

[All Contests](#) > [In21-CS2023-Lab7](#) > [Binary Search Tree : Insertion](#)

Binary Search Tree : Insertion

Problem

Submissions

Leaderboard

Discussions

You are given a pointer to the root of a binary search tree and values to be inserted into the tree. Insert the values into their appropriate position in the binary search tree and return the root of the updated binary tree. You just have to complete the function.

Input Format

You are given a function,

```
Node * insert (Node * root ,int data) {
}
```

Constraints

- No. of nodes in the tree ≤ 500

Output Format

Return the root of the binary search tree after inserting the value into the tree.

Sample Input



The value to be inserted is 6.

Sample Output


[f](#) [t](#) [in](#)

Contest ends in a day

Submissions: 153

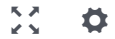
Max Score: 40

Difficulty: Easy

Rate This Challenge:

[More](#)

C++



```
1 #include <iostream>
2
3 using namespace std;
4
5 class Node {
6     public:
7         int data;
8         Node *left;
9         Node *right;
10        Node(int d) {
11            data = d;
12            left = NULL;
13            right = NULL;
14        }
15 };
16
17 class Solution {
18     public:
19
20     void preOrder(Node *root) {
21
22         if( root == NULL )
23             return;
24
25         std::cout << root->data << " ";
26
27         preOrder(root->left);
28         preOrder(root->right);
29     }
30
31     /*
32     Node is defined as
33
34     class Node {
35     public:
36         int data;
37         Node *left;
38         Node *right;
39         Node(int d) {
40             data = d;
41             left = NULL;
42             right = NULL;
43         }
44     };
45
46     */
47     Node * insert(Node * root, int data) {
48         // checking whether the root exists
49         if (root == NULL){
50             // base case
51             Node *temp = new Node(data);
52             return temp;
53         }
54
55         // recursion
56         if (data < root->data){
57             root->left = insert(root->left, data);
58         } else if (data > root->data){
```

```
59         root->right = insert(root->right, data);
60     }
61
62     return root;
63 }
```

```
64 };
65
66 int main() {
67     Solution myTree;
68     Node* root = NULL;
69
70     int t;
71     int data;
72
73     std::cin >> t;
74
75     while(t-- > 0) {
76         std::cin >> data;
77         root = myTree.insert(root, data);
78     }
79
80     myTree.preOrder(root);
81
82     return 0;
83 }
84
85 }
```

Line: 50 Col: 25

[Upload Code as File](#) ☐ Test against custom input

Run Code

Submit Code

Testcase 0 **Congratulations, you passed the sample test case.**Click the **Submit Code** button to run your code against all the test cases.

Input (stdin)

```
6
4 2 3 1 7 6
```

Your Output (stdout)

```
4 2 1 3 7 6
```

Expected Output

```
4 2 1 3 7 6
```