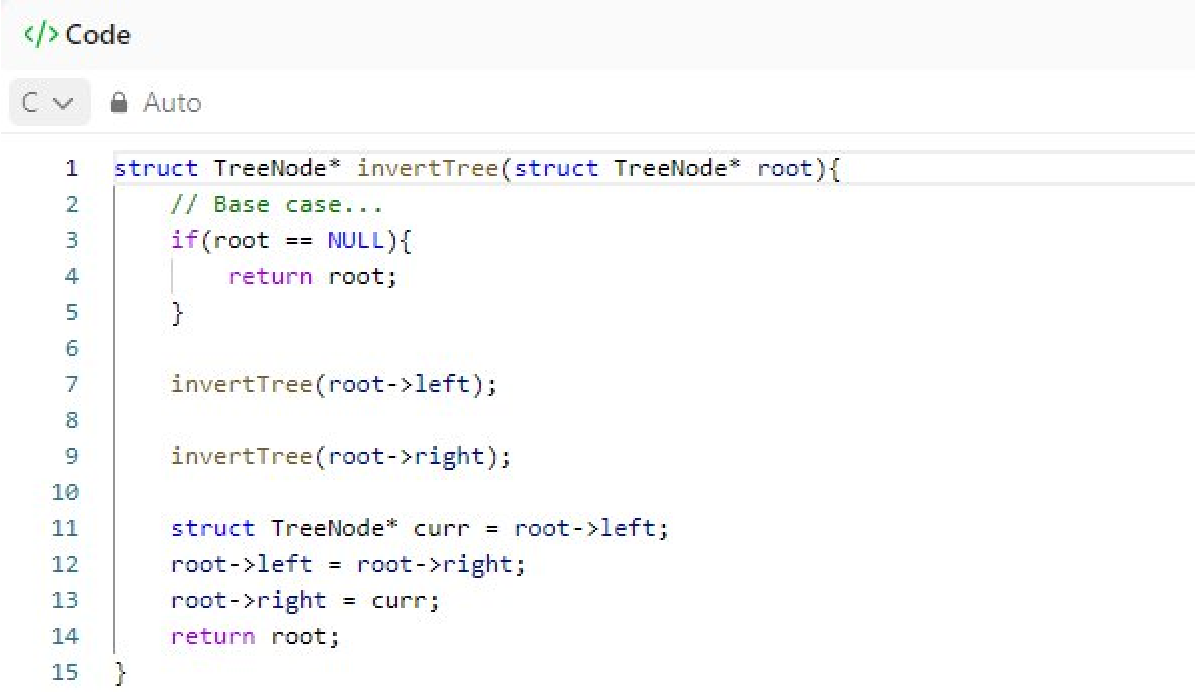
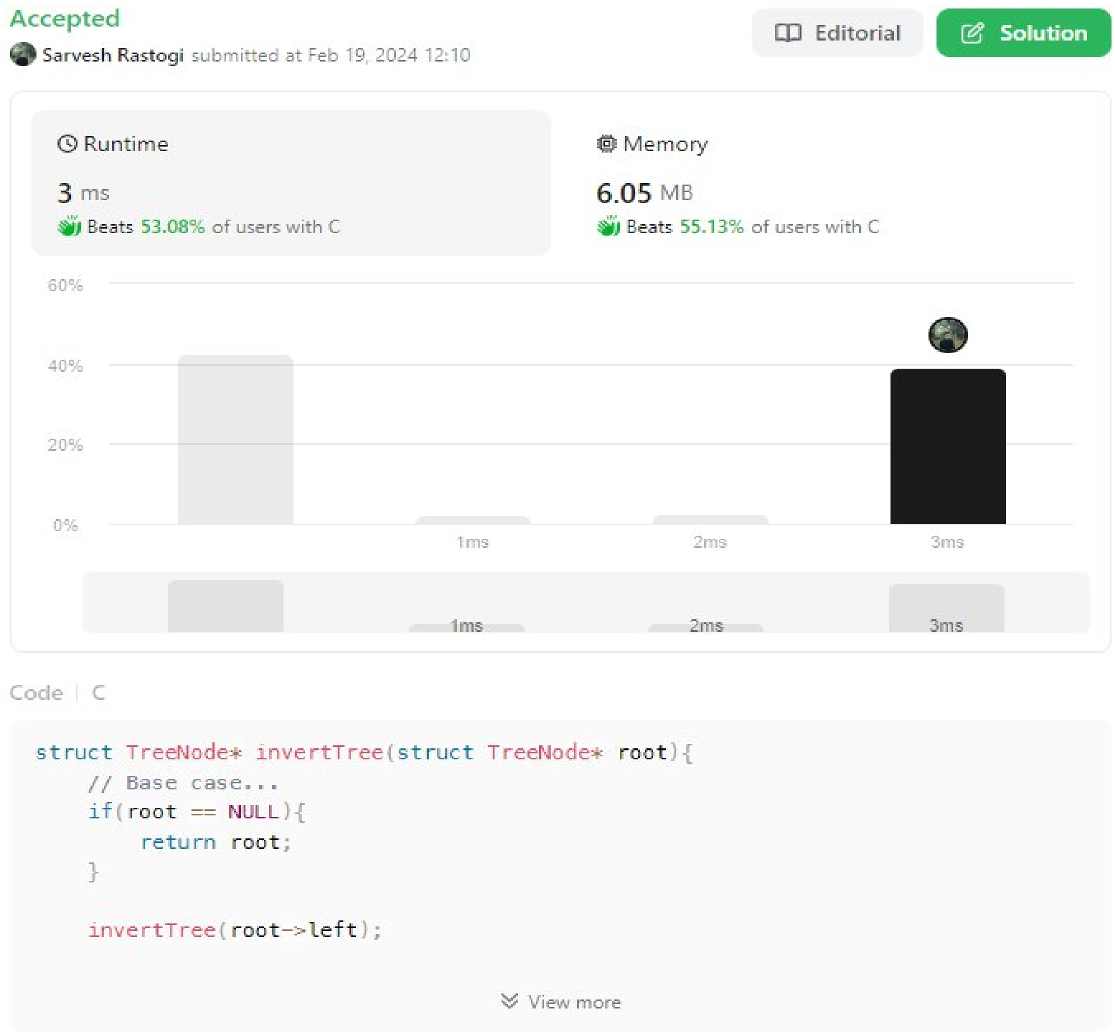


```c
struct TreeNode* invertTree(struct TreeNode* root){
    // Base case...
    if(root == NULL){
        return root;
    }

    invertTree(root->left);

    invertTree(root->right);

    struct TreeNode* curr = root->left;
    root->left = root->right;
    root->right = curr;
    return root;
}
```



Accepted

Sarvesh Rastogi submitted at Feb 19, 2024 12:10

Runtime

3 ms

Beats 53.08% of users with C

Memory

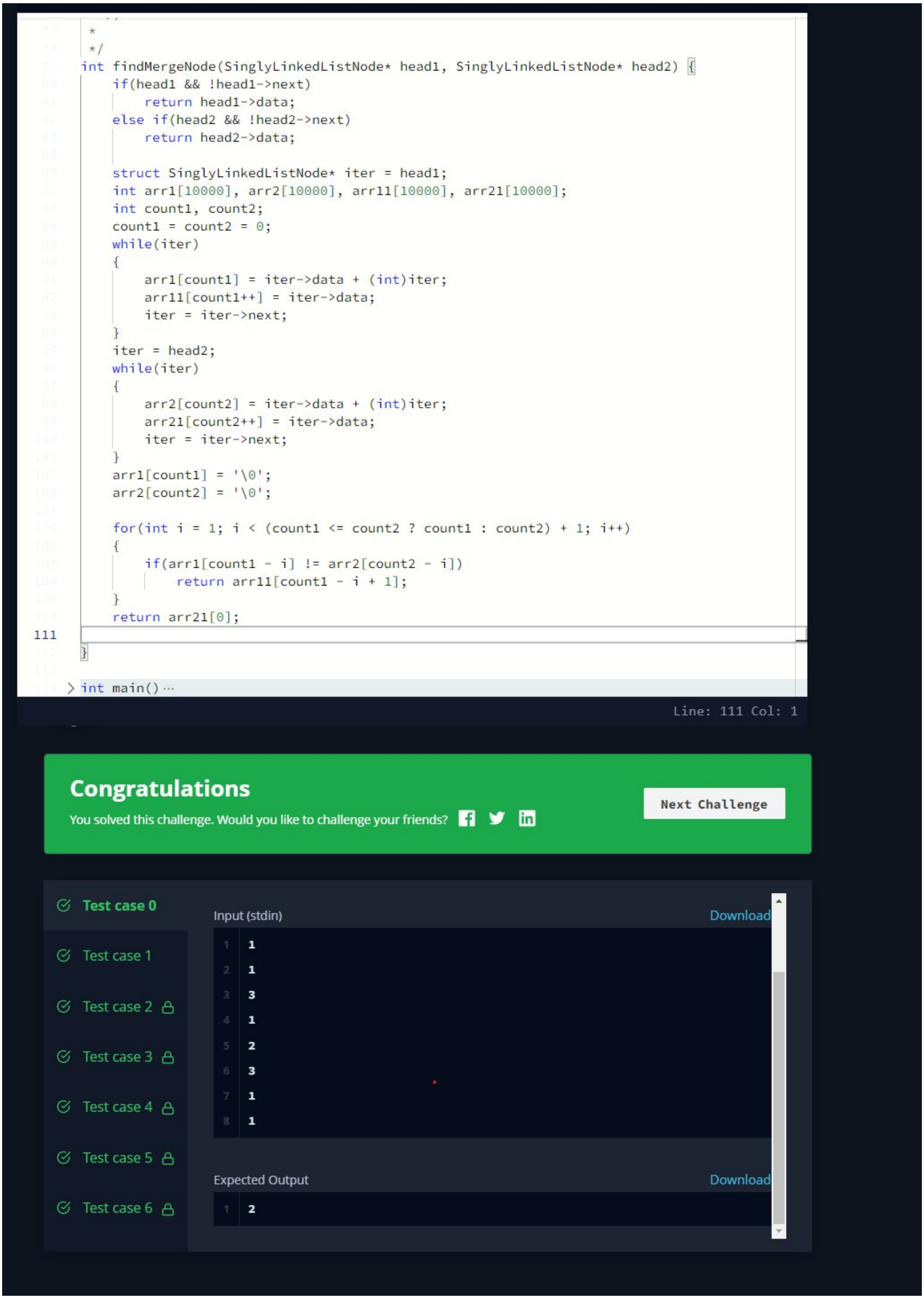
6.05 MB

Beats 55.13% of users with C

Code | C

```c
struct TreeNode* invertTree(struct TreeNode* root){
    // Base case...
    if(root == NULL){
        return root;
    }

    invertTree(root->left);
```



```
 *
 */
int findMergeNode(SinglyLinkedListNode* head1, SinglyLinkedListNode* head2) {
    if(head1 && !head1->next)
        return head1->data;
    else if(head2 && !head2->next)
        return head2->data;

    struct SinglyLinkedListNode* iter = head1;
    int arr1[10000], arr2[10000], arr11[10000], arr21[10000];
    int count1, count2;
    count1 = count2 = 0;
    while(iter)
    {
        arr1[count1] = iter->data + (int)iter;
        arr11[count1++] = iter->data;
        iter = iter->next;
    }
    iter = head2;
    while(iter)
    {
        arr2[count2] = iter->data + (int)iter;
        arr21[count2++] = iter->data;
        iter = iter->next;
    }
    arr1[count1] = '\0';
    arr2[count2] = '\0';

    for(int i = 1; i < (count1 <= count2 ? count1 : count2) + 1; i++)
    {
        if(arr1[count1 - i] != arr2[count2 - i])
            return arr11[count1 - i + 1];
    }
    return arr21[0];

}

> int main() ···
```

Line: 111 Col: 1

Congratulations

You solved this challenge. Would you like to challenge your friends?

Input (stdin)

```
1
1
3
1
2
3
1
1
```

Expected Output

```
2
```