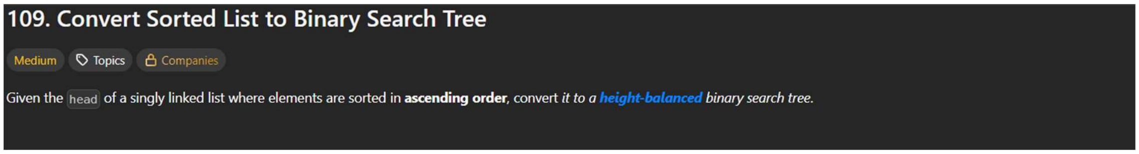
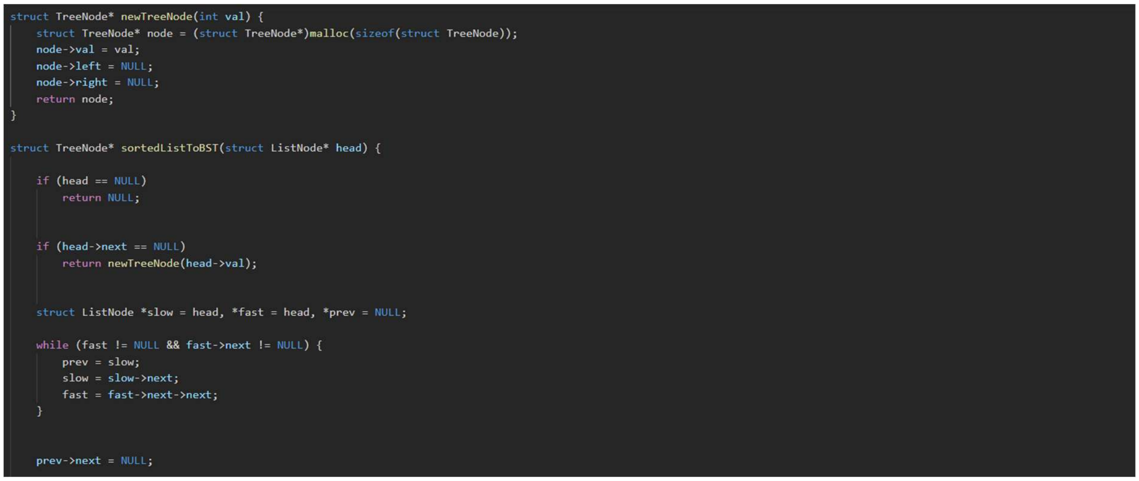
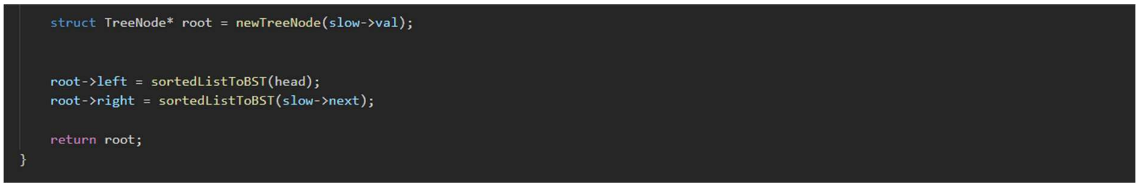


109. Convert Sorted List to Binary Search Tree

Given the `head` of a singly linked list where elements are sorted in **ascending order**, convert *it to a **height-balanced** binary search tree.*

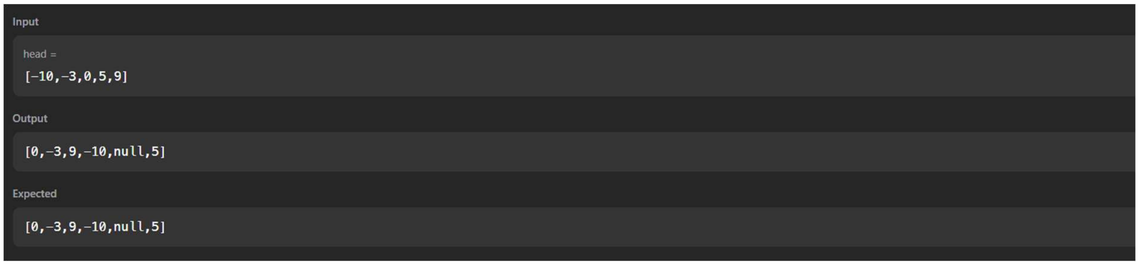


```
struct TreeNode* newTreeNode(int val) {
    struct TreeNode* node = (struct TreeNode*)malloc(sizeof(struct TreeNode));
    node->val = val;
    node->left = NULL;
    node->right = NULL;
    return node;
}

struct TreeNode* sortedListToBST(struct ListNode* head) {

    if (head == NULL)
        return NULL;


    if (head->next == NULL)
        return newTreeNode(head->val);


    struct ListNode *slow = head, *fast = head, *prev = NULL;

    while (fast != NULL && fast->next != NULL) {
        prev = slow;
        slow = slow->next;
        fast = fast->next->next;
    }


    prev->next = NULL;
```



```
    struct TreeNode* root = newTreeNode(slow->val);


    root->left = sortedListToBST(head);
    root->right = sortedListToBST(slow->next);

    return root;
}
```

Output:



Input

head =

```
[-10,-3,0,5,9]
```

Output

```
[0,-3,9,-10,null,5]
```

Expected

```
[0,-3,9,-10,null,5]
```