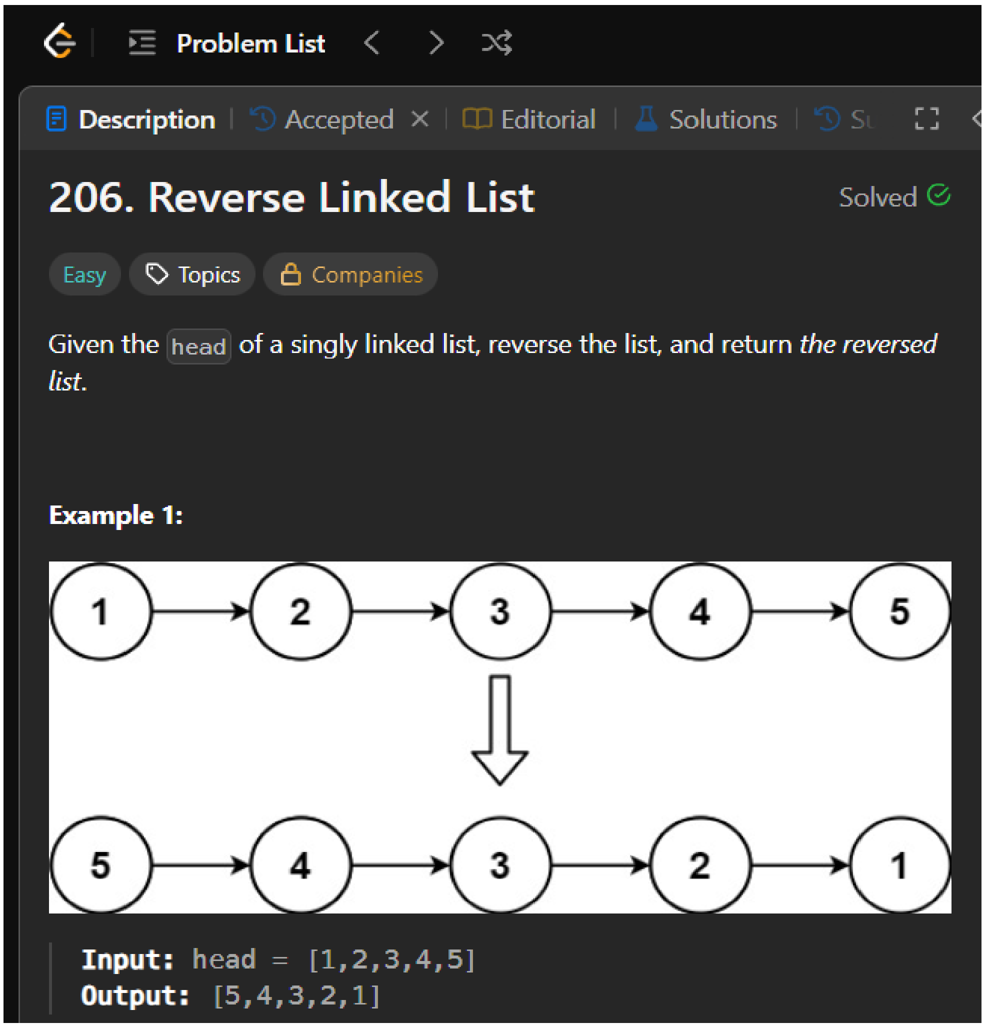
LeetCode

206 – Reverse a linked list:



206. Reverse Linked List

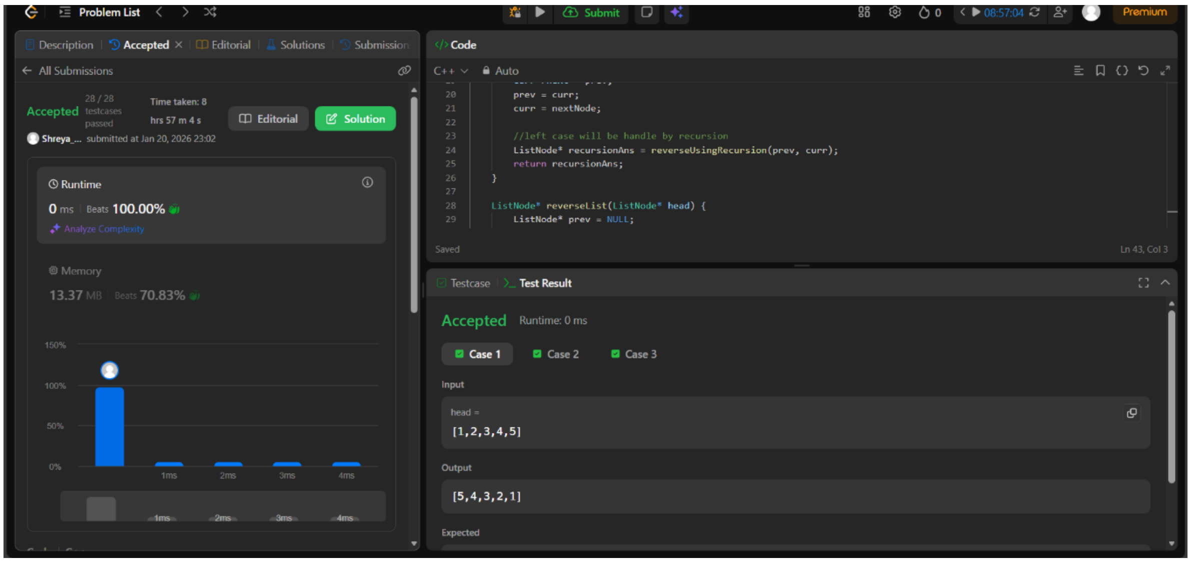
Solved

Easy · Topics · Companies

Given the `head` of a singly linked list, reverse the list, and return *the reversed list*.

Example 1:

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



Accepted 28 / 28 testcases passed Time taken: 8 hrs 57 m 4 s

Shreya_... submitted at Jan 20, 2026 23:02

Runtime 0 ms | Beats 100.00%

Analyze Complexity

Memory 13.37 MB | Beats 70.83%

Code C++ Auto

```
        prev = curr;
        curr = nextNode;

        //left case will be handle by recursion
        ListNode* recursionAns = reverseUsingRecursion(prev, curr);
        return recursionAns;
    }

    ListNode* reverseList(ListNode* head) {
        ListNode* prev = NULL;
```

Saved Ln 43, Col 3

Testcase | Test Result

Accepted Runtime: 0 ms

Case 1 Case 2 Case 3

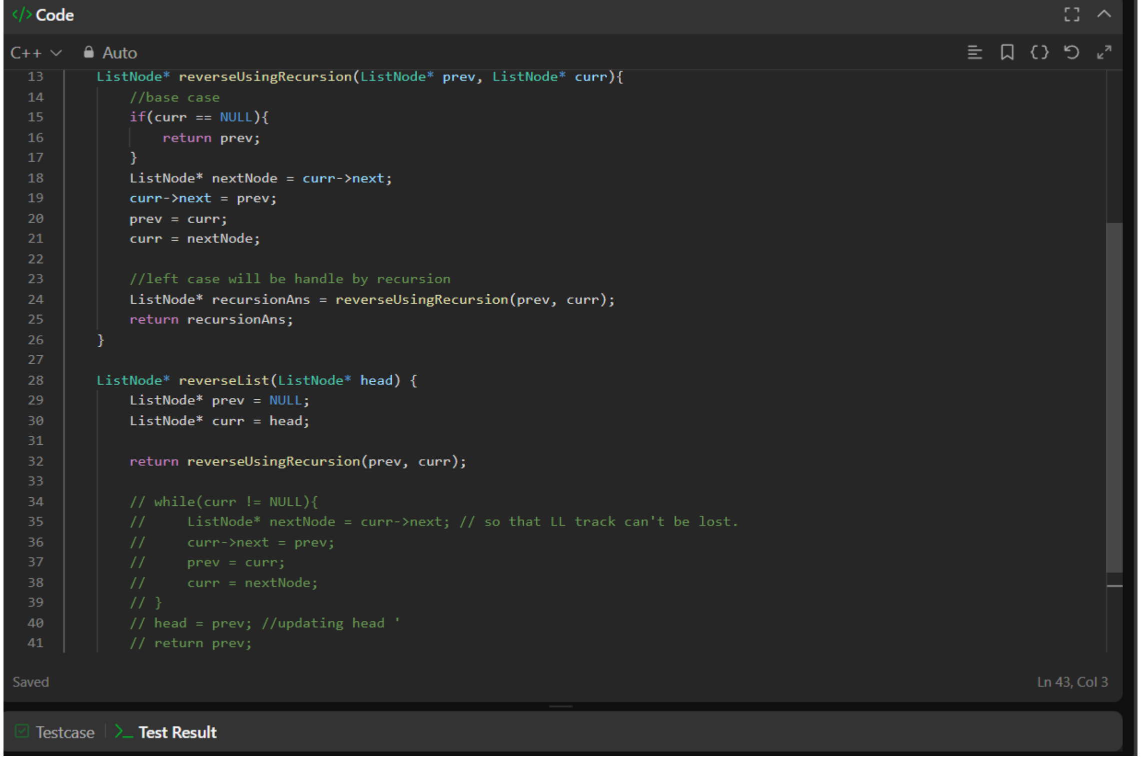
Input

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

Output

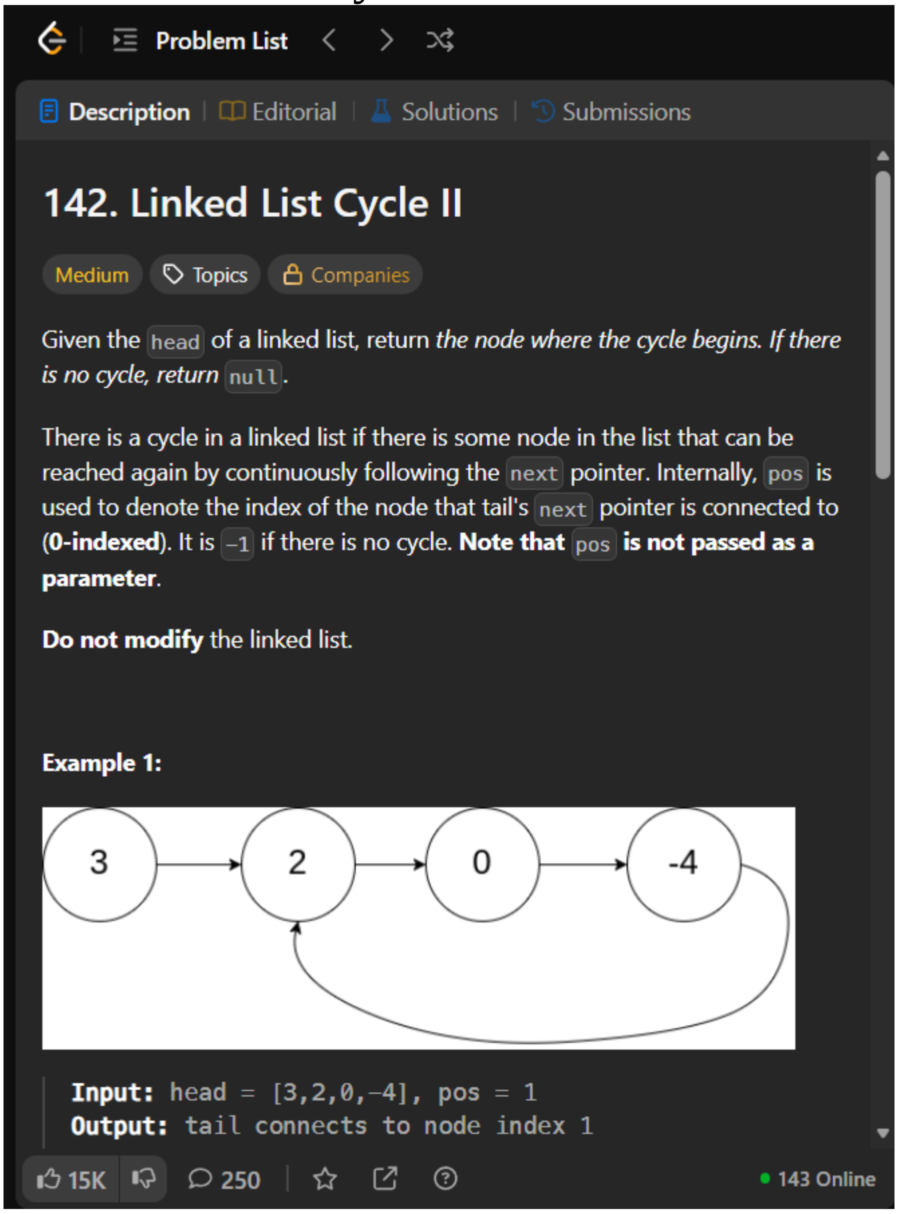
```
[5,4,3,2,1]
```

Expected



```cpp
    ListNode* reverseUsingRecursion(ListNode* prev, ListNode* curr){
        //base case
        if(curr == NULL){
            return prev;
        }
        ListNode* nextNode = curr->next;
        curr->next = prev;
        prev = curr;
        curr = nextNode;

        //left case will be handle by recursion
        ListNode* recursionAns = reverseUsingRecursion(prev, curr);
        return recursionAns;
    }

    ListNode* reverseList(ListNode* head) {
        ListNode* prev = NULL;
        ListNode* curr = head;

        return reverseUsingRecursion(prev, curr);

        // while(curr != NULL){
        //     ListNode* nextNode = curr->next; // so that LL track can't be lost.
        //     curr->next = prev;
        //     prev = curr;
        //     curr = nextNode;
        // }
        // head = prev; //updating head '
        // return prev;
```

142. Linked List Cycle II:



142. Linked List Cycle II

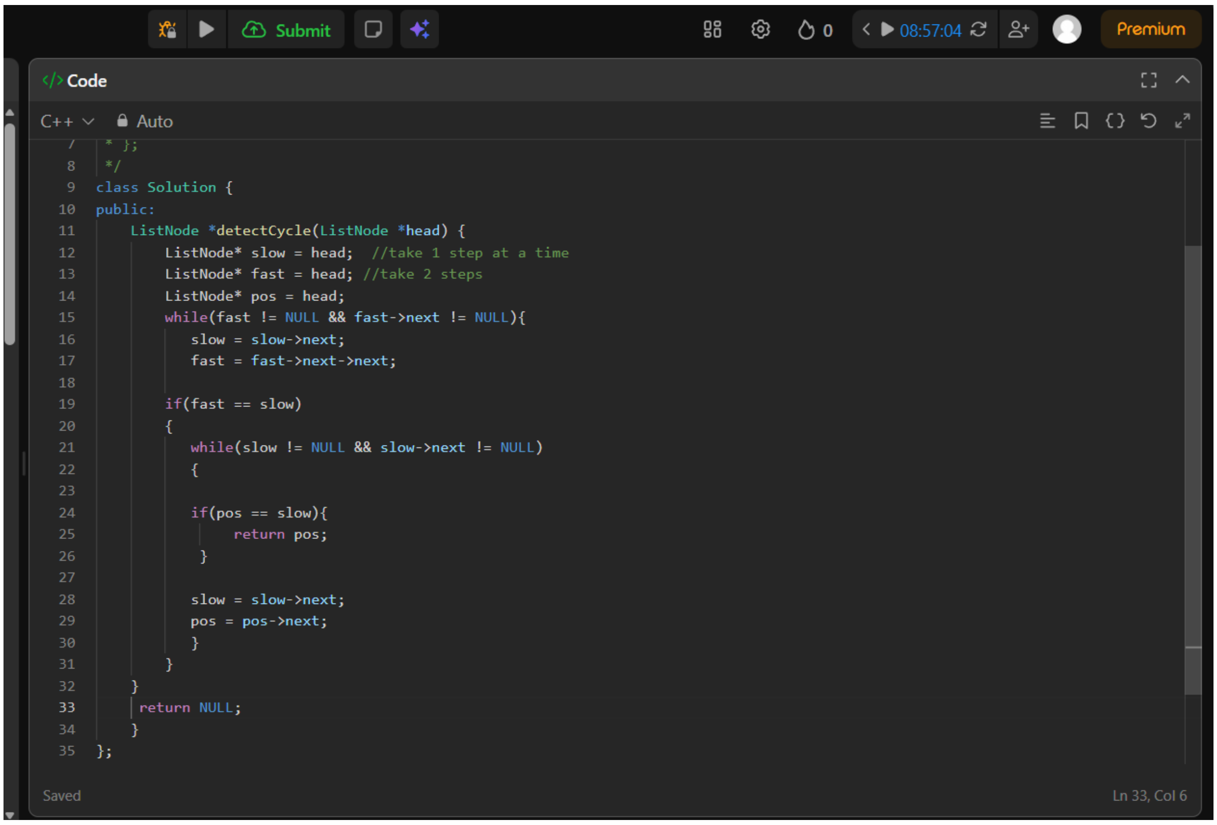
Given the `head` of a linked list, return *the node where the cycle begins. If there is no cycle, return* `null`.

There is a cycle in a linked list if there is some node in the list that can be reached again by continuously following the `next` pointer. Internally, `pos` is used to denote the index of the node that tail's `next` pointer is connected to (**0-indexed**). It is `-1` if there is no cycle. **Note that** `pos` **is not passed as a parameter**.

Do not modify the linked list.

Example 1:

```
Input: head = [3,2,0,-4], pos = 1
Output: tail connects to node index 1
```



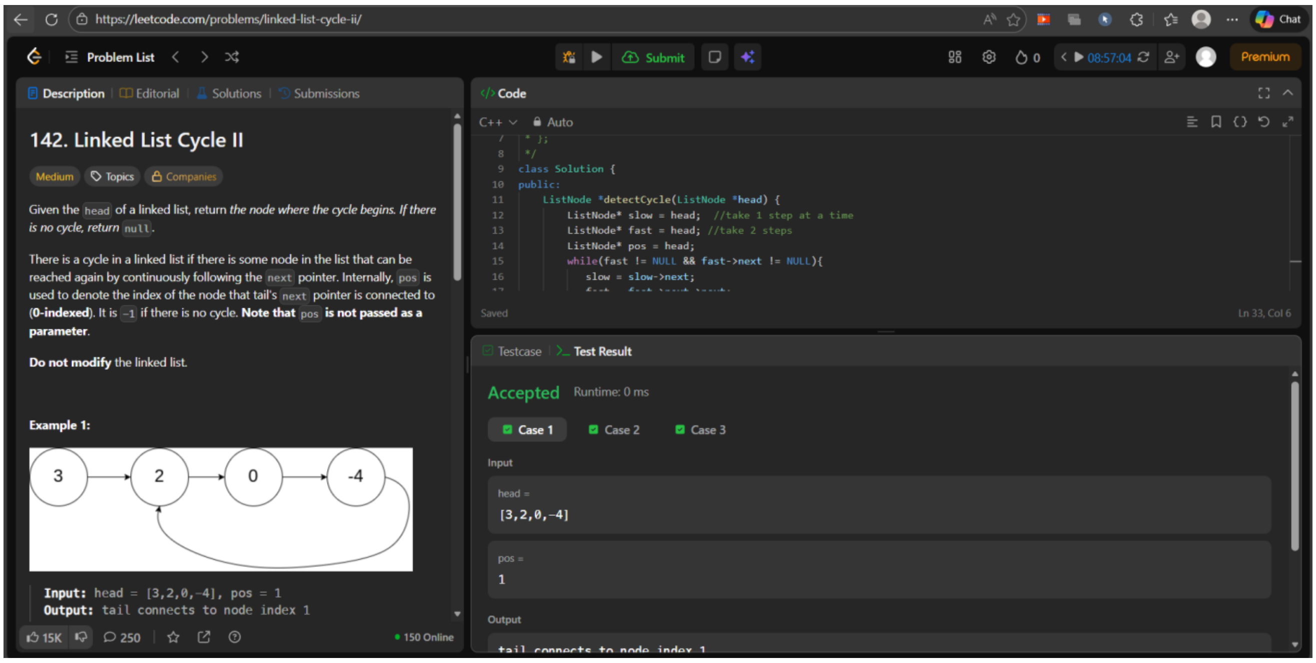
Submit 08:57:04 Premium

Code

C++ Auto

```cpp
 * };
 */
class Solution {
public:
    ListNode *detectCycle(ListNode *head) {
        ListNode* slow = head;  //take 1 step at a time
        ListNode* fast = head; //take 2 steps
        ListNode* pos = head;
        while(fast != NULL && fast->next != NULL){
           slow = slow->next;
           fast = fast->next->next;

        if(fast == slow)
        {
           while(slow != NULL && slow->next != NULL)
           {

           if(pos == slow){
                return pos;
            }

           slow = slow->next;
           pos = pos->next;
           }
        }
    }
     return NULL;
    }
};
```

Saved Ln 33, Col 6



https://leetcode.com/problems/linked-list-cycle-ii/

Problem List Submit 08:57:04 Premium

Description | Editorial | Solutions | Submissions

142. Linked List Cycle II

Medium Topics Companies

Given the `head` of a linked list, return *the node where the cycle begins. If there is no cycle, return* `null`.

There is a cycle in a linked list if there is some node in the list that can be reached again by continuously following the `next` pointer. Internally, `pos` is used to denote the index of the node that tail's `next` pointer is connected to (**0-indexed**). It is `-1` if there is no cycle. **Note that** `pos` **is not passed as a parameter**.

Do not modify the linked list.

Example 1:

```
Input: head = [3,2,0,-4], pos = 1
Output: tail connects to node index 1
```

15K 250 150 Online

Testcase | Test Result

Accepted Runtime: 0 ms

Case 1 Case 2 Case 3

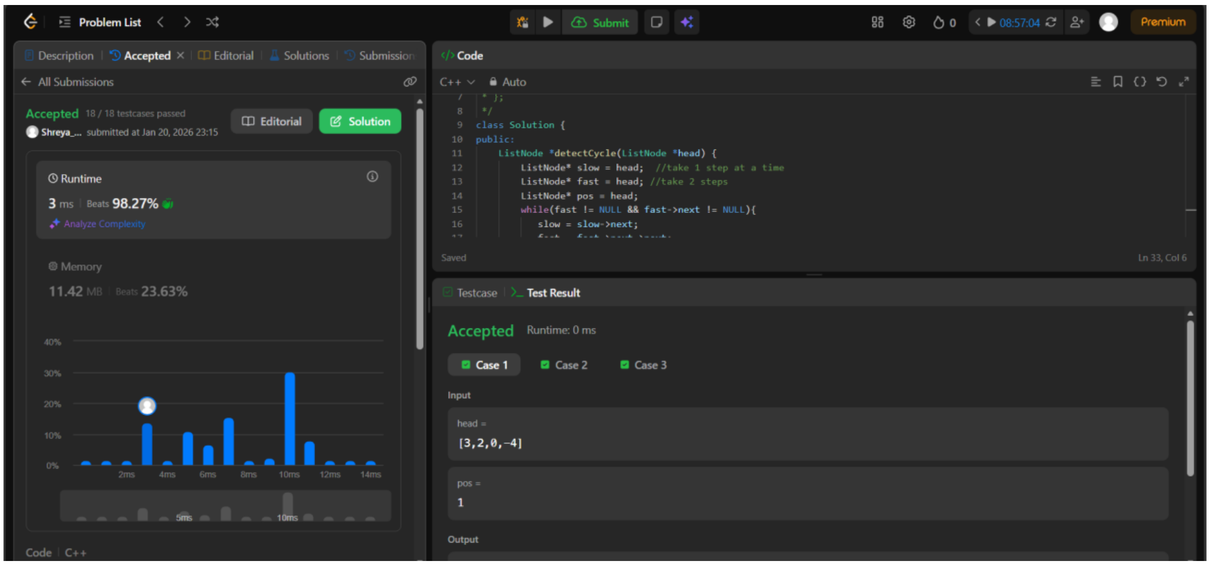
Input

```
head =
[3,2,0,-4]
```

```
pos =
1
```

Output

```
tail connects to node index 1
```



Problem List Submit 08:57:04 Premium

Description | Accepted | Editorial | Solutions | Submission

All Submissions

Accepted 18 / 18 testcases passed

Shreya_... submitted at Jan 20, 2026 23:15

Editorial Solution

Runtime

3 ms | Beats 98.27%

Analyze Complexity

Memory

11.42 MB | Beats 23.63%

Code | C++

Testcase | Test Result

Accepted Runtime: 0 ms

Case 1 Case 2 Case 3

Input

```
head =
[3,2,0,-4]
```

```
pos =
1
```

Output