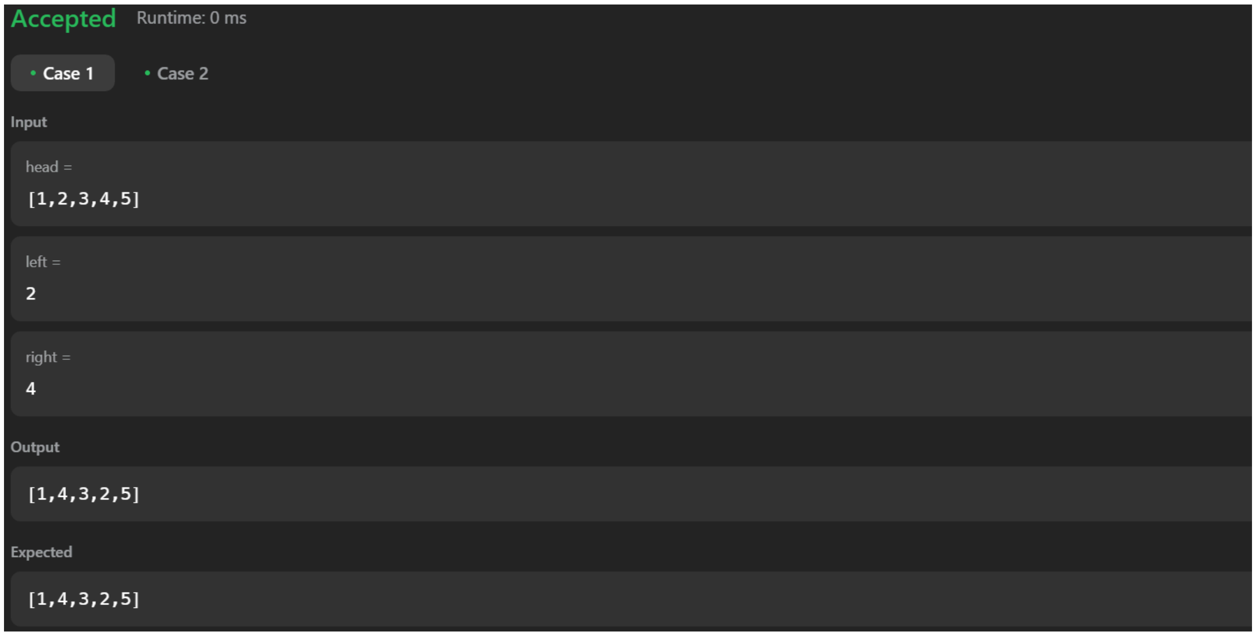
```c
/**
 * Definition for singly-linked list.
 * struct ListNode {
 *     int val;
 *     struct ListNode *next;
 * };
 */

struct ListNode* reverseBetween(struct ListNode* head, int left, int right) {
    if (head == NULL || left == right) {
        return head;
    }

    struct ListNode* dummy = (struct ListNode*)malloc(sizeof(struct ListNode));
    dummy->next = head;
    struct ListNode* prev = dummy;

    // Move to the node just before the reversal starts
    for (int i = 1; i < left; ++i) {
        prev = prev->next;
    }

    // Reverse the nodes from left to right
    struct ListNode* current = prev->next;
    struct ListNode* next = NULL;
    struct ListNode* tail = current;

    for (int i = left; i <= right; ++i) {
        struct ListNode* temp = current->next;
        current->next = next;
        next = current;
        current = temp;
    }

    // Connect the reversed portion with the rest of the list
    prev->next = next;
    tail->next = current;

    // Return the modified list
    struct ListNode* result = dummy->next;
    free(dummy); // Free the dummy node
    return result;
}
```



Accepted Runtime: 0 ms

• Case 1 • Case 2

Input

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

```
left =
2
```

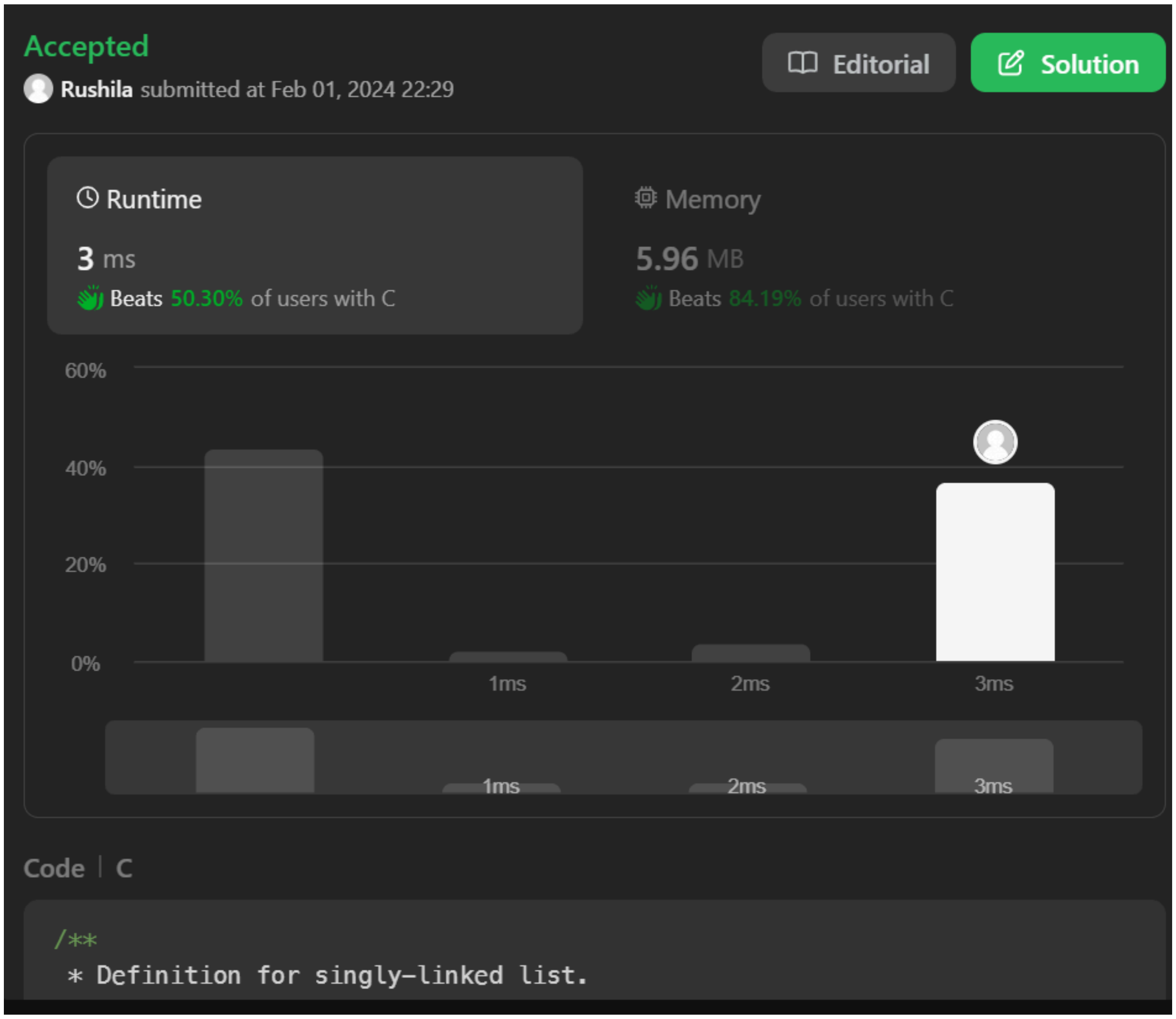
```
right =
4
```

Output

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

Expected

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


Accepted

Rushila submitted at Feb 01, 2024 22:29

Runtime

3 ms

Beats 50.30% of users with C

Memory

5.96 MB

Beats 84.19% of users with C

Code | C

```
/**
 * Definition for singly-linked list.
```