

Reverse linked list

```
class Solution {
public:
    ListNode* reverseList(ListNode* head) {
        ListNode* prev = nullptr;
        ListNode* next = nullptr;
        ListNode* curr = head;
        while (curr != nullptr) {
            next = curr->next;
            curr->next = prev;
            prev = curr;
            curr = next;
        }

        return prev;
    }
};
```

The screenshot displays a code editor interface for a C++ solution. The left sidebar shows the 'Problem List' and 'Submissions' tabs. The main editor area displays the C++ code for reversing a linked list. The right sidebar shows the 'Code' tab with the same code, and the 'Test Result' tab showing 'Accepted' status with a runtime of 0 ms. Below the code editor, there is a performance metrics section showing 'Runtime: 0 ms' and 'Memory: 13.23 MB'. A bar chart shows the runtime performance across different test cases, with the first case being the most time-consuming. The bottom of the interface shows the 'Input' field for the test cases.

Problem List < > 🔍

Description Editorial Solutions Accepted Submissions

All Submissions

Accepted 28 / 28 testcases passed

12ananya submitted at Feb 13, 2025 21:47

Editorial Solution

Runtime 0 ms | Beats 100.00% 🏆

Memory 13.23 MB | Beats 90.90% 🏆

Analyze Complexity

150% 100% 50% 0%

1ms 2ms 3ms 4ms

1ms 2ms 3ms 4ms

Code C++

</> Code

C++ v Auto

```
9 *;
10 */
11 class Solution {
12 public:
13     ListNode* reverseList(ListNode* head) {
14         ListNode* prev = nullptr;
15         ListNode* next = nullptr;
16         ListNode* curr = head;
17         while (curr != nullptr) {
18             next = curr->next;
19             curr->next = prev;
20             prev = curr;
21             curr = next;
22         }
23     }
24 }
```

Saved Ln 20, Col 26

Testcase > Test Result

Accepted Runtime: 0 ms

Case 1 Case 2 Case 3

Input