

PREPARE<sup>NEW</sup>

CERTIFY

COMPETE

Search



yasinarafat2413 ▾

[All Contests](#) > [Assignment 04](#) | [Basic Data Structures](#) | [Batch 03](#) > [Sum of Tree](#)

# Sum of Tree

Problem

Submissions

Leaderboard

Discussions

Submitted 6 hours ago • Score: 20.00

Status: **Accepted**

Test Case #0



Test Case #1



Test Case #2



Test Case #3



Test Case #4



Test Case #5



Test Case #6



Test Case #7



Test Case #8



Test Case #9



Test Case #10



Test Case #11

## Submitted Code

Language: C++20

Open in editor

```
1 #include <bits/stdc++.h>
2 using namespace std;
3
4 class Node
5 {
6 public:
7     int val;
8     Node *left;
9     Node *right;
10    Node(int val)
11    {
12        this->val = val;
13        this->left = NULL;
14        this->right = NULL;
15    }
16 };
17
18 Node *inputTree()
19 {
20     int val;
21     cin >> val;
22     Node *root;
23     if (val == -1)
24         root = NULL;
25     else
26         root = new Node(val);
27
28     queue<Node *> q;
29     if (root)
30         q.push(root);
31
32     while (!q.empty())
```

```
33     {
34         Node *f = q.front();
35         q.pop();
36
37         int l, r;
38         cin >> l >> r;
39         Node *myLeft, *myRight;
40
41         if (l == -1)
42             myLeft = NULL;
43         else
44             myLeft = new Node(l);
45
46         if (r == -1)
47             myRight = NULL;
48         else
49             myRight = new Node(r);
50
51         f->left = myLeft;
52         f->right = myRight;
53
54         if (f->left)
55             q.push(f->left);
56         if (f->right)
57             q.push(f->right);
58     }
59     return root;
60 }
61
62 long long int levelOrderSum(Node *root)
63 {
64     if(root == NULL)
65     {
66         return 0;
67     }
68     queue<Node *> q;
69     q.push(root);
70
71
72     long long int sum =0;
73     while(!q.empty())
74     {
75         Node *f = q.front();
76         q.pop();
77
78         // cout<<f->val<<" ";
79         sum+= f->val;
80
81         if(f->left) q.push(f->left);
82         if(f->right) q.push(f->right);
83     }
84     return sum;
85 }
86
87 int main()
88 {
89     Node *root = inputTree();
90     long long int s = levelOrderSum(root);
91     cout<<s<<endl;
92
93     return 0;
94 }
```

[Interview Prep](#) | [Blog](#) | [Scoring](#) | [Environment](#) | [FAQ](#) | [About Us](#) | [Support](#) | [Careers](#) | [Terms Of Service](#) | [Privacy Policy](#) |