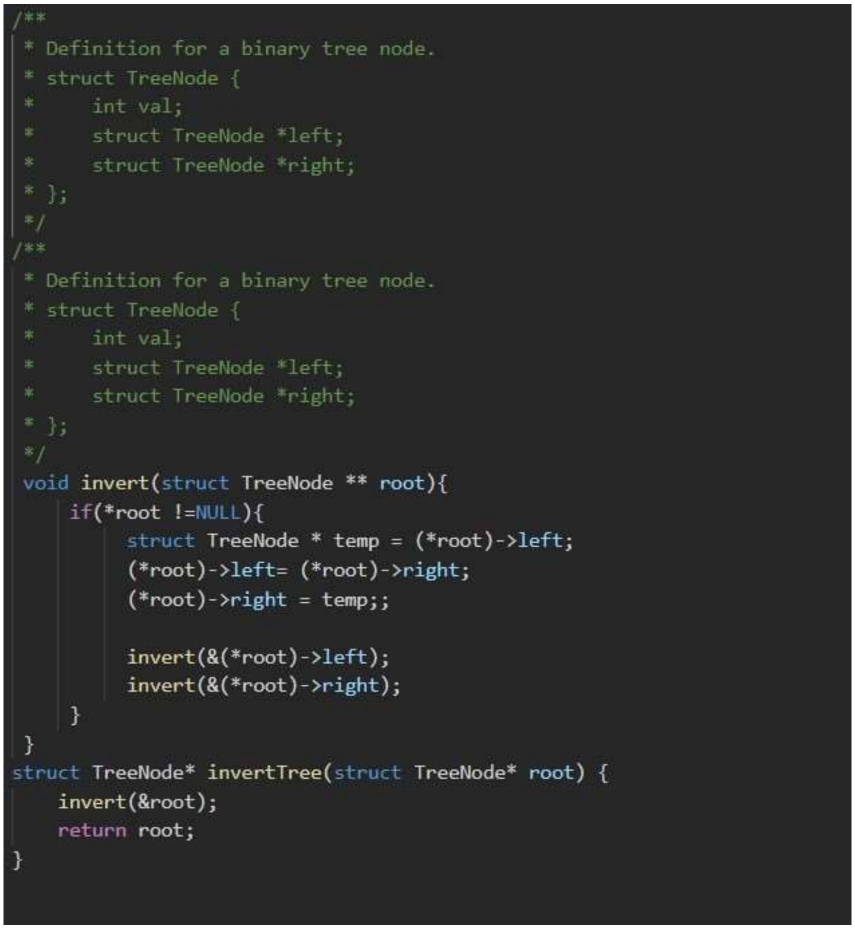
2.Invert Binary Tree

Shashank Patel C J

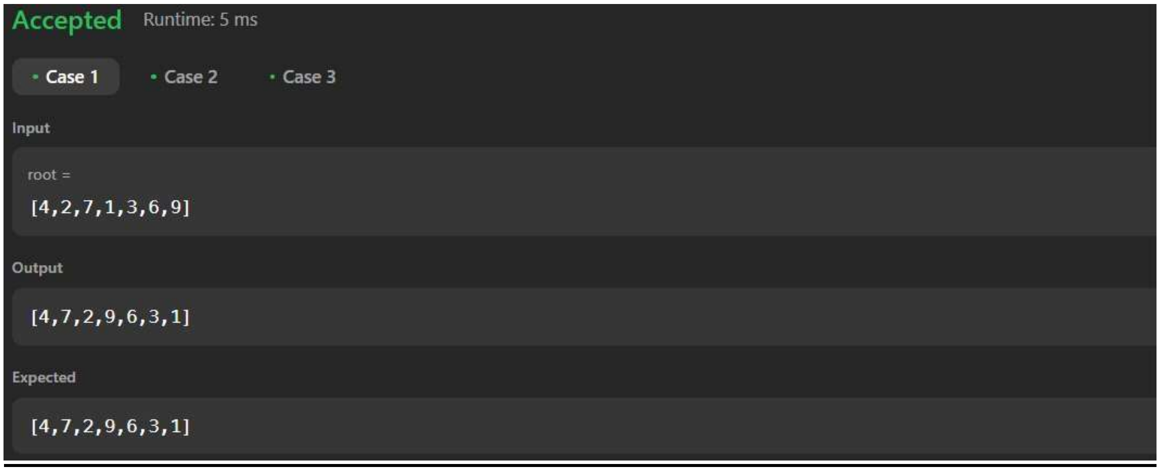
1BM22CS255



```c
/**
 * Definition for a binary tree node.
 * struct TreeNode {
 *     int val;
 *     struct TreeNode *left;
 *     struct TreeNode *right;
 * };
 */
/**
 * Definition for a binary tree node.
 * struct TreeNode {
 *     int val;
 *     struct TreeNode *left;
 *     struct TreeNode *right;
 * };
 */
 void invert(struct TreeNode ** root){
     if(*root !=NULL){
          struct TreeNode * temp = (*root)->left;
          (*root)->left= (*root)->right;
          (*root)->right = temp;;

          invert(&(*root)->left);
          invert(&(*root)->right);
     }
 }
struct TreeNode* invertTree(struct TreeNode* root) {
    invert(&root);
    return root;
}
```

Output:



Accepted Runtime: 5 ms

Input

root =

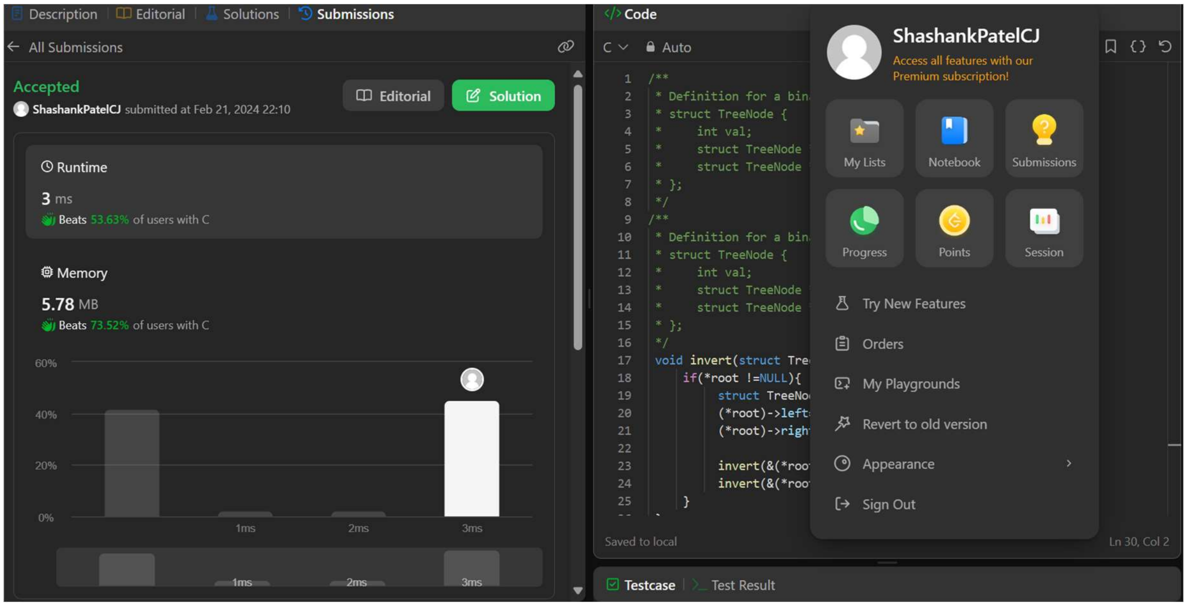
```
[4,2,7,1,3,6,9]
```

Output

```
[4,7,2,9,6,3,1]
```

Expected

```
[4,7,2,9,6,3,1]
```



Accepted

ShashankPatelCJ submitted at Feb 21, 2024 22:10

Runtime

3 ms

Beats 53.63% of users with C

Memory

5.78 MB

Beats 73.52% of users with C

```
/**
 * Definition for a bin
 * struct TreeNode {
 *     int val;
 *     struct TreeNode
 *     struct TreeNode
 * };
 */
/**
 * Definition for a bin
 * struct TreeNode {
 *     int val;
 *     struct TreeNode
 *     struct TreeNode
 * };
 */
void invert(struct Tre
    if(*root !=NULL){
        struct TreeNo
        (*root)->left
        (*root)->righ

        invert(&(*roo
        invert(&(*roo
    }
```

Saved to local

Ln 30, Col 2