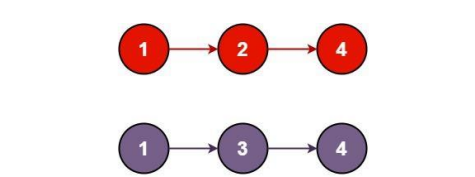
Practical - 9

AIM: Merge Two Sorted Lists You are given the heads of two sorted linked lists list1 and list2. Merge the two lists in a one sorted list. The list should be made by splicing together the nodes of the first two lists. Return the head of the merged linked list.

Example 1:





Input: list1 = [1,2,4], list2 = [1,3,4]

Output: [1,1,2,3,4,4]

* Program and Output

class Solution {

public:

ListNode\* mergeTwoLists(ListNode\* list1, ListNode\* list2) {

if(list1 == NULL && list2 == NULL){

return list1;

}

else if(list1 == NULL && list2!=NULL){

return list2;

}

else if(list1 != NULL && list2 == NULL){

return list1;

}

else if(list1->val >= list2->val){

ListNode\* first = list2;

ListNode\* trav = list2->next;

ListNode\* Next = list1->next;

if(trav == NULL && Next == NULL){

first->next = list1;

}

while(trav!=NULL || Next!=NULL){

if(trav==NULL){

list2->next = list1;

Next = Next->next;

break;

}

else if(trav->val >= list1->val){

list1->next = trav;

list2->next = list1;

list2 = list1;

list1 = Next;

if(Next == NULL){

break;

}

Next = Next->next;

}

else{

if(trav->next == NULL){

trav->next = list1;

break;

}

trav = trav->next;

list2 = list2->next;

}

}

return first;

}

else if(list1->val < list2->val){

ListNode\* first = list1;

ListNode\* trav = list1->next;

ListNode\* Next = list2->next;

if(trav == NULL && Next == NULL){

first->next = list2;

}

while(trav!=NULL || Next!=NULL){

if(trav==NULL){

list1->next = list2;

Next = Next->next;

break;

}

else if(trav->val >= list2->val){

list2->next = trav;

list1->next = list2;

list1 = list2;

list2 = Next;

if(Next == NULL){

break;

}

Next = Next->next;

}

else{

if(trav->next == NULL){

trav->next = list2;

break;

}

trav = trav->next;

list1 = list1->next;

}

}

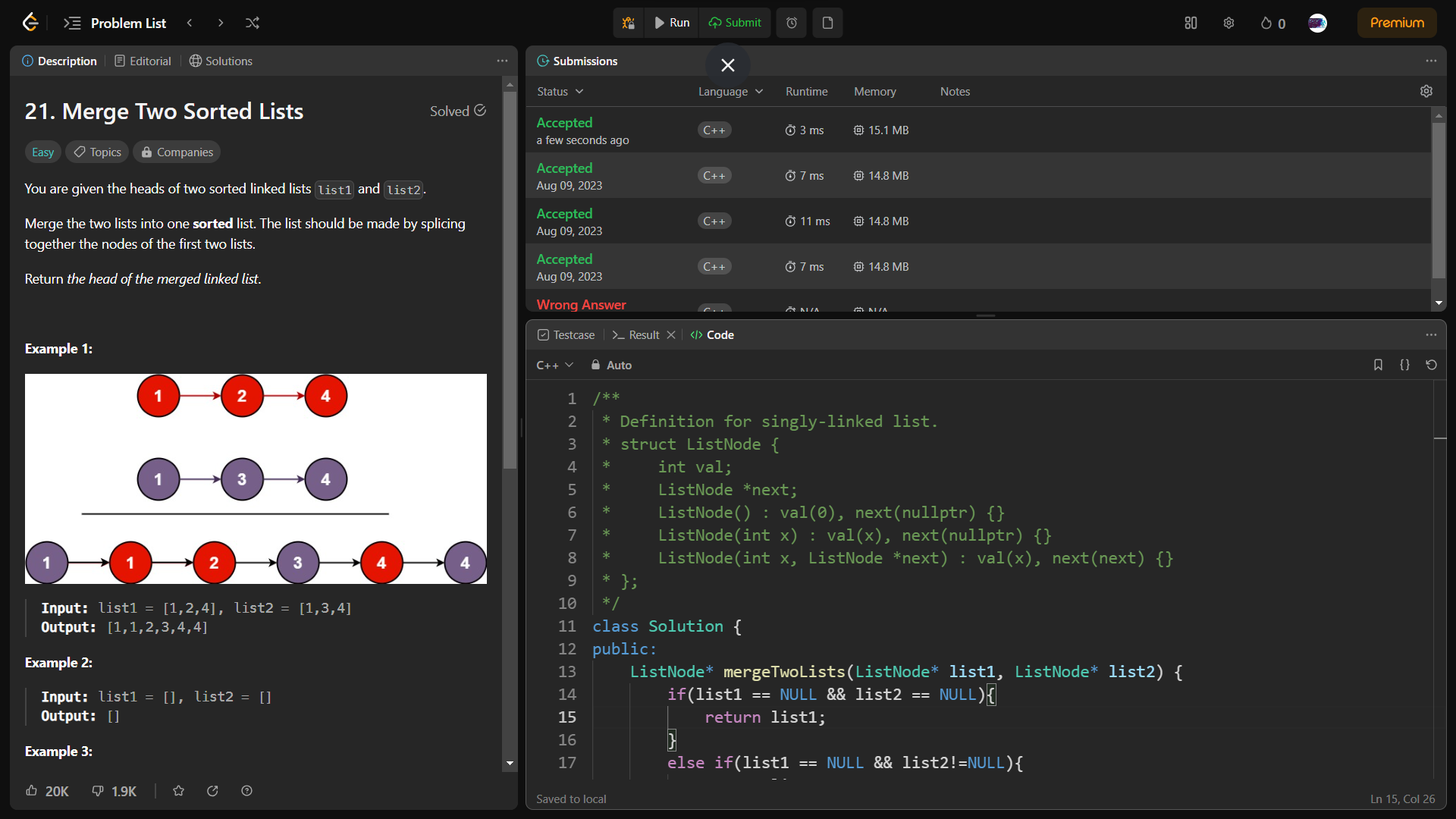
return first;

}

return list1;

}

};



* Conclusion

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_

Student Signature Faculty Signature Marks