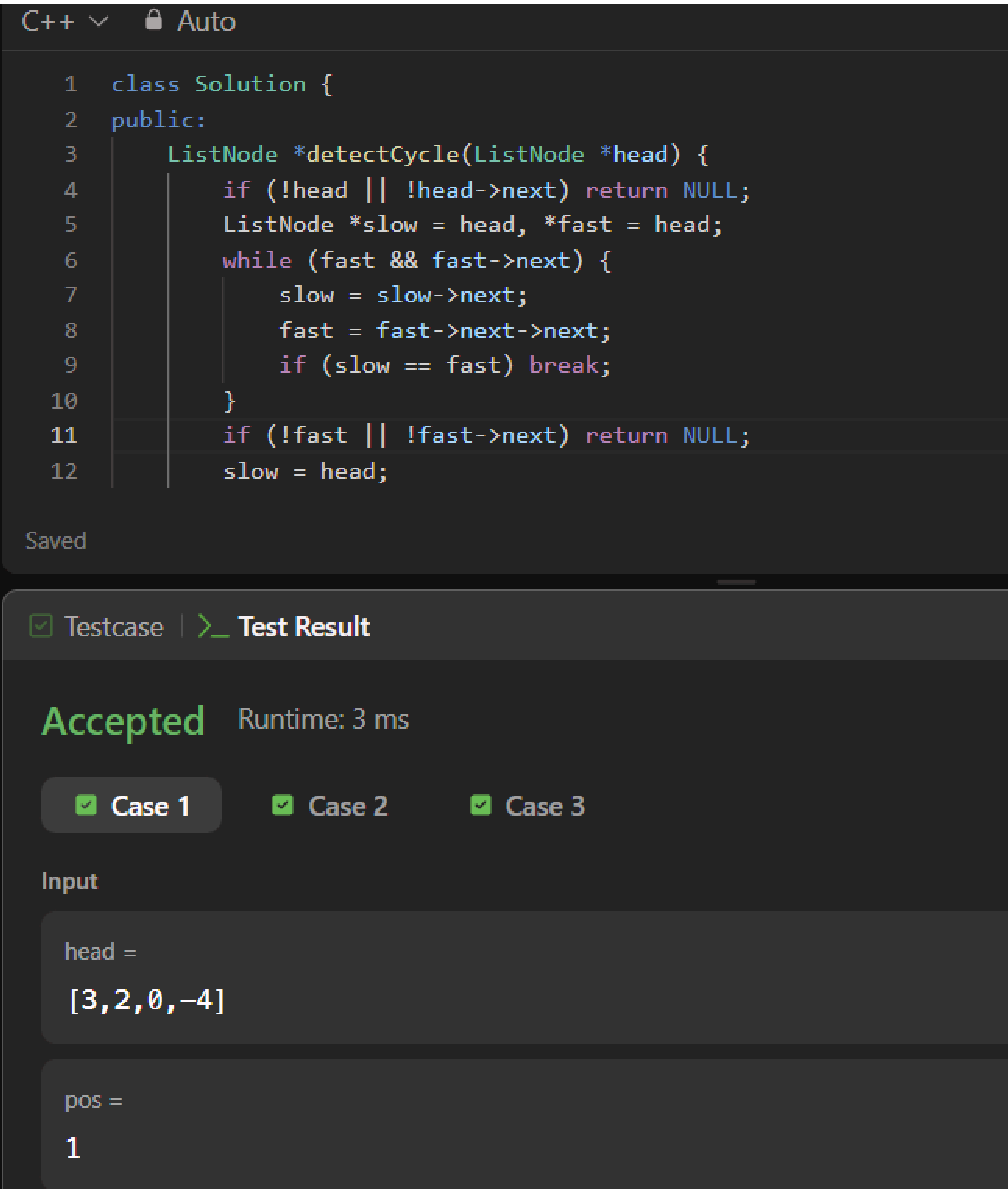
PROBLEM:-142

```
class Solution {
public:
    ListNode *detectCycle(ListNode *head) {
        if (!head || !head->next) return NULL;
        ListNode *slow = head, *fast = head;
        while (fast && fast->next) {
            slow = slow->next;
            fast = fast->next->next;
            if (slow == fast) break;
        }
        if (!fast || !fast->next) return NULL;
        slow = head;
        while (slow != fast) {
            slow = slow->next;
            fast = fast->next;
        }
        return slow;
    }
};
```



```cpp
class Solution {
public:
    ListNode *detectCycle(ListNode *head) {
        if (!head || !head->next) return NULL;
        ListNode *slow = head, *fast = head;
        while (fast && fast->next) {
            slow = slow->next;
            fast = fast->next->next;
            if (slow == fast) break;
        }
        if (!fast || !fast->next) return NULL;
        slow = head;
```

Accepted Runtime: 3 ms

✓ Case 1 ✓ Case 2 ✓ Case 3

Input

head =

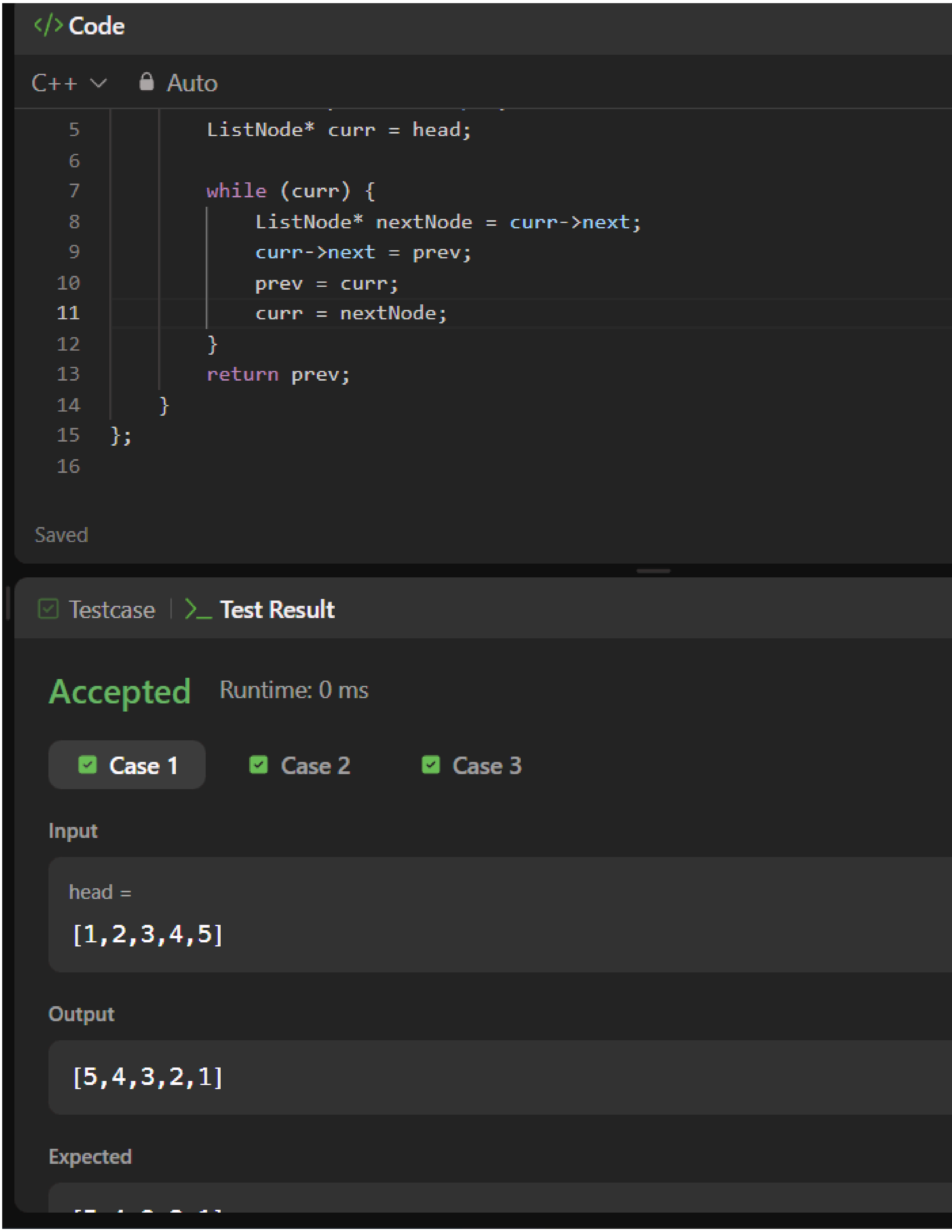
```
[3,2,0,-4]
```

pos =

```
1
```

PROBLEM:- 206

```
class Solution {

public:

    ListNode* reverseList(ListNode* head) {

        ListNode* prev = nullptr;

        ListNode* curr = head;

        while (curr) {

            ListNode* nextNode = curr->next;

            curr->next = prev;

            prev = curr;

            curr = nextNode;

        }

        return prev;

    }

};
```


</> Code

C++ Auto

```cpp
        ListNode* curr = head;

        while (curr) {
            ListNode* nextNode = curr->next;
            curr->next = prev;
            prev = curr;
            curr = nextNode;
        }
        return prev;
    }
};
```

Saved

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