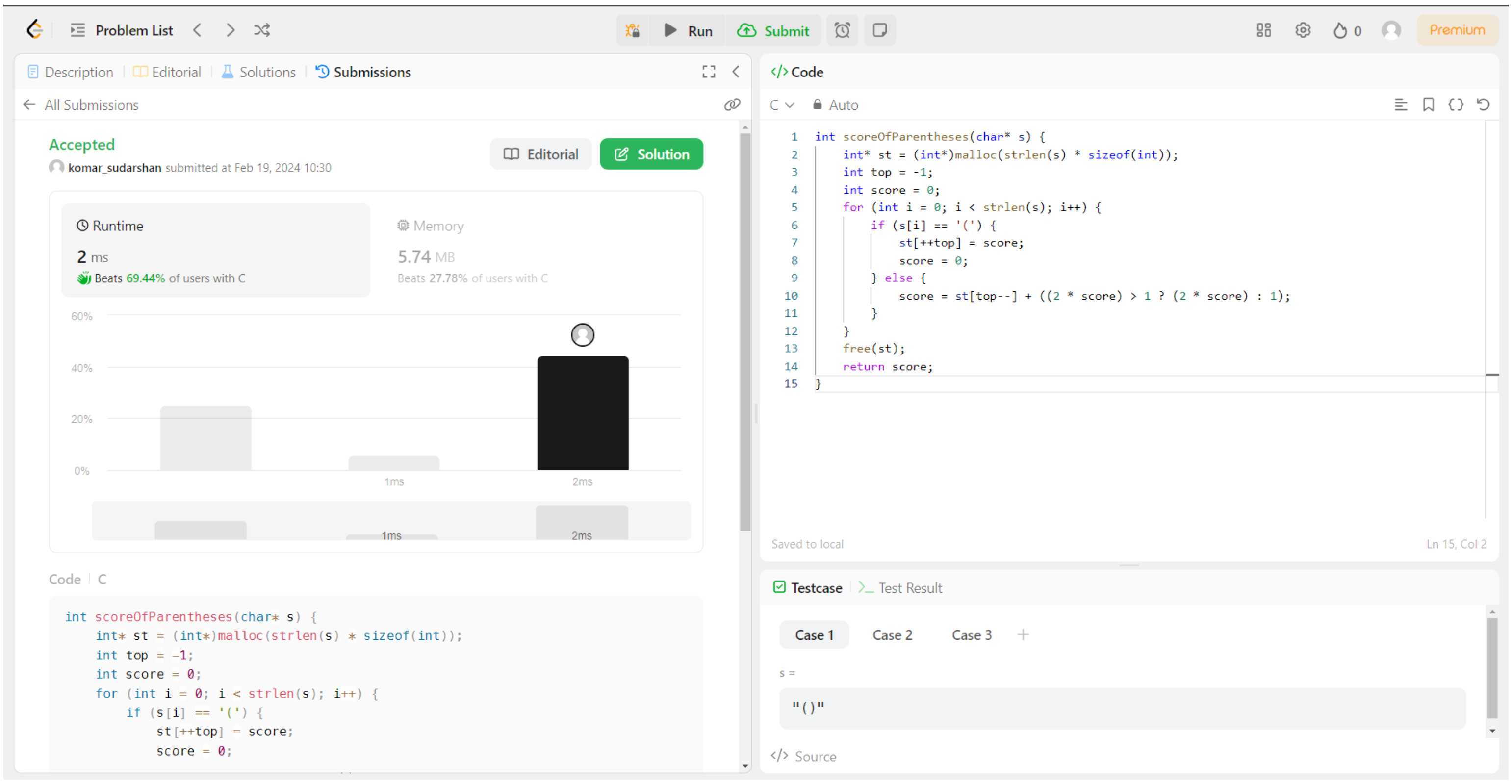
Leetcode problems

1.Score of parantheses



Accepted
komar_sudarshan submitted at Feb 19, 2024 10:30

Runtime
2 ms
Beats 69.44% of users with C

Memory
5.74 MB
Beats 27.78% of users with C

Code | C

```c
int scoreOfParentheses(char* s) {
    int* st = (int*)malloc(strlen(s) * sizeof(int));
    int top = -1;
    int score = 0;
    for (int i = 0; i < strlen(s); i++) {
        if (s[i] == '(') {
            st[++top] = score;
            score = 0;
```

C Auto

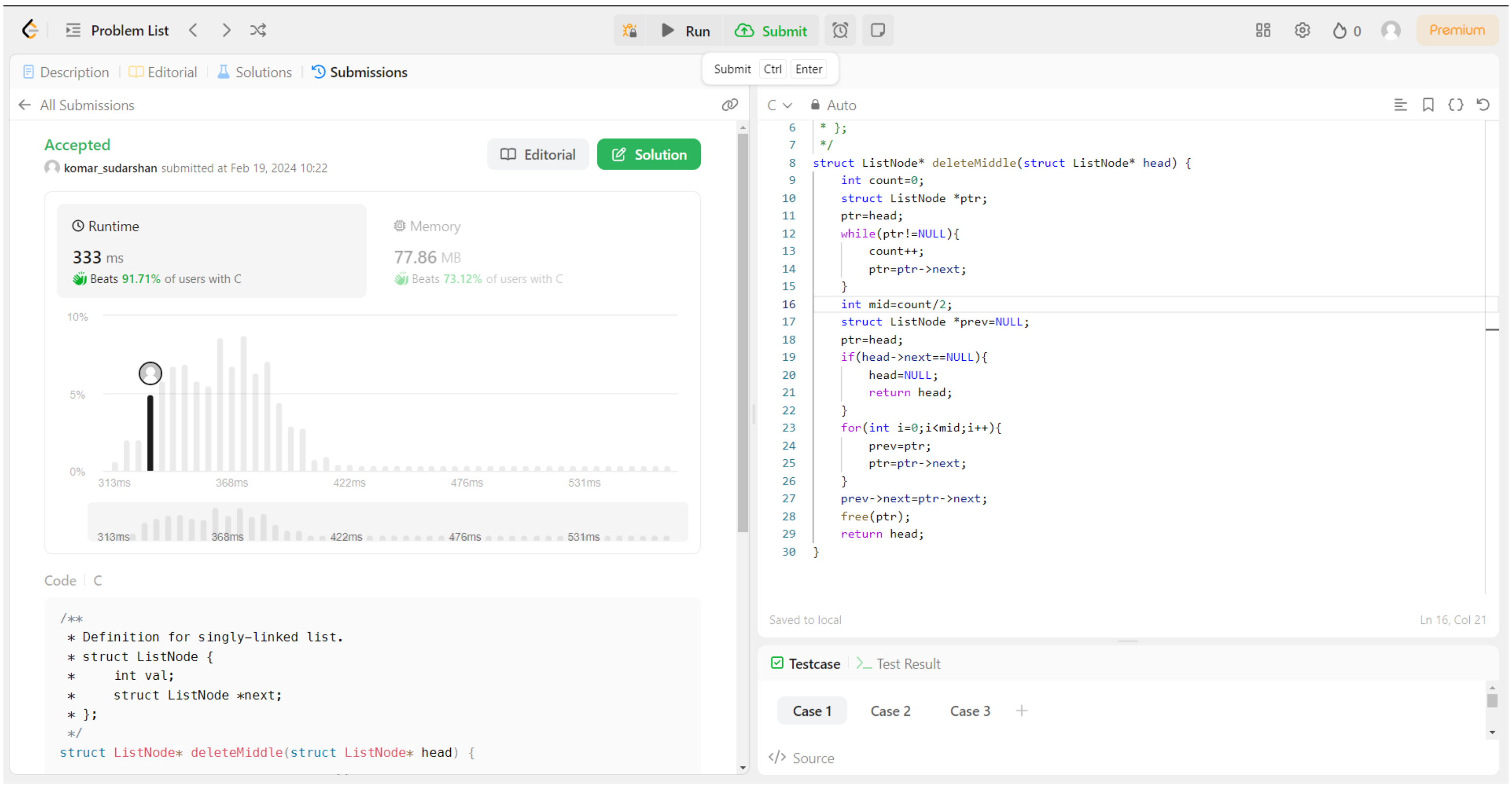
```c
int scoreOfParentheses(char* s) {
    int* st = (int*)malloc(strlen(s) * sizeof(int));
    int top = -1;
    int score = 0;
    for (int i = 0; i < strlen(s); i++) {
        if (s[i] == '(') {
            st[++top] = score;
            score = 0;
        } else {
            score = st[top--] + ((2 * score) > 1 ? (2 * score) : 1);
        }
    }
    free(st);
    return score;
}
```

Saved to local   Ln 15, Col 2

s =

"()"

2.Deleting middle node in a single linked list



Accepted
komar_sudarshan submitted at Feb 19, 2024 10:22

Runtime
333 ms
Beats 91.71% of users with C

Memory
77.86 MB
Beats 73.12% of users with C

Code | C

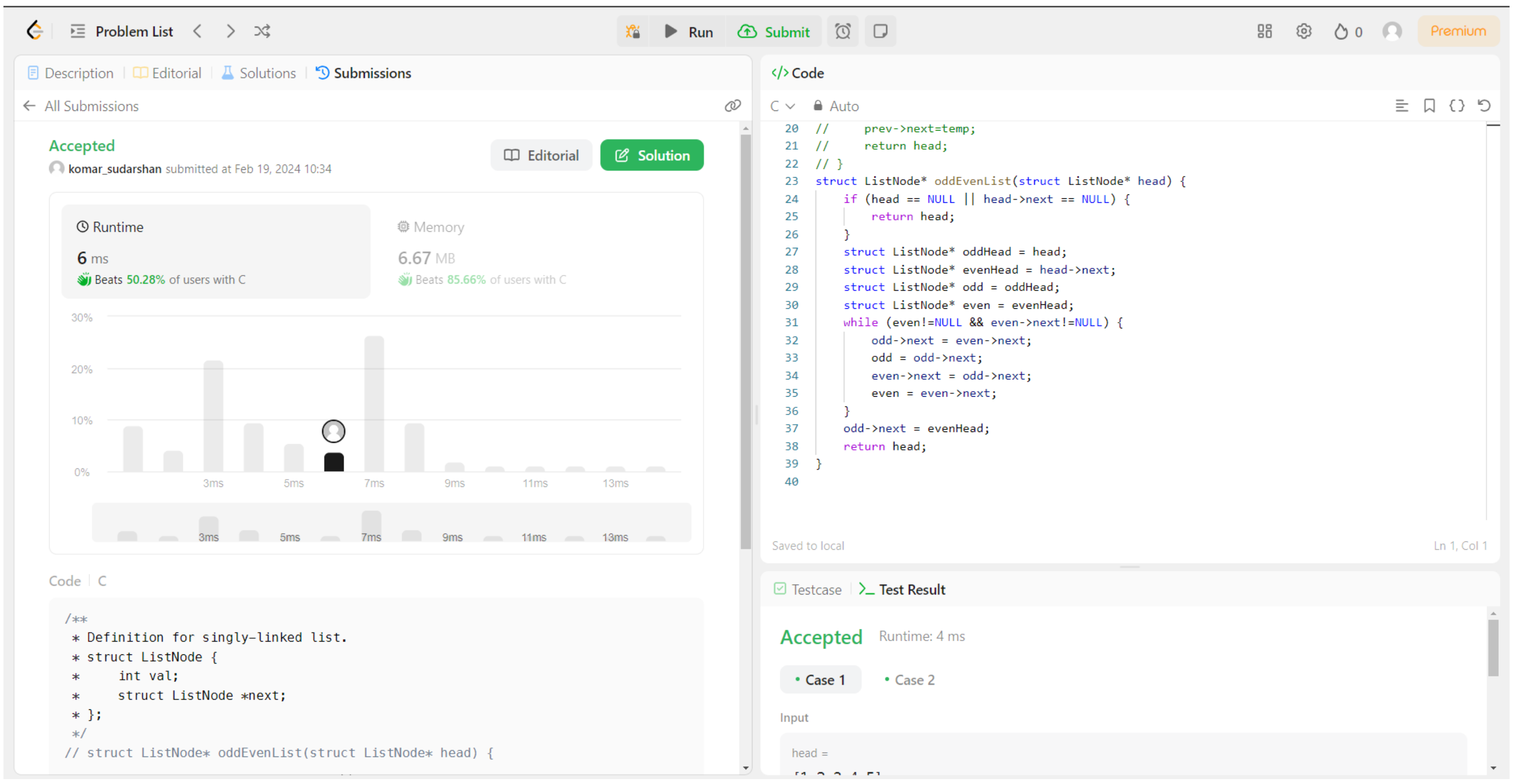
```c
/**
 * Definition for singly-linked list.
 * struct ListNode {
 *     int val;
 *     struct ListNode *next;
 * };
 */
struct ListNode* deleteMiddle(struct ListNode* head) {
```

C Auto

```c
 * };
 */
struct ListNode* deleteMiddle(struct ListNode* head) {
    int count=0;
    struct ListNode *ptr;
    ptr=head;
    while(ptr!=NULL){
        count++;
        ptr=ptr->next;
    }
    int mid=count/2;
    struct ListNode *prev=NULL;
    ptr=head;
    if(head->next==NULL){
        head=NULL;
        return head;
    }
    for(int i=0;i<mid;i++){
        prev=ptr;
        ptr=ptr->next;
    }
    prev->next=ptr->next;
    free(ptr);
    return head;
}
```

Saved to local   Ln 16, Col 21

3.Odd even grouping in single linked list



Problem List   Run   Submit   Premium

Description | Editorial | Solutions | Submissions

All Submissions

Accepted
komar_sudarshan submitted at Feb 19, 2024 10:34

Editorial   Solution

Runtime
6 ms
Beats 50.28% of users with C

Memory
6.67 MB
Beats 85.66% of users with C

Code | C

```
/**
 * Definition for singly-linked list.
 * struct ListNode {
 *     int val;
 *     struct ListNode *next;
 * };
 */
// struct ListNode* oddEvenList(struct ListNode* head) {
```

Code

C   Auto

```
//     prev->next=temp;
//     return head;
// }
struct ListNode* oddEvenList(struct ListNode* head) {
    if (head == NULL || head->next == NULL) {
        return head;
    }
    struct ListNode* oddHead = head;
    struct ListNode* evenHead = head->next;
    struct ListNode* odd = oddHead;
    struct ListNode* even = evenHead;
    while (even!=NULL && even->next!=NULL) {
        odd->next = even->next;
        odd = odd->next;
        even->next = odd->next;
        even = even->next;
    }
    odd->next = evenHead;
    return head;
}
```

Saved to local   Ln 1, Col 1

Testcase | Test Result

Accepted   Runtime: 4 ms

Case 1   Case 2

Input

head =