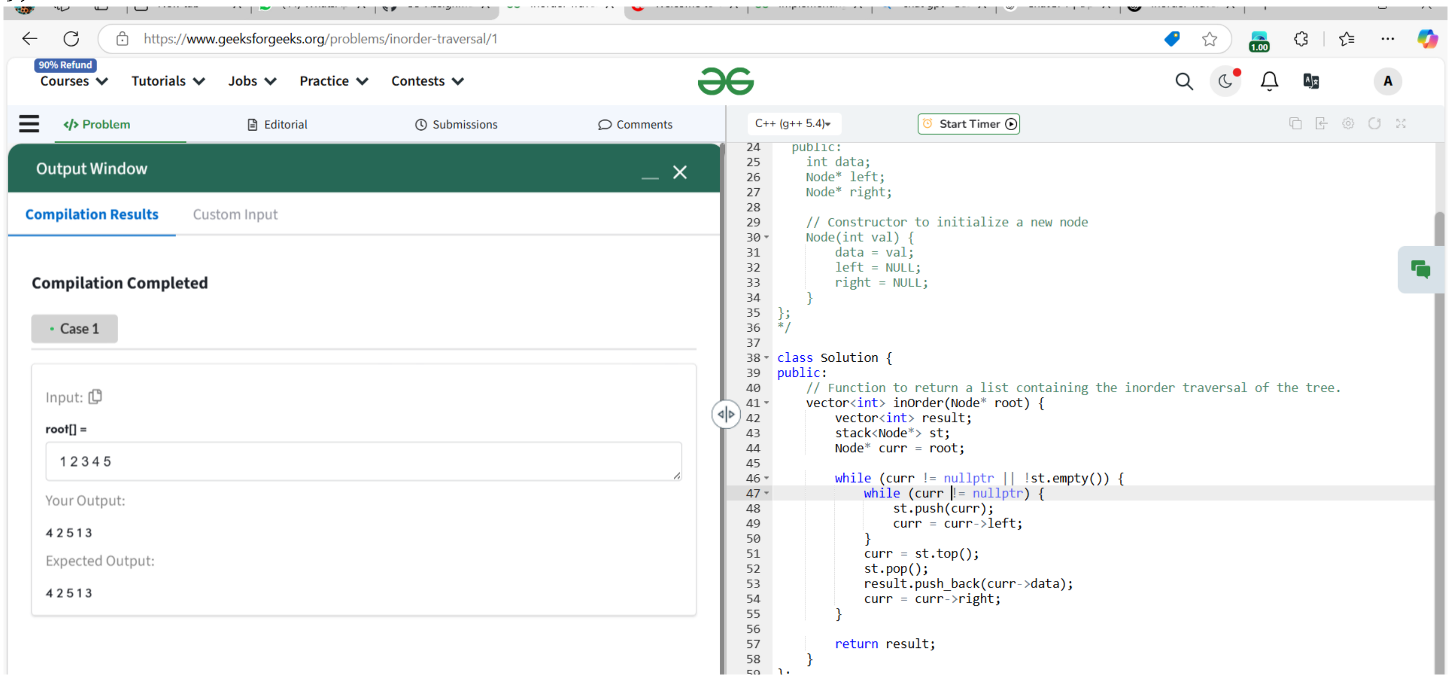
Inorder traversal using stack

```
class Solution {
public:
    // Function to return a list containing the inorder traversal of the tree.
    vector<int> inOrder(Node* root) {
        vector<int> result;
        stack<Node*> st;
        Node* curr = root;

        while (curr != nullptr || !st.empty()) {
            while (curr != nullptr) {
                st.push(curr);
                curr = curr->left;
            }
            curr = st.top();
            st.pop();
            result.push_back(curr->data);
            curr = curr->right;
        }

        return result;
    }
};
```



https://www.geeksforgeeks.org/problems/inorder-traversal/1

Output Window

Compilation Results | Custom Input

Compilation Completed

Case 1

```
Input:
root[] =
1 2 3 4 5
Your Output:
4 2 5 1 3
Expected Output:
4 2 5 1 3
```

C++ (g++ 5.4)

```
  public:
    int data;
    Node* left;
    Node* right;

    // Constructor to initialize a new node
    Node(int val) {
        data = val;
        left = NULL;
        right = NULL;
    }
};
*/

class Solution {
public:
    // Function to return a list containing the inorder traversal of the tree.
    vector<int> inOrder(Node* root) {
        vector<int> result;
        stack<Node*> st;
        Node* curr = root;

        while (curr != nullptr || !st.empty()) {
            while (curr != nullptr) {
                st.push(curr);
                curr = curr->left;
            }
            curr = st.top();
            st.pop();
            result.push_back(curr->data);
            curr = curr->right;
        }

        return result;
    }
```