Print Tree

Problem | Submissions | Leaderboard | Discussions

Submitted 5 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

Submitted Code

Language: C++20 Open in editor

```
#include <bits/stdc++.h>
using namespace std;

class Node
{
public:
    int val;
    Node *left;
    Node *right;
    Node(int val)
    {
        this->val = val;
        this->left = NULL;
        this->right = NULL;
    }
};

stack<int>st;

Node *inputTree()
{
    int val;
    cin >> val;
    Node *root;
    if (val == -1)
        root = NULL;
    else
        root = new Node(val);

    queue<Node *> q;
    if (root)
        q.push(root);

```

```
    while (!q.empty())
    {
        Node *f = q.front();
        q.pop();

        int l, r;
        cin >> l >> r;
        Node *myLeft, *myRight;

        if (l == -1)
            myLeft = NULL;
        else
            myLeft = new Node(l);

        if (r == -1)
            myRight = NULL;
        else
            myRight = new Node(r);

        f->left = myLeft;
        f->right = myRight;

        if (f->left)
            q.push(f->left);
        if (f->right)
            q.push(f->right);
    }
    return root;
}

void levelOrder(Node *root)
{
    if(root == NULL)
    {
        return;
    }
    queue<Node *> q;
    q.push(root);



    while(!q.empty())
    {
        Node *f = q.front();
        q.pop();

        // cout<<f->val<<" ";
        st.push(f->val);

        if(f->right) q.push(f->right);
        if(f->left) q.push(f->left);

    }

}

int main()
{
    Node *root = inputTree();
    levelOrder(root);

    while(!st.empty())
    {
        cout<<st.top()<<" ";
        st.pop();
    }
```

```
100
101     return 0;
102 }
```