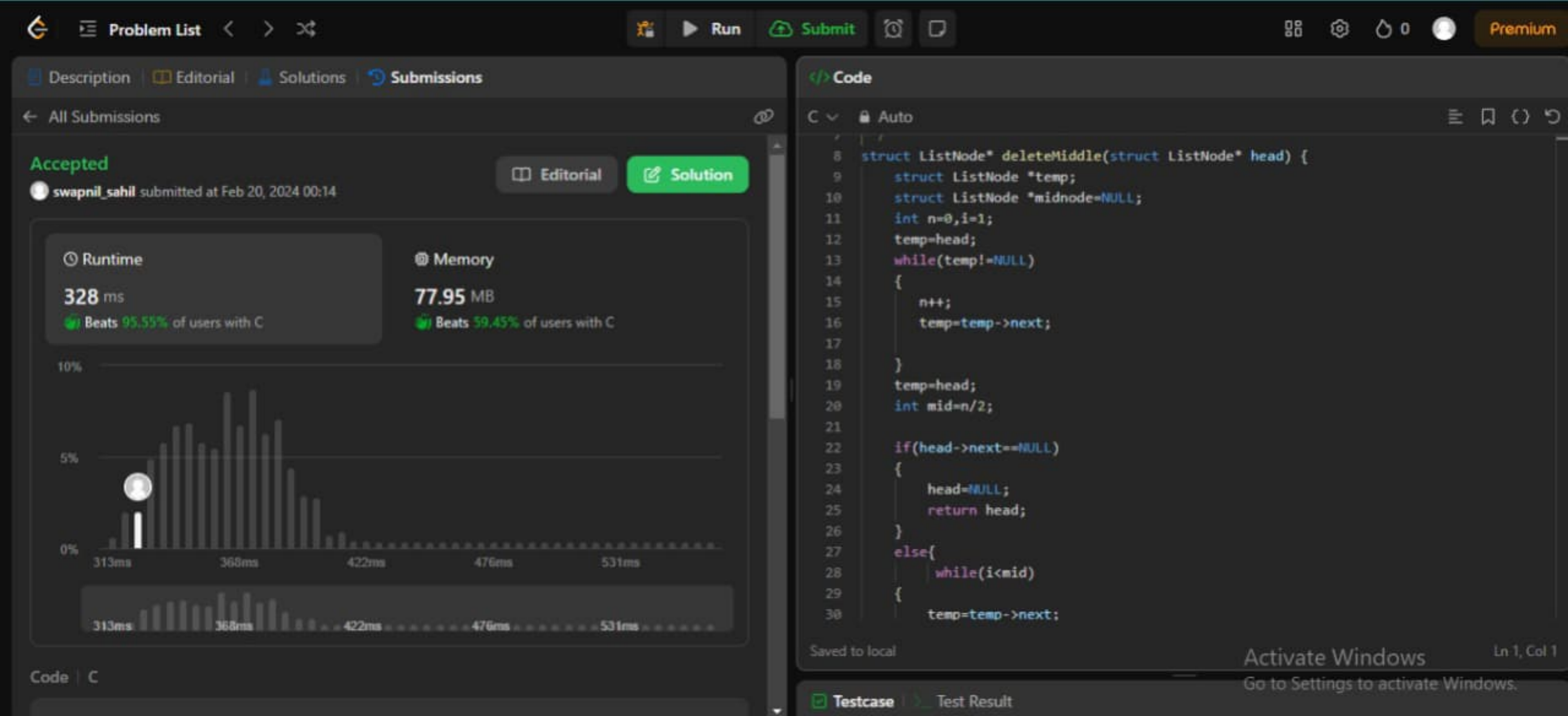
Date :
Page No :

2. leetcode : Delete middlenode of a linkedlist

```
struct ListNode *deleteMiddle(struct ListNode *head)
{
    struct ListNode *temp;
    struct ListNode *midnode = NULL;
    int n=0, i=1;
    temp=head;
    while (temp!=NULL)
    {
        n++;
        temp=temp->next;
    }
    temp=head;
    int mid=n/2;
    if (head->next==NULL)
    {   head=NULL;
        return head;
    }
    else {
            while (i<mid)
        {
            temp=temp->next;
              i++;
        }
    }
    midnode=temp->next;
    temp->next=midnode->next;
    free (midnode);

    return head;
}
```



Accepted

swapnil_sahil submitted at Feb 20, 2024 00:14

Runtime
328 ms
Beats 95.55% of users with C

Memory
77.95 MB
Beats 59.45% of users with C

Code | C

```
struct ListNode* deleteMiddle(struct ListNode* head) {
    struct ListNode *temp;
    struct ListNode *midnode=NULL;
    int n=0,i=1;
    temp=head;
    while(temp!=NULL)
    {
        n++;
        temp=temp->next;

    }
    temp=head;
    int mid=n/2;

    if(head->next==NULL)
    {
        head=NULL;
        return head;
    }
    else{
        while(i<mid)
    {
        temp=temp->next;
```

Saved to local

Ln 1, Col 1

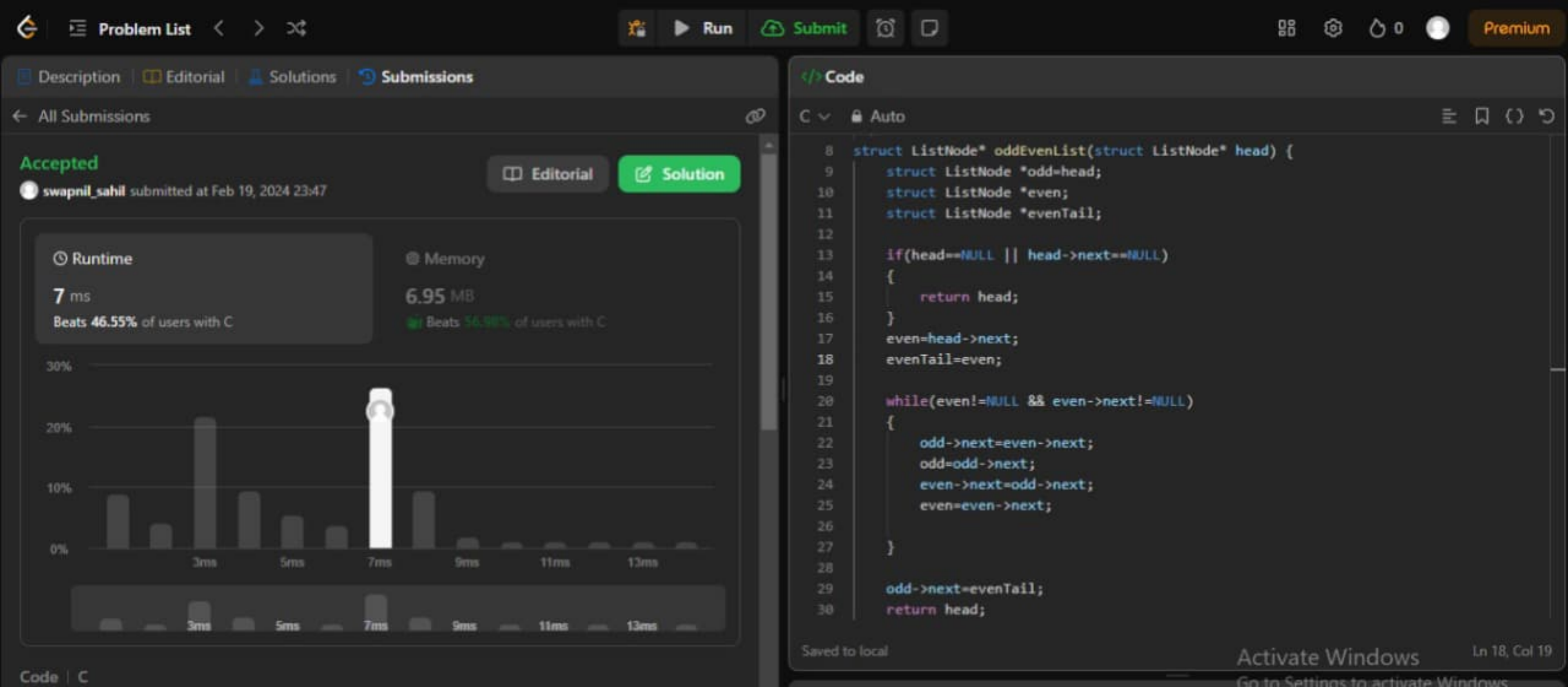
Date :
Page No :

3. leetcode : Odd even linked list.

```
struct ListNode * oddEvenList(struct ListNode *head){
     struct ListNode *odd=head;
     struct ListNode *even, *evenTail;

     if (head==NULL || head->next==NULL)
      {  return head;
      }
     even =head->next;
     evenTail=even;

     while(even!=NULL && even->next!=NULL)
      {
         odd->next=even->next;
         odd = odd->next;
         even->next= odd->next;
         even=even->next;
      }
      odd->next=evenTail;
      return head;
}
```



Accepted

swapnil_sahil submitted at Feb 19, 2024 23:47

Runtime

7 ms

Beats 46.55% of users with C

Memory

6.95 MB

Beats 56.98% of users with C

Code | C

```c
struct ListNode* oddEvenList(struct ListNode* head) {
    struct ListNode *odd=head;
    struct ListNode *even;
    struct ListNode *evenTail;

    if(head==NULL || head->next==NULL)
    {
        return head;
    }
    even=head->next;
    evenTail=even;

    while(even!=NULL && even->next!=NULL)
    {
        odd->next=even->next;
        odd=odd->next;
        even->next=odd->next;
        even=even->next;

    }

    odd->next=evenTail;
    return head;
```

Saved to local

Ln 18, Col 19