RAJKUMAR V

CSE -C

22CS132

DATASTRUCTURE

ASSIGNMENT

1)Problem

Linked List Cycle II

Code

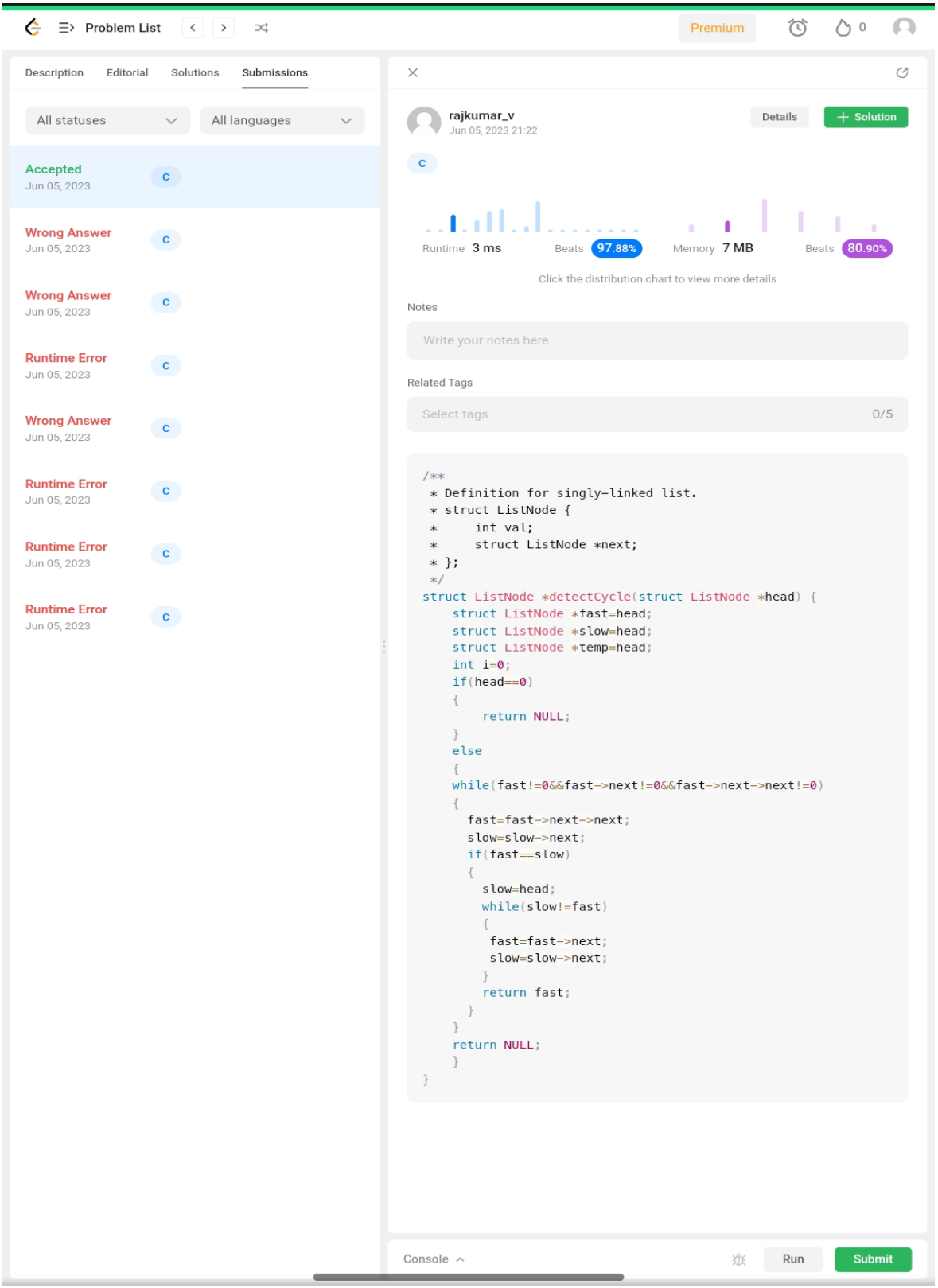
```
/**
 * Definition for singly-linked list.
 * struct ListNode {
 *     int val;
 *     struct ListNode *next;
 * };
 */
struct ListNode *detectCycle(struct ListNode *head) {
    struct ListNode *fast=head;
    struct ListNode *slow=head;
    struct ListNode *temp=head;
```

```
 int i=0;
 if(head==0)
 {
    return NULL;
 }
 else
 {
 while(fast!=0&&fast->next!=0&&fast->next->next!=0)
{



fast=fast->next->next;
   slow=slow->next;
   if(fast==slow)
   {
    slow=head;
    while(slow!=fast)
    {
     fast=fast->next;
     slow=slow->next;
    }
    return fast;
  }
 }
 return NULL;
```

```
    }
}
```



2) Problem

Search a 2D Matrix

Code

```
bool searchMatrix(int** matrix, int matrixSize, int* matrixColSize, int target){
    for(int i=0;i<matrixSize;i++){
        for(int j=0;j<*matrixColSize;j++){
            if(target==matrix[i][j])
                return 1;
        }
    }
    return 0;
}
```

Search a 2D Matrix

Submission Detail

133 / 133 test cases passed.

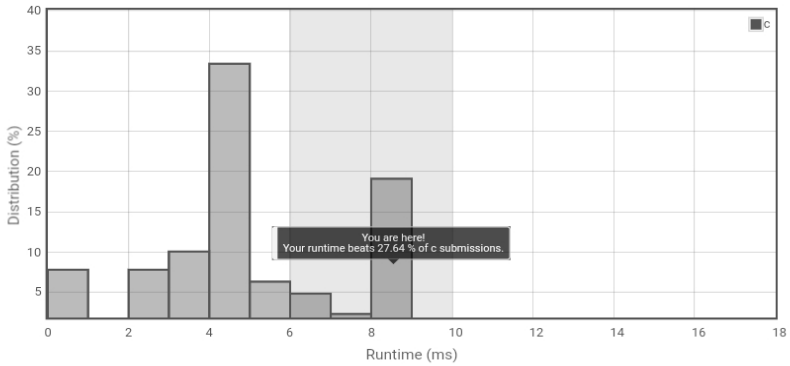
Status: **Accepted**

Runtime: **8 ms**

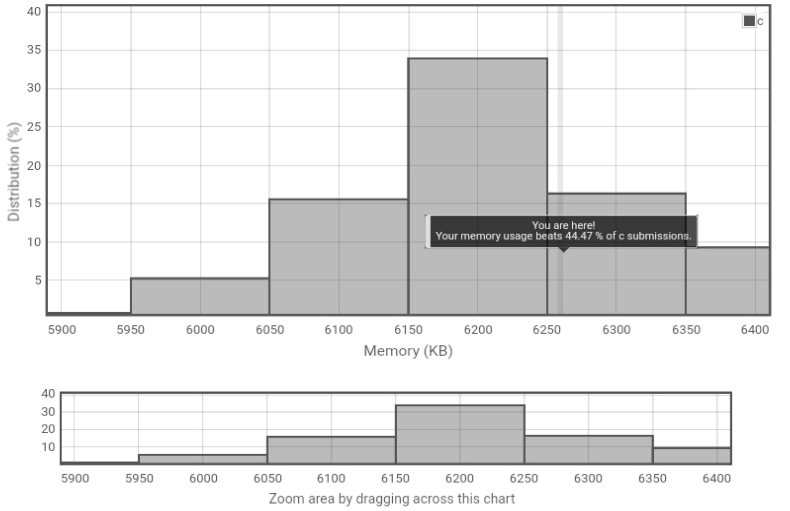
Memory Usage: **6.3 MB**

Submitted: **0 minutes ago**

Accepted Solutions Runtime Distribution



Accepted Solutions Memory Distribution



Invite friends to challenge **Search a 2D Matrix**

Submitted Code: 0 minutes ago

Language: c

Edit Code

```c
bool searchMatrix(int** matrix, int matrixSize, int* matrixColSize, int target
    ){
    for(int i=0;i<matrixSize;i++){
        for(int j=0;j<*matrixColSize;j++){
            if(target==matrix[i][j])
                return 1;
        }
    }
    return 0;
}
```

3) Problem

Merge Two Sorted Lists

Code

```
/**
 * Definition for singly-linked list.
 * struct ListNode {
 *     int val;
 *     struct ListNode *next;
 * };
 */
struct ListNode* mergeTwoLists(struct ListNode* list1, struct ListNode* list2){
    struct ListNode* temp=list1;
    int k;
    if(temp==0)
    {
       return list2;
    }
    else
    {
       while(temp->next!=0)
       {
          temp=temp->next;
       }
    }
```

```
        temp->next=list2;
        temp=list1;
    for(struct ListNode* i=list1;i!=0;i=i->next)
    {
        for(struct ListNode* j=i->next;j!=0;j=j->next)
        {
            if(i->val>=j->val)
            {
                k=i->val;
                i->val=j->val;
                j->val=k;
            }
        }
    }
    return list1;
}
```

Description Editorial Solutions Submissions

All statuses All languages

Jun 06, 2023 14:46

Jun 06, 2023

C

rajkumar_v
Jun 06, 2023 14:46

Details

C

Runtime 2 ms Beats 84.82% Memory 6 MB Beats

Click the distribution chart to view more details

Notes

Write your notes here

Related Tags

Select tags

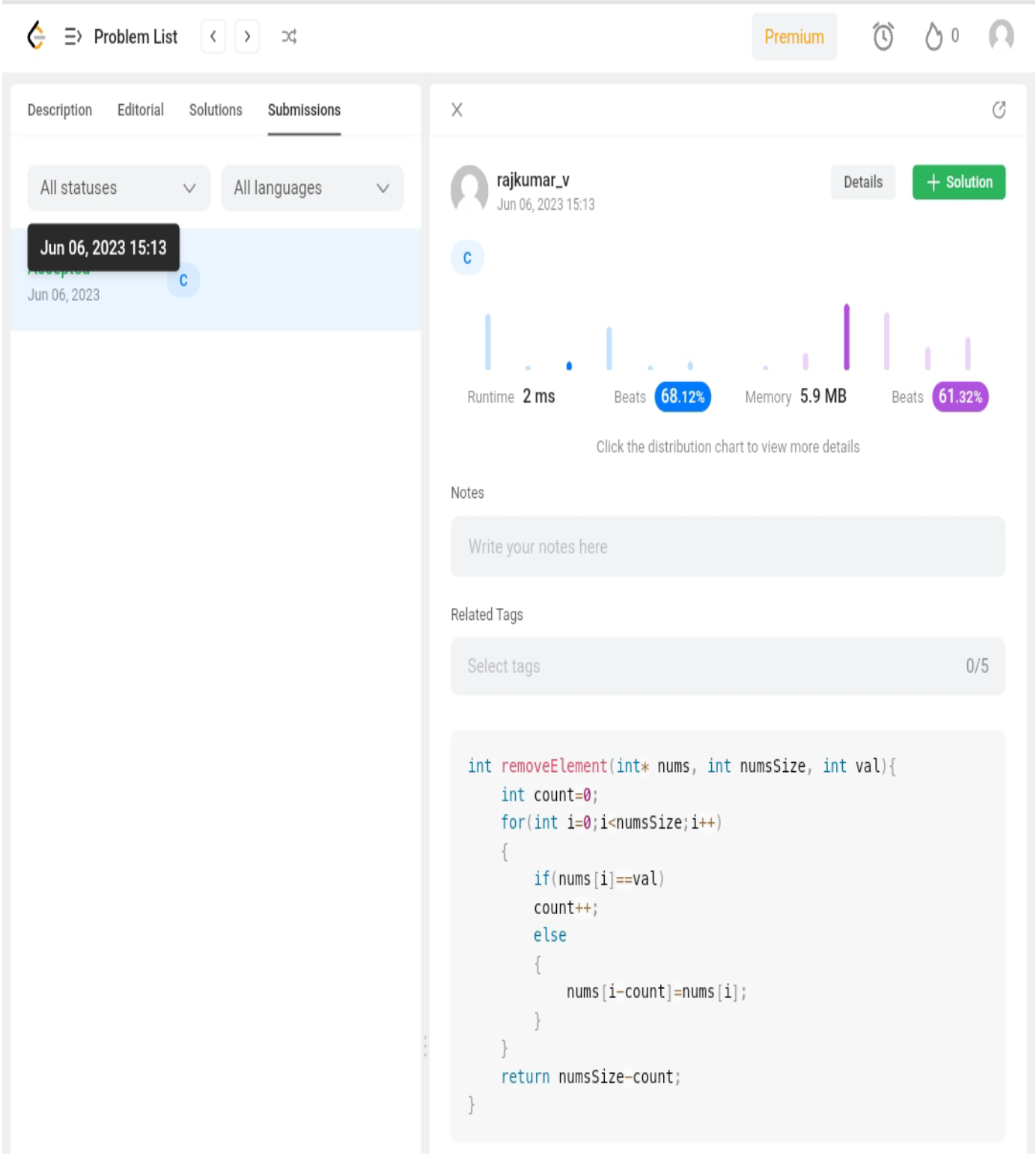
```
/**
 * Definition for singly-linked list.
 * struct ListNode {
 *     int val;
 *     struct ListNode *next;
 * };
 */
struct ListNode* mergeTwoLists(struct ListNode* list1, s
      struct ListNode* temp=list1;
      int k;
      if(temp==0)
      {
          return list2;
      }
      else
      {
          while(temp->next!=0)
          {
              temp=temp->next;
          }
      }
      temp->next=list2;
      temp=list1;
     for(struct ListNode* i=list1;i!=0;i=i->next)
     {
         for(struct ListNode* j=i->next;j!=0;j=j->next)
         {
             if(i->val>=j->val)
             {
                 k=i->val;
                 i->val=j->val;
                 j->val=k;
             }
         }
```

4) Problem

Remove Element

Code

```
int removeElement(int* nums, int numsSize, int val){
    int count=0;
    for(int i=0;i<numsSize;i++)
    {
        if(nums[i]==val)
        count++;
        else
        {
            nums[i-count]=nums[i];
        }
    }
    return numsSize-count;
}
```



5) Problem

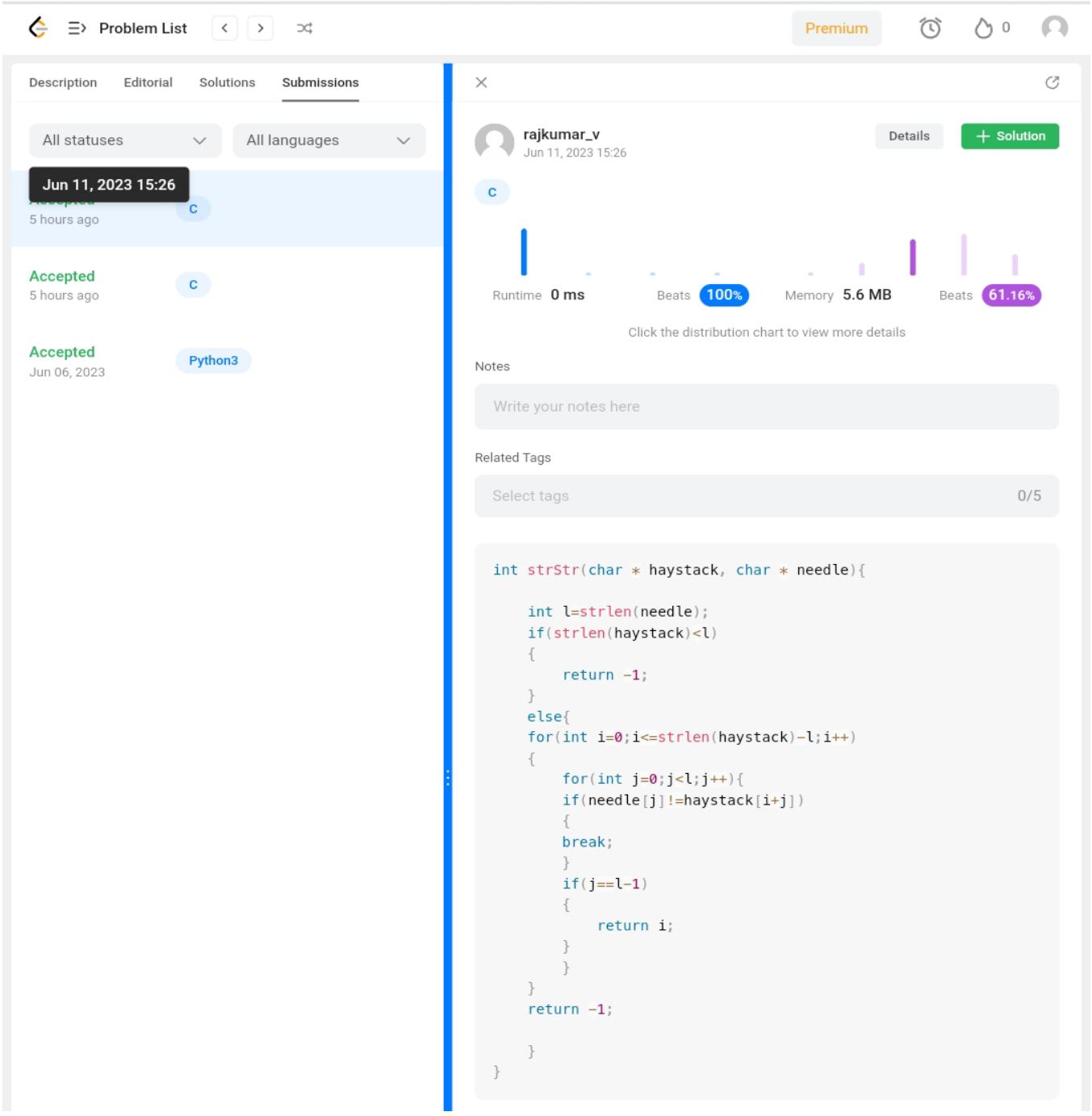
Find the Index of the First Occurrence in a String

Code

```
int strStr(char * haystack, char * needle){


    int l=strlen(needle);
    if(strlen(haystack)<l)
    {
        return -1;
    }
    else{
    for(int i=0;i<=strlen(haystack)-l;i++)
    {
        for(int j=0;j<l;j++){
        if(needle[j]!=haystack[i+j])
        {
        break;
        }
        if(j==l-1)
        {
            return i;
        }
        }
    }
    return -1;
```

```
    }
}
```



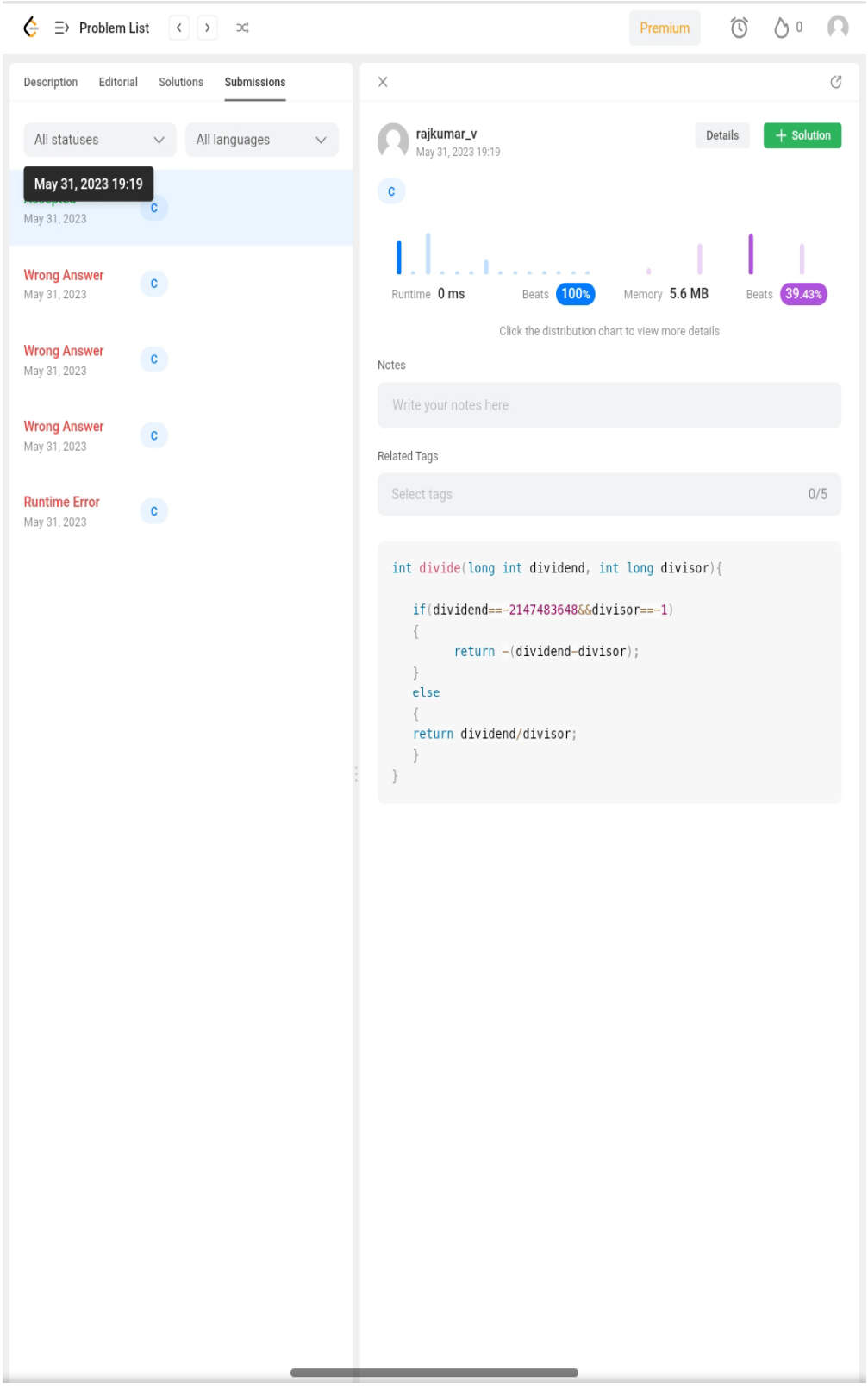
6)

Divide Two Integers

Code

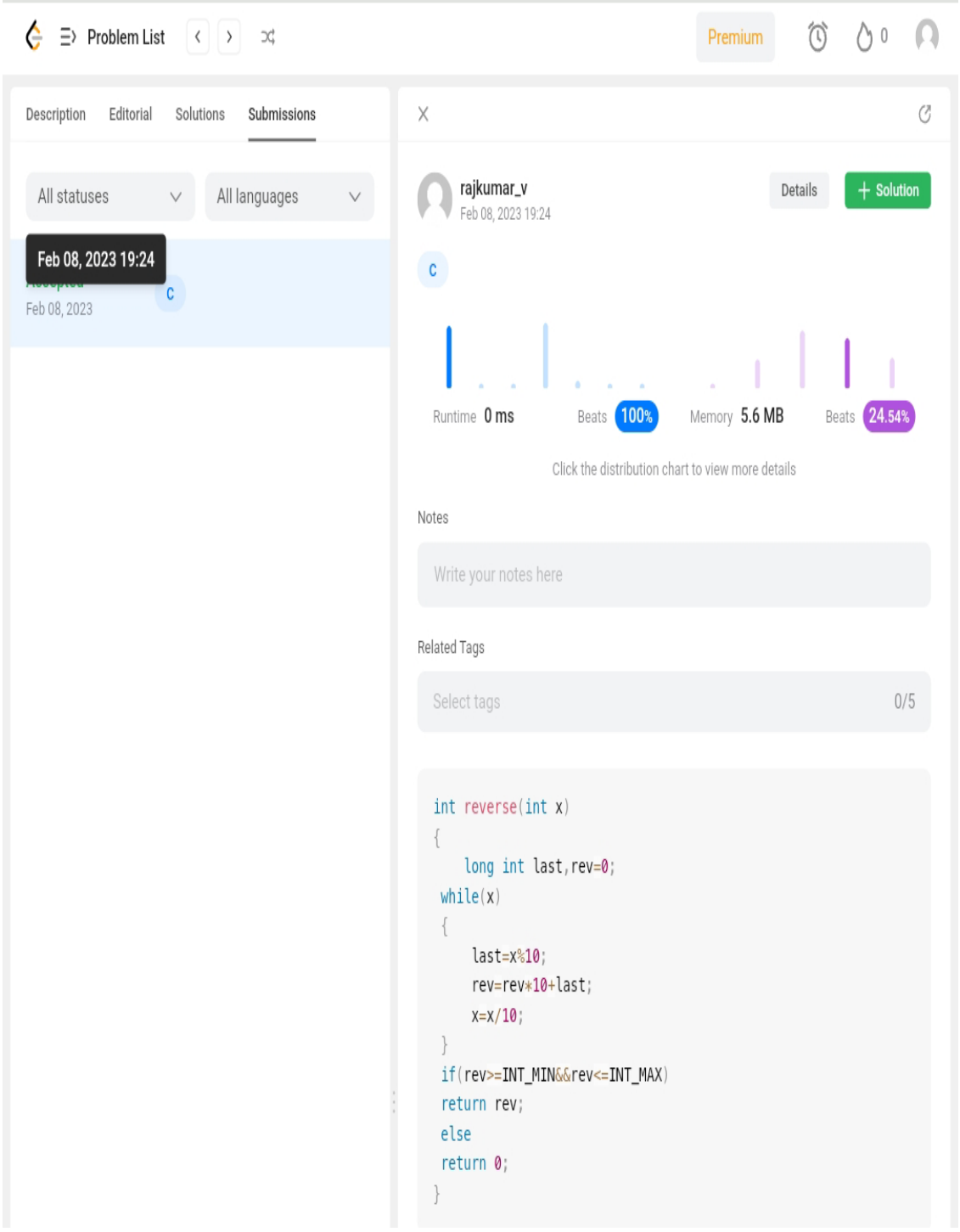
```
int divide(long int dividend, int long divisor){


  if(dividend==-2147483648&&divisor==-1)
  {
      return -(dividend-divisor);
  }
  else
  {
  return dividend/divisor;
  }
}
```



7)Problem

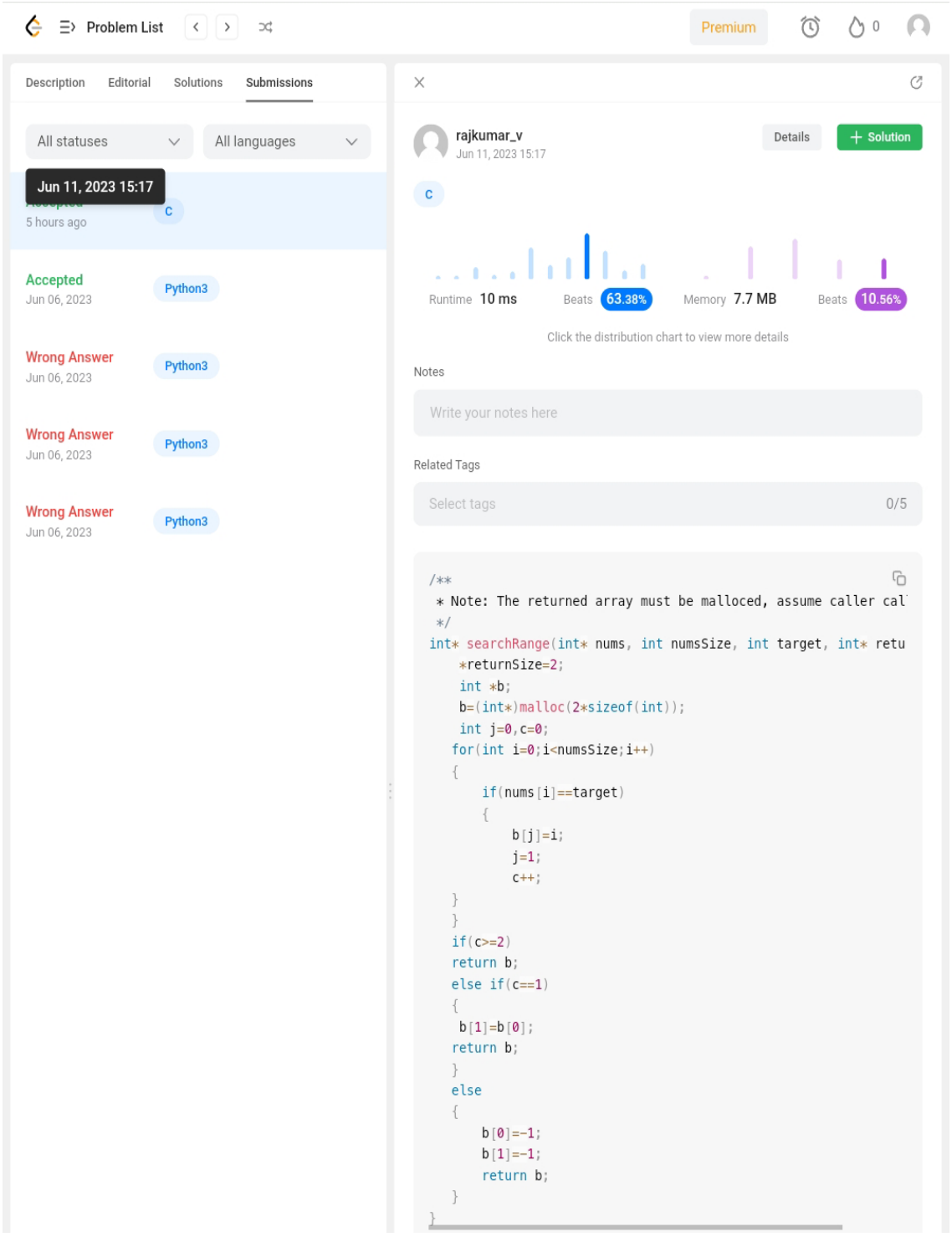
```
int reverse(int x)
{
   long int last,rev=0;
 while(x)
 {
    last=x%10;
    rev=rev*10+last;
    x=x/10;
 }
 if(rev>=INT_MIN&&rev<=INT_MAX)
 return rev;
 else
 return 0;
}
```



8) Problem

Find First and Last Position of Element in Sorted Array

```
int reverse(int x)
{
   long int last,rev=0;
 while(x)
 {
    last=x%10;
    rev=rev*10+last;
    x=x/10;
 }
 if(rev>=INT_MIN&&rev<=INT_MAX)
 return rev;
 else
 return 0;
}
```



9) Problem

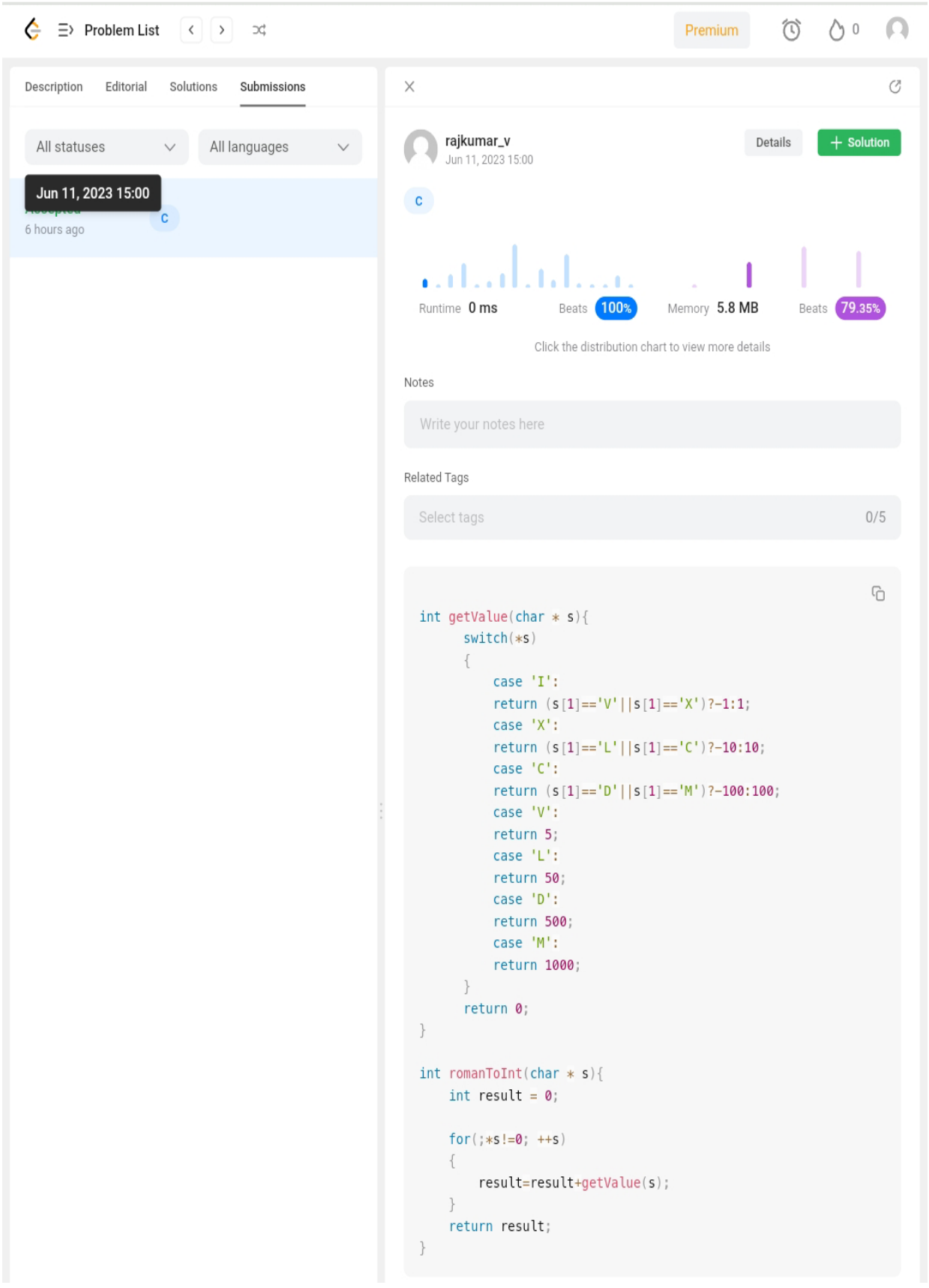
Roman to Integer

Code

```
int getValue(char * s){
    switch(*s)
    {
       case 'I':
       return (s[1]=='V'||s[1]=='X')?-1:1;
       case 'X':
       return (s[1]=='L'||s[1]=='C')?-10:10;
       case 'C':
       return (s[1]=='D'||s[1]=='M')?-100:100;
       case 'V':
       return 5;
       case 'L':
       return 50;
       case 'D':
       return 500;
       case 'M':
       return 1000;
    }
    return 0;
}
```

```
int romanToInt(char * s){
    int result = 0;


    for(;*s!=0; ++s)
    {
        result=result+getValue(s);
    }
    return result;
}
```



10)Problem

Remove Nth Node From End of List

Code

```
/**
 * Definition for singly-linked list.
 * struct ListNode {
 *     int val;
 *     struct ListNode *next;
 * };
 */
struct ListNode* removeNthFromEnd(struct ListNode* head, int n){
  struct ListNode* temp=head;
  struct ListNode* prev=NULL;
  int count=0;
  if(head->next==0)
  {
    return NULL;
  }
  else
  {
     while(temp!=0)
     {
       temp=temp->next;
       count++;
     }
```

```
}
if(n==count)
{
  head=head->next;
}
else
{
  temp=head;
for(int i=0;i<count-n-1;i++)
{
  temp=temp->next;
}
temp->next=temp->next->next;
}
return head;
  }
```

Description Editorial Solutions Submissions

All statuses All languages

Jun 07, 2023 12:09

Jun 07, 2023 C

Runtime Error
Jun 07, 2023 C

Wrong Answer
Jun 07, 2023 C

Wrong Answer
Jun 06, 2023 C

Wrong Answer
Jun 06, 2023 C

rajkumar_v
Jun 07, 2023 12:09

Details + Solution

C

Runtime 2 ms Beats 64.32% Memory 5.6 MB Beats 99.62%

Click the distribution chart to view more details

Notes

Write your notes here

Related Tags

Select tags 0/5

```c
/**
 * Definition for singly-linked list.
 * struct ListNode {
 *     int val;
 *     struct ListNode *next;
 * };
 */
struct ListNode* removeNthFromEnd(struct ListNode* head, int n){
  struct ListNode* temp=head;
  struct ListNode* prev=NULL;
  int count=0;
  if(head->next==0)
  {
    return NULL;
  }
  else
  {
     while(temp!=0)
     {
       temp=temp->next;
       count++;
     }
  }
  if(n==count)
  {
    head=head->next;
  }
  else
  {
    temp=head;
  for(int i=0;i<count-n-1;i++)
  {
    temp=temp->next;
  }
  temp->next=temp->next->next;
  }
  return head;
    }
```