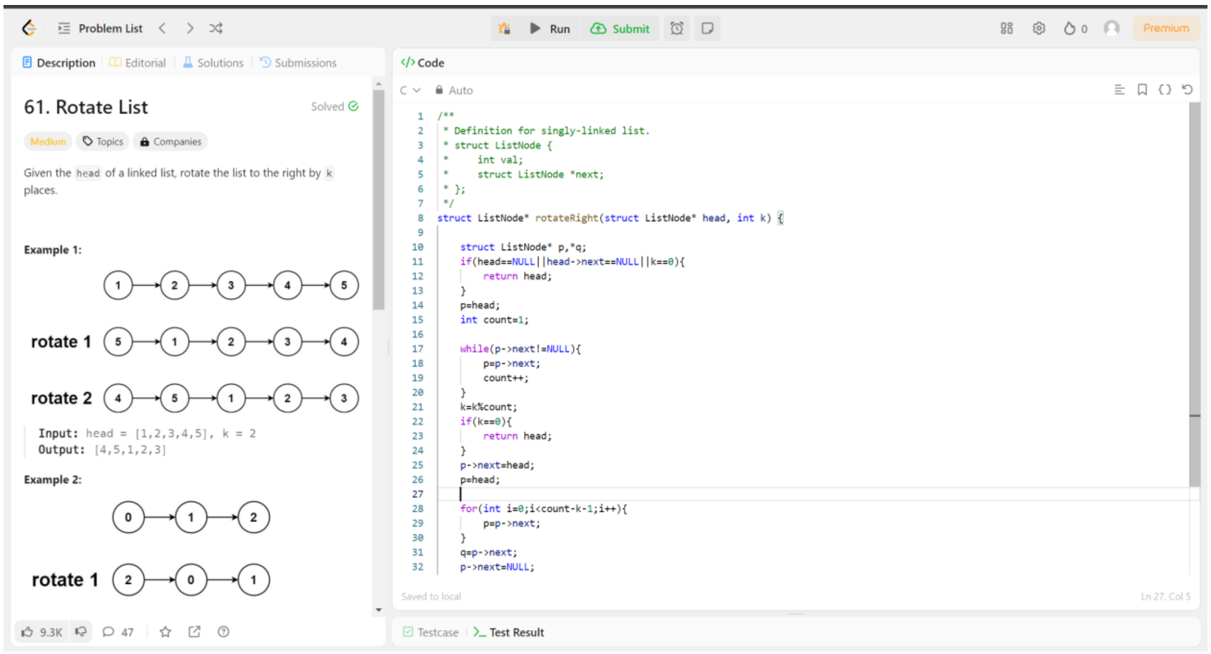
Rotate list leetcode



61. Rotate List

Solved

Medium · Topics · Companies

Given the `head` of a linked list, rotate the list to the right by `k` places.

Example 1:

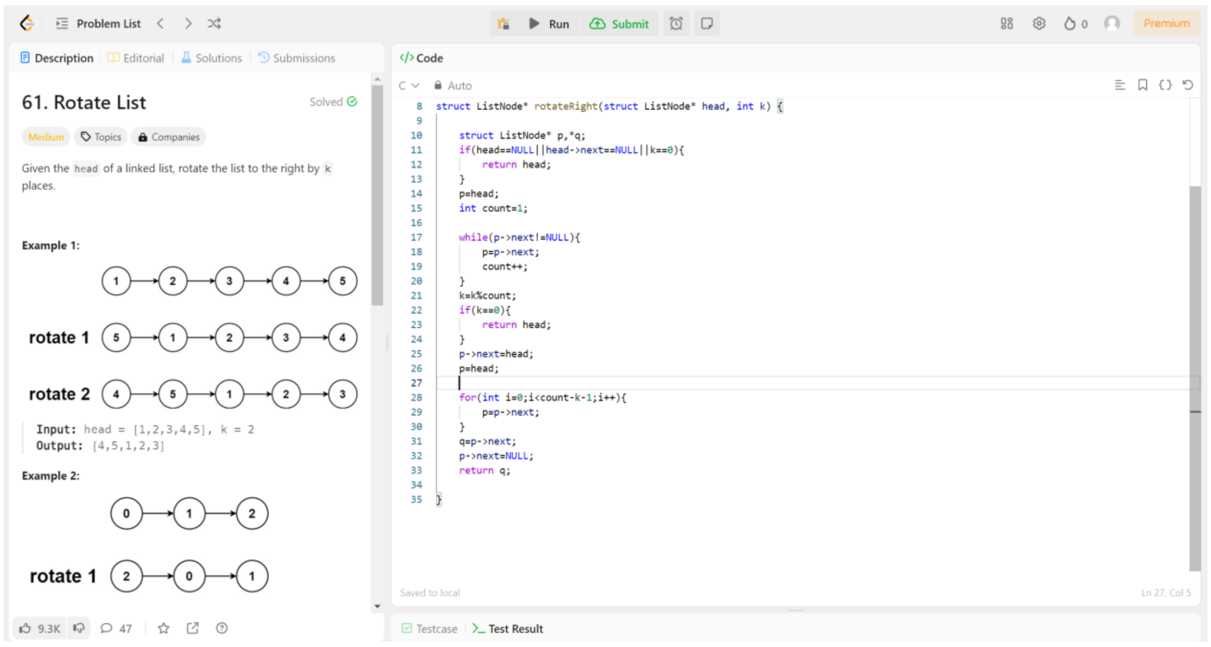
```
Input: head = [1,2,3,4,5], k = 2
Output: [4,5,1,2,3]
```

Example 2:

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

    struct ListNode* p,*q;
    if(head==NULL||head->next==NULL||k==0){
        return head;
    }
    p=head;
    int count=1;

    while(p->next!=NULL){
        p=p->next;
        count++;
    }
    k=k%count;
    if(k==0){
        return head;
    }
    p->next=head;
    p=head;

    for(int i=0;i<count-k-1;i++){
        p=p->next;
    }
    q=p->next;
    p->next=NULL;
```

Saved to local · Ln 27, Col 5

Testcase · Test Result



61. Rotate List

Solved

Medium · Topics · Companies

Given the `head` of a linked list, rotate the list to the right by `k` places.

Example 1:

```
Input: head = [1,2,3,4,5], k = 2
Output: [4,5,1,2,3]
```

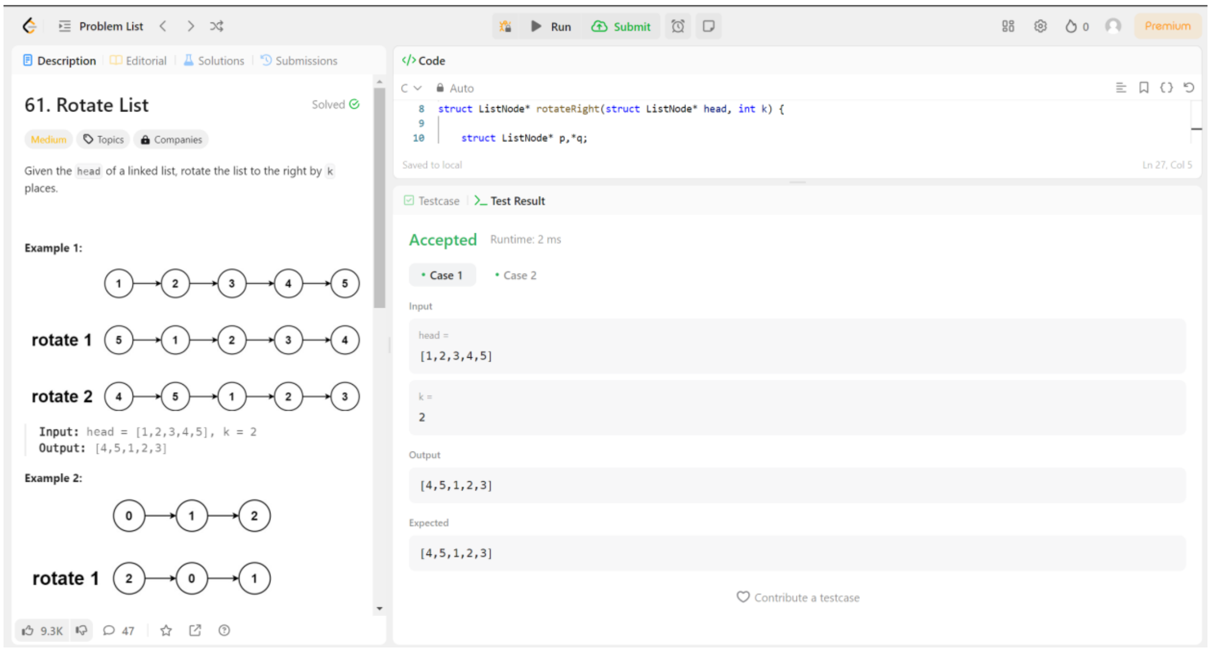
Example 2:

```c
struct ListNode* rotateRight(struct ListNode* head, int k) {

    struct ListNode* p,*q;
    if(head==NULL||head->next==NULL||k==0){
        return head;
    }
    p=head;
    int count=1;

    while(p->next!=NULL){
        p=p->next;
        count++;
    }
    k=k%count;
    if(k==0){
        return head;
    }
    p->next=head;
    p=head;

    for(int i=0;i<count-k-1;i++){
        p=p->next;
    }
    q=p->next;
    p->next=NULL;
    return q;

}
```

Saved to local · Ln 27, Col 5

Testcase · Test Result

Output



61. Rotate List

Solved

Medium · Topics · Companies

Given the `head` of a linked list, rotate the list to the right by `k` places.

Example 1:

```
Input: head = [1,2,3,4,5], k = 2
Output: [4,5,1,2,3]
```

Example 2:

9.3K · 47

C · Auto

```
struct ListNode* rotateRight(struct ListNode* head, int k) {

    struct ListNode* p,*q;
```

Saved to local · Ln 27, Col 5

Testcase · Test Result

Accepted Runtime: 2 ms

• Case 1 • Case 2

Input

head =

```
[1,2,3,4,5]
```

k =

```
2
```

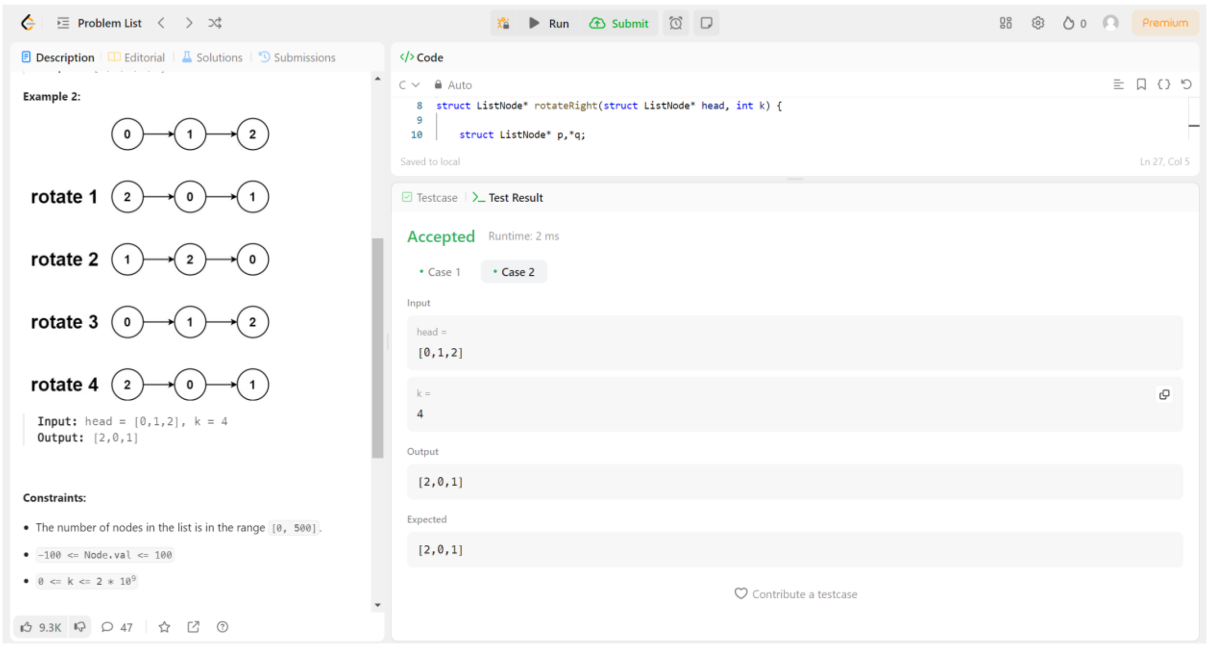
Output

```
[4,5,1,2,3]
```

Expected

```
[4,5,1,2,3]
```

Contribute a testcase



Example 2:

```
Input: head = [0,1,2], k = 4
Output: [2,0,1]
```

Constraints:

- The number of nodes in the list is in the range `[0, 500]`.
- `-100 <= Node.val <= 100`
- $0 <= k <= 2 * 10^9$

9.3K · 47

C · Auto

```
struct ListNode* rotateRight(struct ListNode* head, int k) {

    struct ListNode* p,*q;
```

Saved to local · Ln 27, Col 5

Testcase · Test Result

Accepted Runtime: 2 ms

• Case 1 • Case 2

Input

head =

```
[0,1,2]
```

k =

```
4
```

Output

```
[2,0,1]
```

Expected

```
[2,0,1]
```

Contribute a testcase

Code

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

```c
    struct ListNode* p,*q;
    if(head==NULL||head->next==NULL||k==0){
        return head;
    }
    p=head;
    int count=1;

    while(p->next!=NULL){
        p=p->next;
        count++;
    }
    k=k%count;
    if(k==0){
        return head;
    }
    p->next=head;
    p=head;

    for(int i=0;i<count-k-1;i++){
        p=p->next;
    }
    q=p->next;
    p->next=NULL;
    return q;

}
```