NULL) return true;

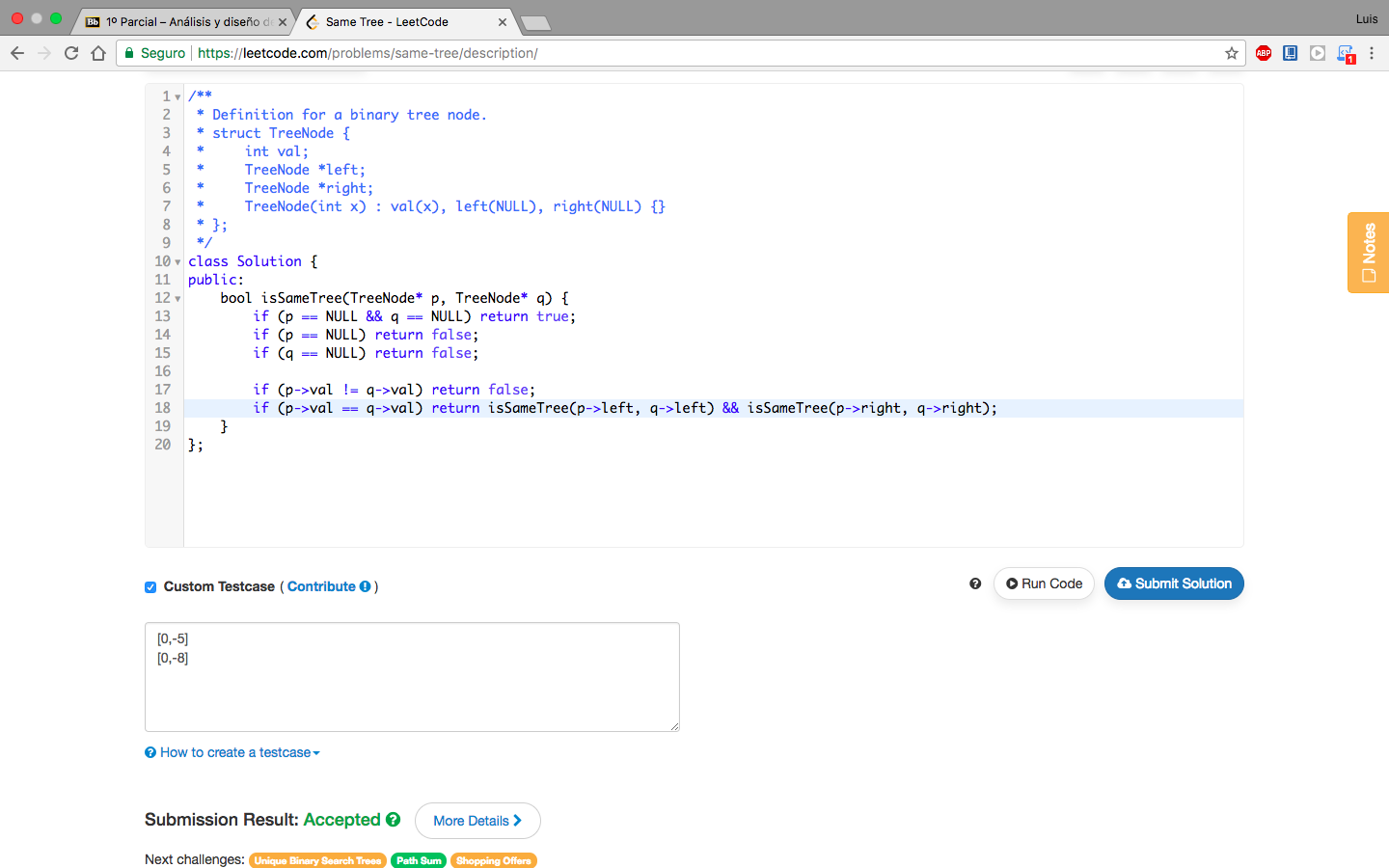
if (p == NULL) return false;

if (q == NULL) return false;

if (p->val != q->val) return false;

if (p->val == q->val) return isSameTree(p->left, q->left) && isSameTree(p->right, q->right);

}

};

672. Merge two binary tres

class Solution {

public:

TreeNode\* mergeTrees(TreeNode\* t1, TreeNode\* t2) {

if (t1 == NULL) return t2;

if (t2 == NULL) return t1;

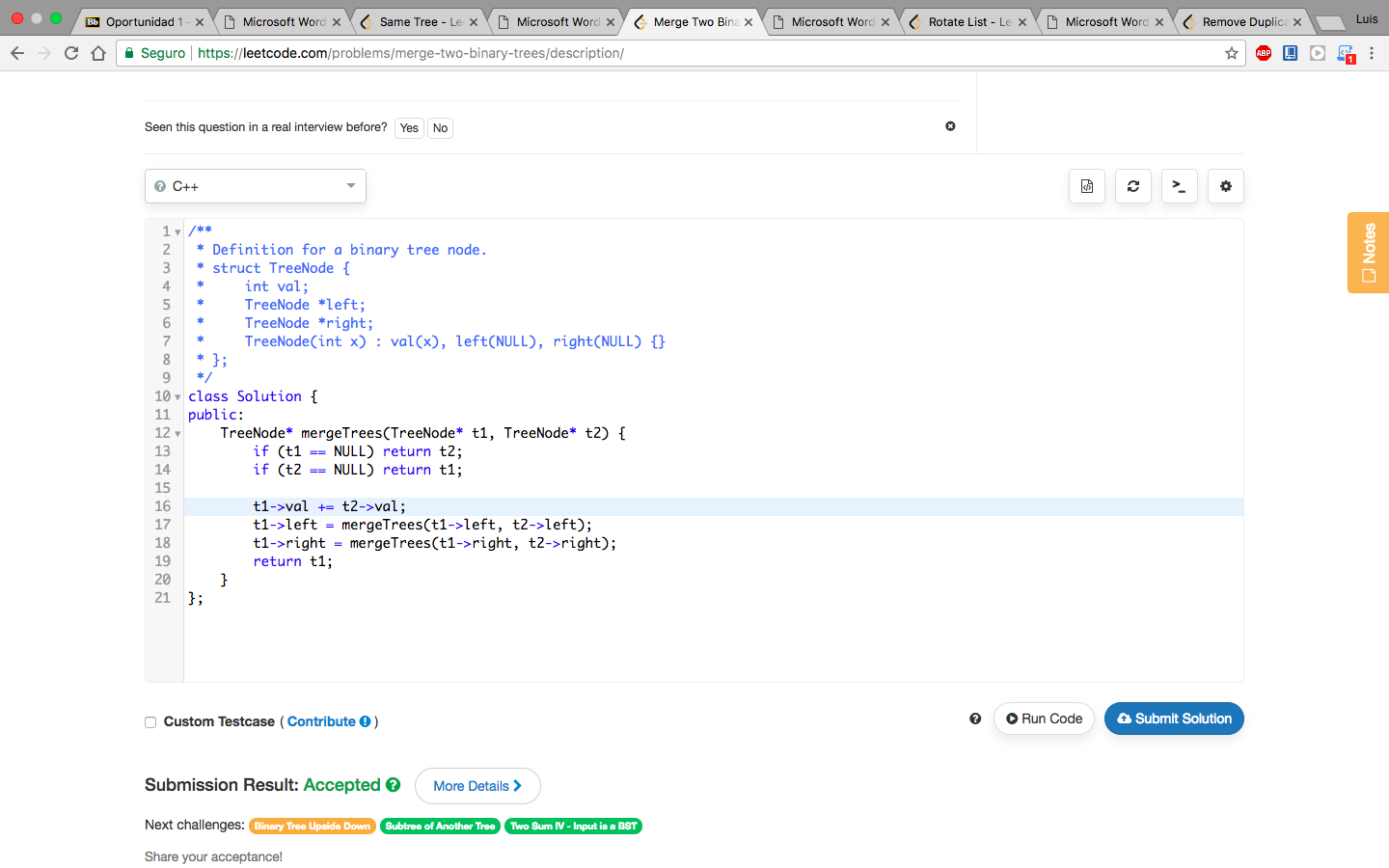
t1->val += t2->val;

t1->left = mergeTrees(t1->left, t2->left);

t1->right = mergeTrees(t1->right, t2->right);

return t1;

}

};

61. Rotate List

class Solution {

public:

ListNode\* rotateRight(ListNode\* head, int k) {

int size = 1;

ListNode \*pP = head;

if (pP == NULL) return head;

//count number of nodes

while(pP->next != NULL) {

size++;

pP = pP->next;

}

pP->next = head;

while(k >= size) {

k = k - size;

}

k = size - k;

pP = head;

for(int i=1; i < k; i++){

pP = pP->next;

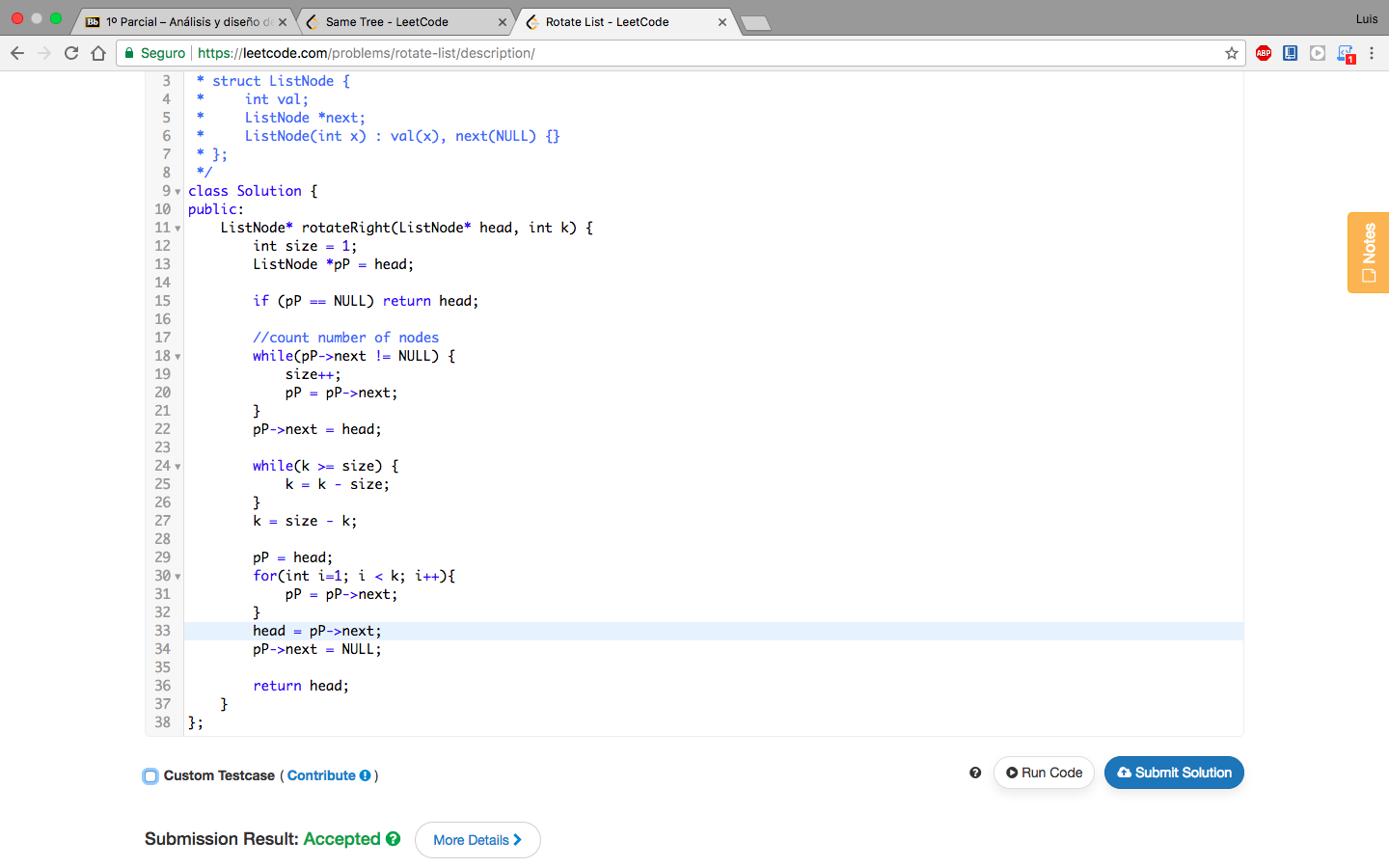
}

head = pP->next;

pP->next = NULL;

return head;

}

};

83. Remove Duplicates from Sorted List

class Solution {

public:

ListNode\* deleteDuplicates(ListNode\* head) {

ListNode \*pP = head;

ListNode \*pQ = head;

int x;

while (pQ != NULL && pQ->next != NULL) {

pP = pP->next;

x = pQ->val;

if (pP->val == x) {

pQ->next = pP->next;

delete pP;

pP = pQ;

} else

pQ = pQ->next;

}

return head;

}

};