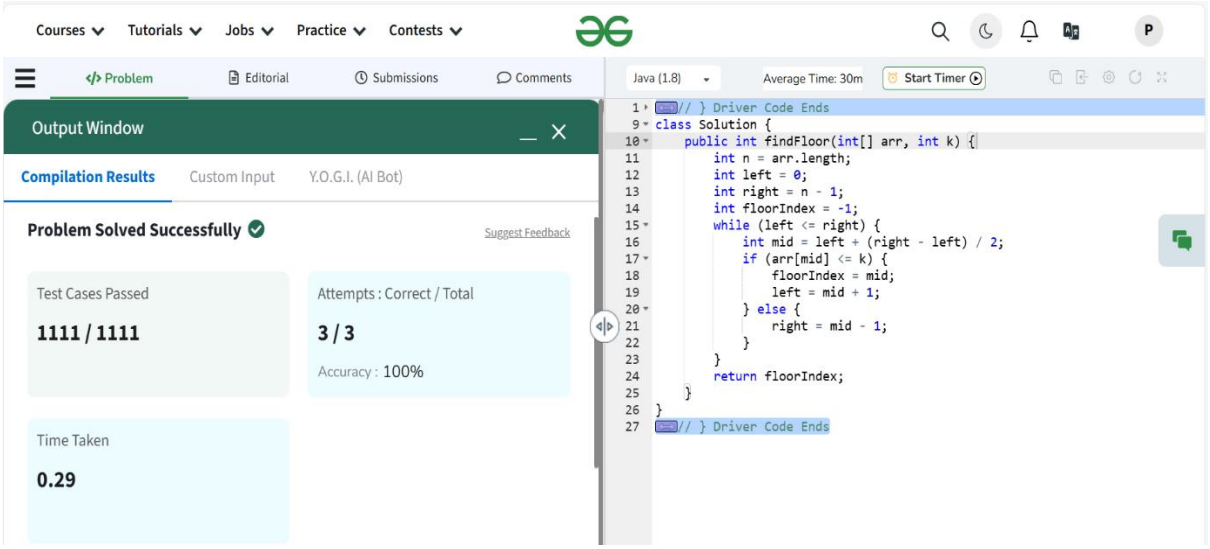
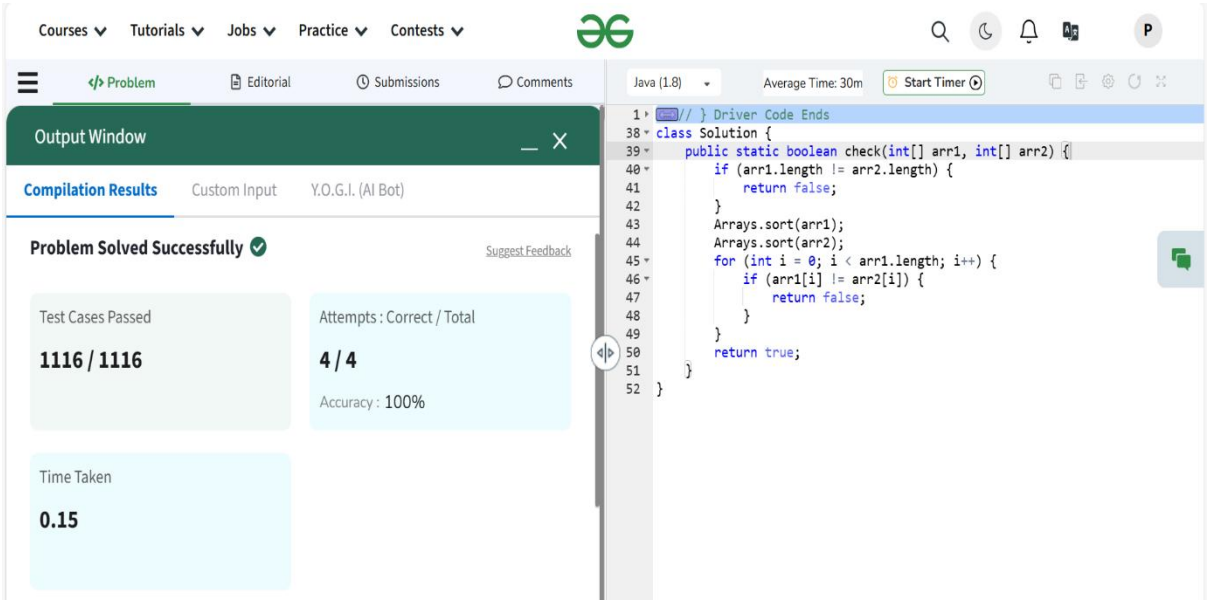
DSA Practice – 2

1. Floor in sorted array



Time Complexity : O(log n)

2. Check equal arrays

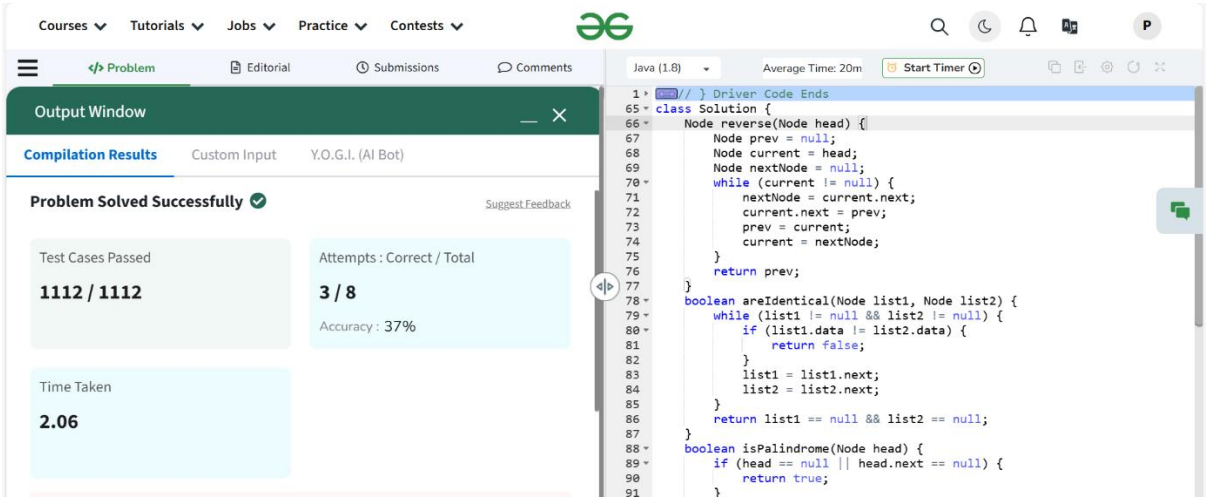


Time Complexity : O(n)

3.Palindrome linked list

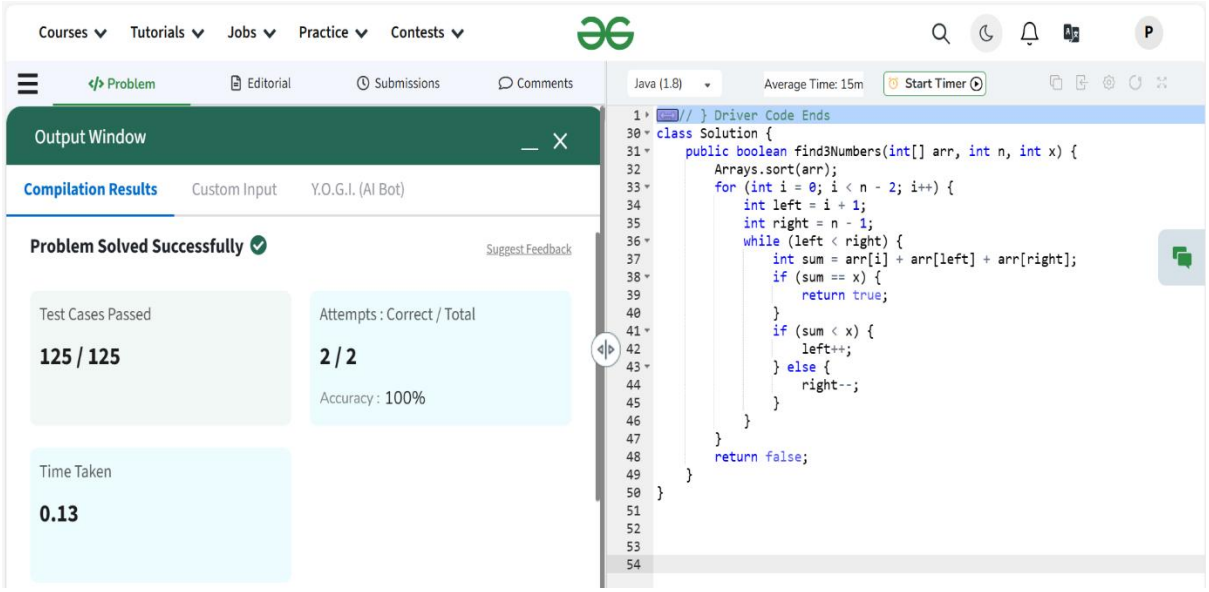
```
class Solution {
    Node reverse(Node head) {
        Node prev = null;
        Node current = head;
        Node nextNode = null;
        while (current != null) {
            nextNode = current.next;
            current.next = prev;
            prev = current;
            current = nextNode;
        }
        return prev;
    }
    boolean areIdentical(Node list1, Node list2) {
        while (list1 != null && list2 != null) {
            if (list1.data != list2.data) {
                return false;
            }
            list1 = list1.next;
            list2 = list2.next;
        }
        return list1 == null && list2 == null;
    }
    boolean isPalindrome(Node head) {
        if (head == null || head.next == null) {
            return true;
```

```
        }
        Node slow = head;
        Node fast = head;
        while (fast != null && fast.next != null) {
            slow = slow.next;
            fast = fast.next.next;
        }
        Node secondHalf = reverse(slow);
        Node firstHalf = head;
        while (secondHalf != null) {
            if (firstHalf.data != secondHalf.data) {
                return false;
            }
            firstHalf = firstHalf.next;
            secondHalf = secondHalf.next;
        }
        return true;
    }
}
```



Time Complexity : O(n)

4.Triplet sum in array



Time Complexity : O(n ^ 2)