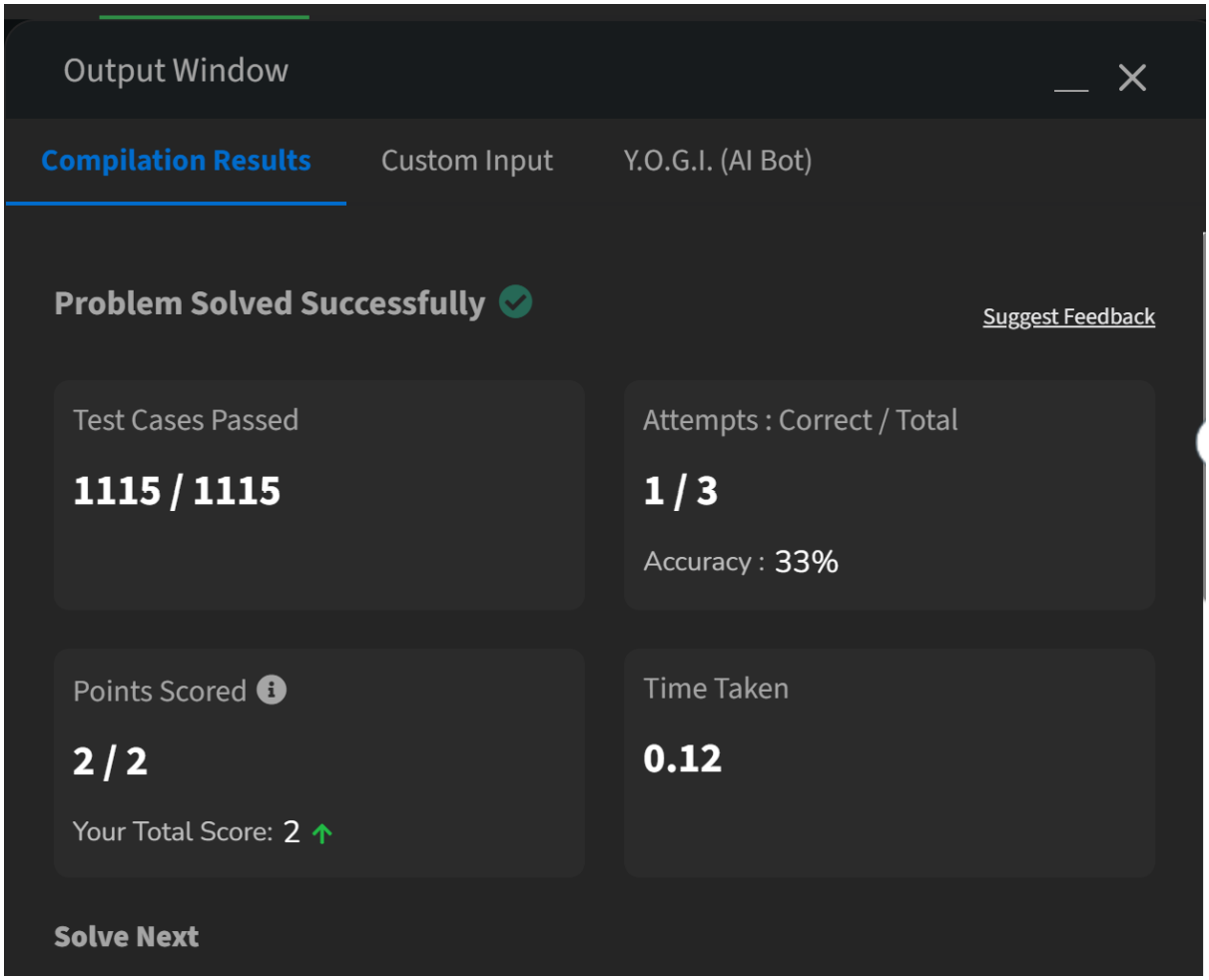
Qualcomm Interview Coding Questions

- Code 1:- Find length of Loop

```
class Solution {
  public:
    int lengthOfLoop(Node *head) {
        Node* slow = head;
        Node* fast = head;

        while (fast && fast->next) {
            slow = slow->next;
            fast = fast->next->next;

            if (slow == fast) {
                int count = 1;
                Node* temp = slow->next;
                while (temp != slow) {
                    count++;
                    temp = temp->next;
                }
                return count;
            }
        }
        return 0;
    }
};
```

Qualcomm Interview Coding Questions

Output:-



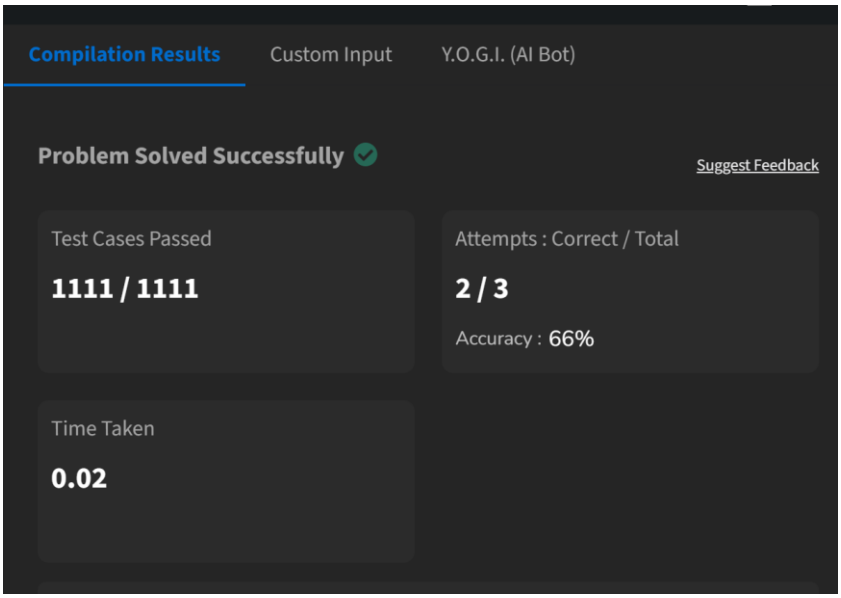
- Code 2:- Implement strstr

```
class Solution {
  public:
    int firstOccurence(string& txt, string& pat) {
        // code here
          int n = txt.size();
    int m = pat.size();


    if (m == 0) return 0;
```

Qualcomm Interview Coding Questions

```
  for (int i = 0; i <= n - m;i++) {
    int j = 0;
    while (j < m && txt[i+j]==pat[j] ) {
      j++;
    }
    if (j == m) {
      return i;
    }
  }
  return -1;
  }
};
```

Output:-



- Code 3:- Left View of Binary Tree

/* A binary tree node

struct Node

Qualcomm Interview Coding Questions

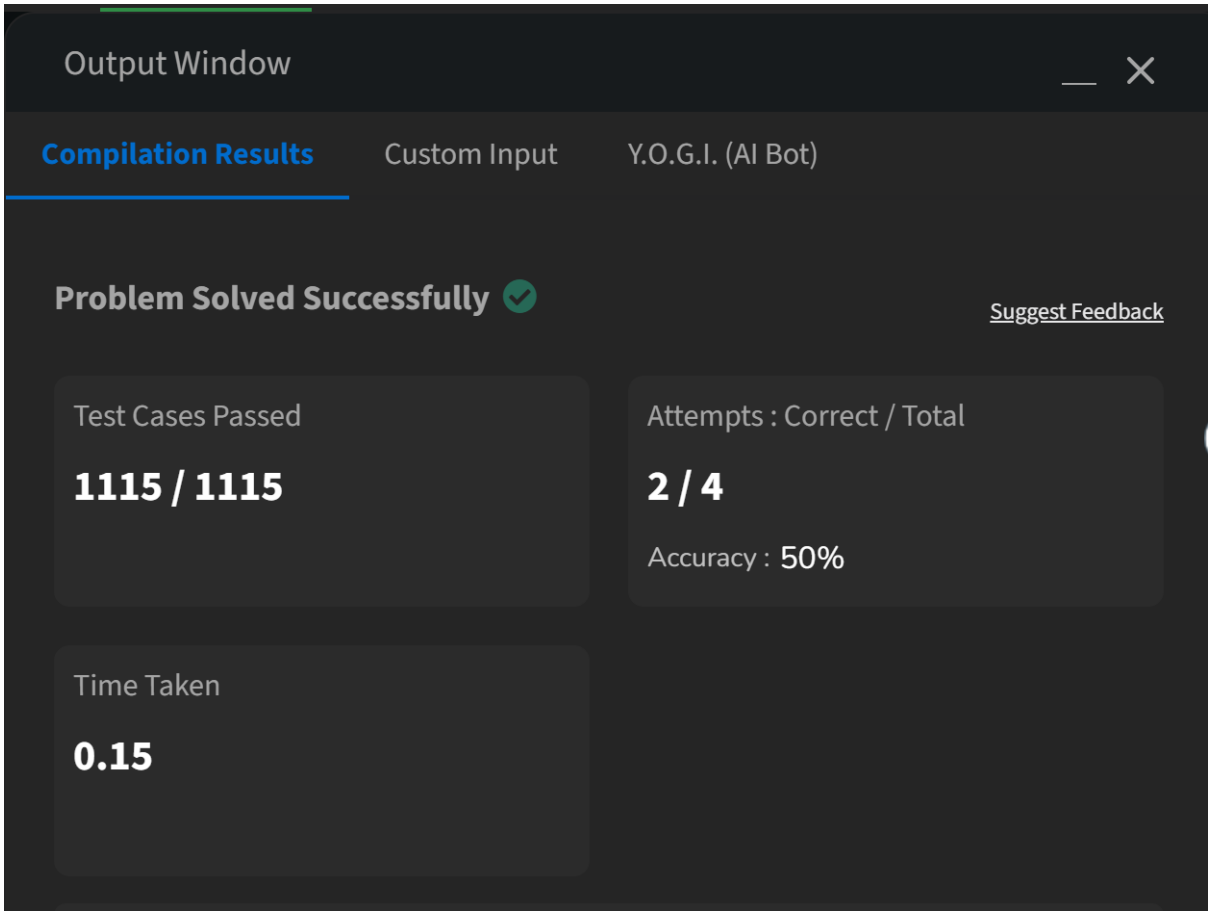
```
{
   int data;
   struct Node* left;
   struct Node* right;

   Node(int x){
      data = x;
      left = right = NULL;
   }
};
 */

#include <vector>
#include <queue>
using namespace std;

class Solution {
  public:
   vector<int> leftView(Node *root) {
      vector<int> result;
      if(root == nullptr)
         return result;

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

      while(!q.empty()) {
```

Qualcomm Interview Coding Questions

```
        int n = q.size();

        for(int i = 0; i < n; i++) {
            Node* node = q.front();
            q.pop();

            if(i == 0)
                result.push_back(node->data);

            if(node->left != nullptr)
                q.push(node->left);
            if(node->right != nullptr)
                q.push(node->right);
        }
      }

      return result;
    }
};
```

Output:-

Qualcomm Interview Coding Questions

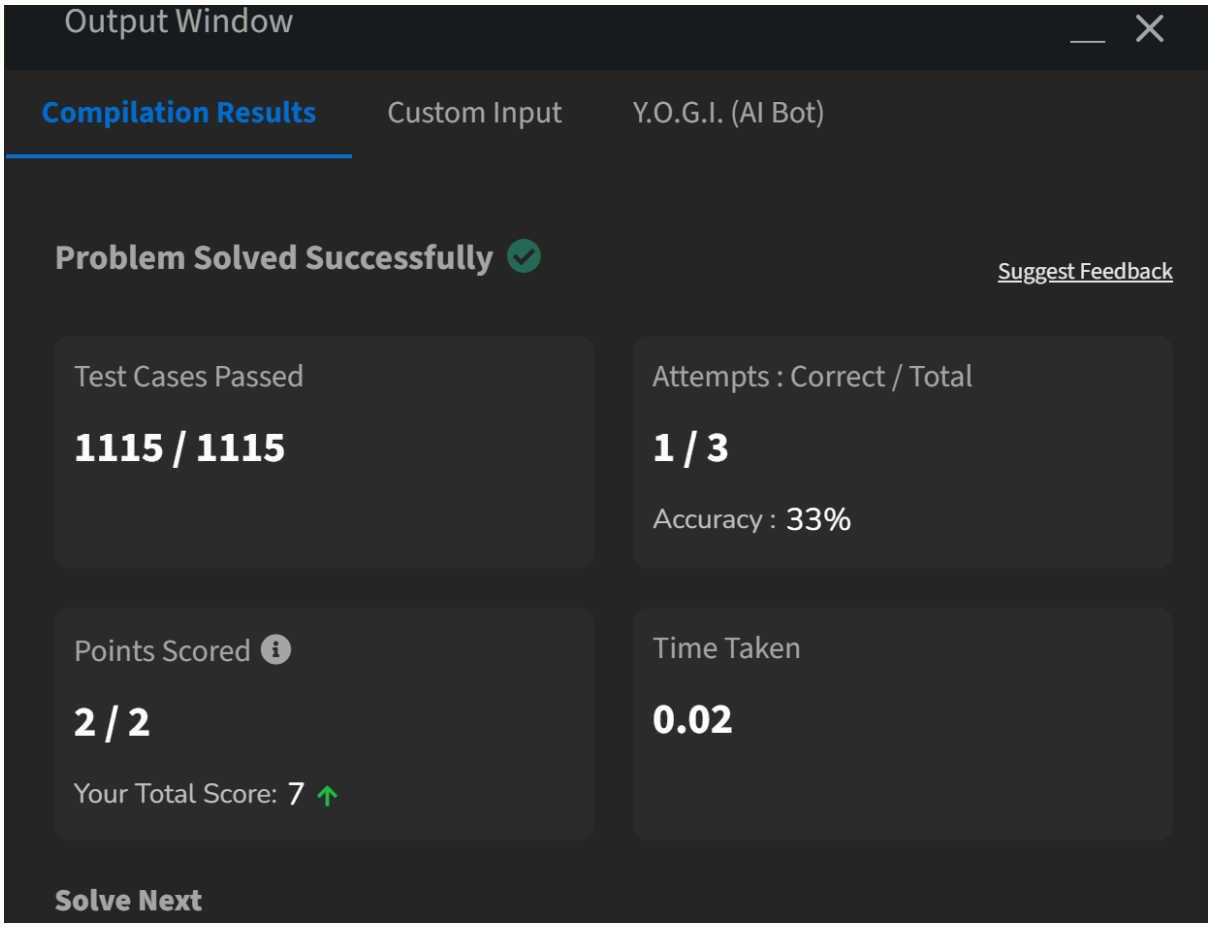


- Code 4:- Check for BST

```
class Solution {
 public:
  long long reversedBits(long long x) {
     long long result = 0;
     for(int i = 0; i < 32; i++) {
        int bit = (x >> i) & 1;
        result |= ((long long)bit << (31 - i));
     }
     return result;
  }
};
```

Qualcomm Interview Coding Questions

Output:-



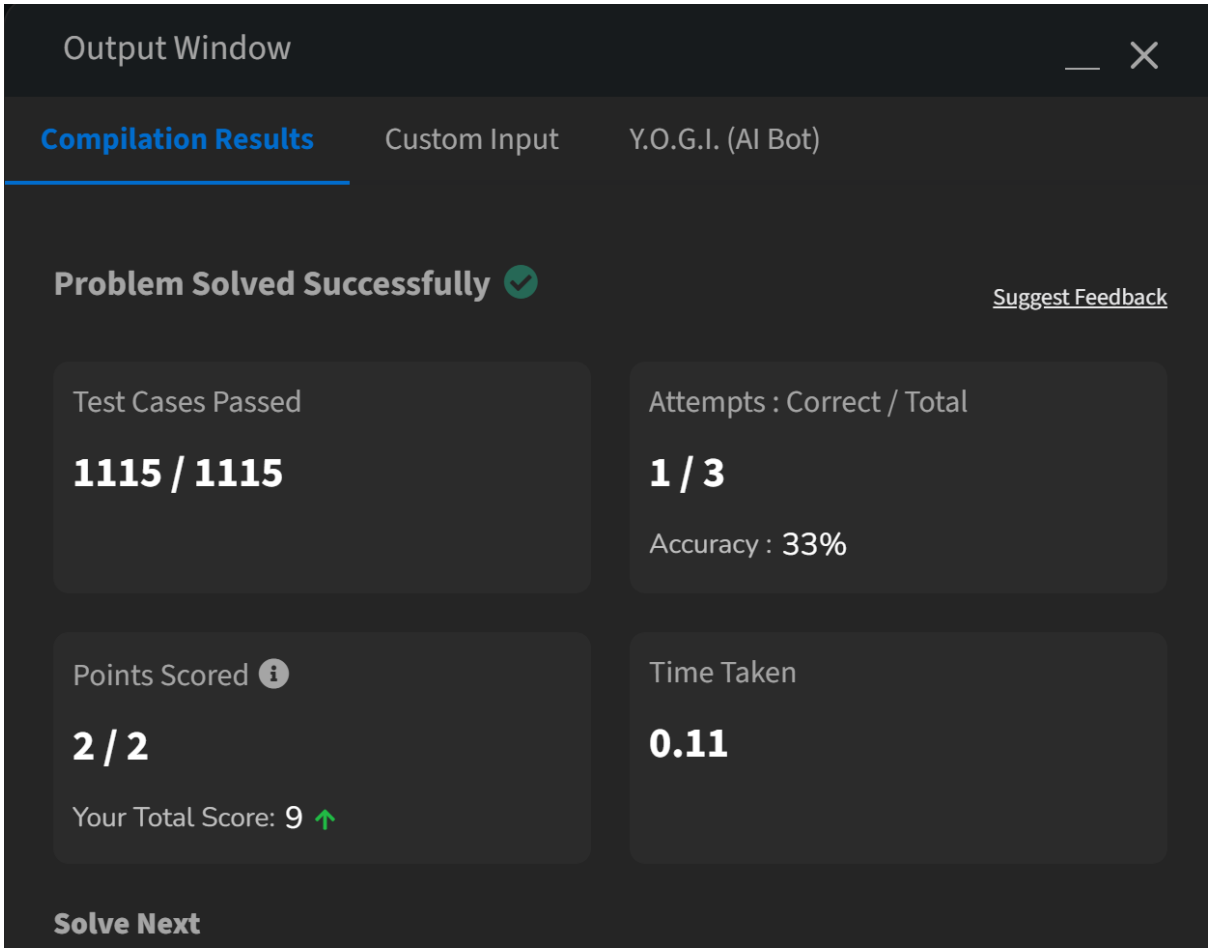
- Code 5:- Reverse Bits

```
class Solution {
 public:
  Node* reverseList(Node* head) {
    Node* prev = NULL;
    Node* curr = head;
    Node* next = NULL;

    while (curr != NULL) {
      next = curr->next;
      curr->next = prev;
```

Qualcomm Interview Coding Questions

```
        prev = curr;
        curr = next;
    }

    return prev;
  }
};
```

Output:-



Qualcomm Interview Coding Questions

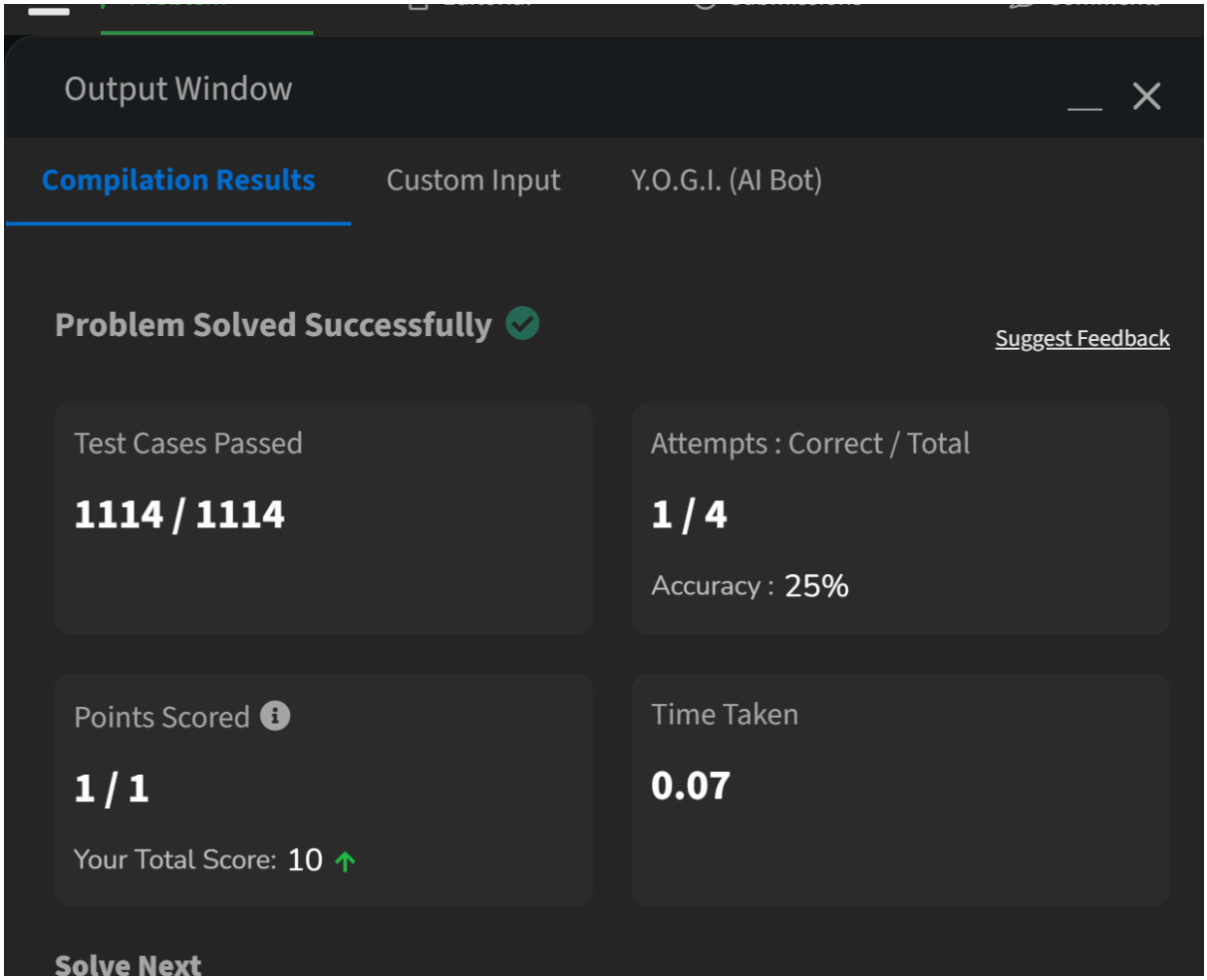
- Code:- 6 Reverse a linked list

```
#include <algorithm>
using namespace std;

class Solution {
 public:
  bool isSubset(vector<int> &a, vector<int> &b) {
    sort(a.begin(), a.end());
    sort(b.begin(), b.end());

    int n = a.size();
    int m = b.size();

    int i = 0;
    int j = 0;

    while (i < n && j < m) {
      if (a[i] == b[j]) {
        i++;
        j++;
      } else if (a[i] < b[j]) {
        i++;
      } else {
        return false;
      }
    }
```

Qualcomm Interview Coding Questions

```
        return (j == m);
    }
};
```

Output:-



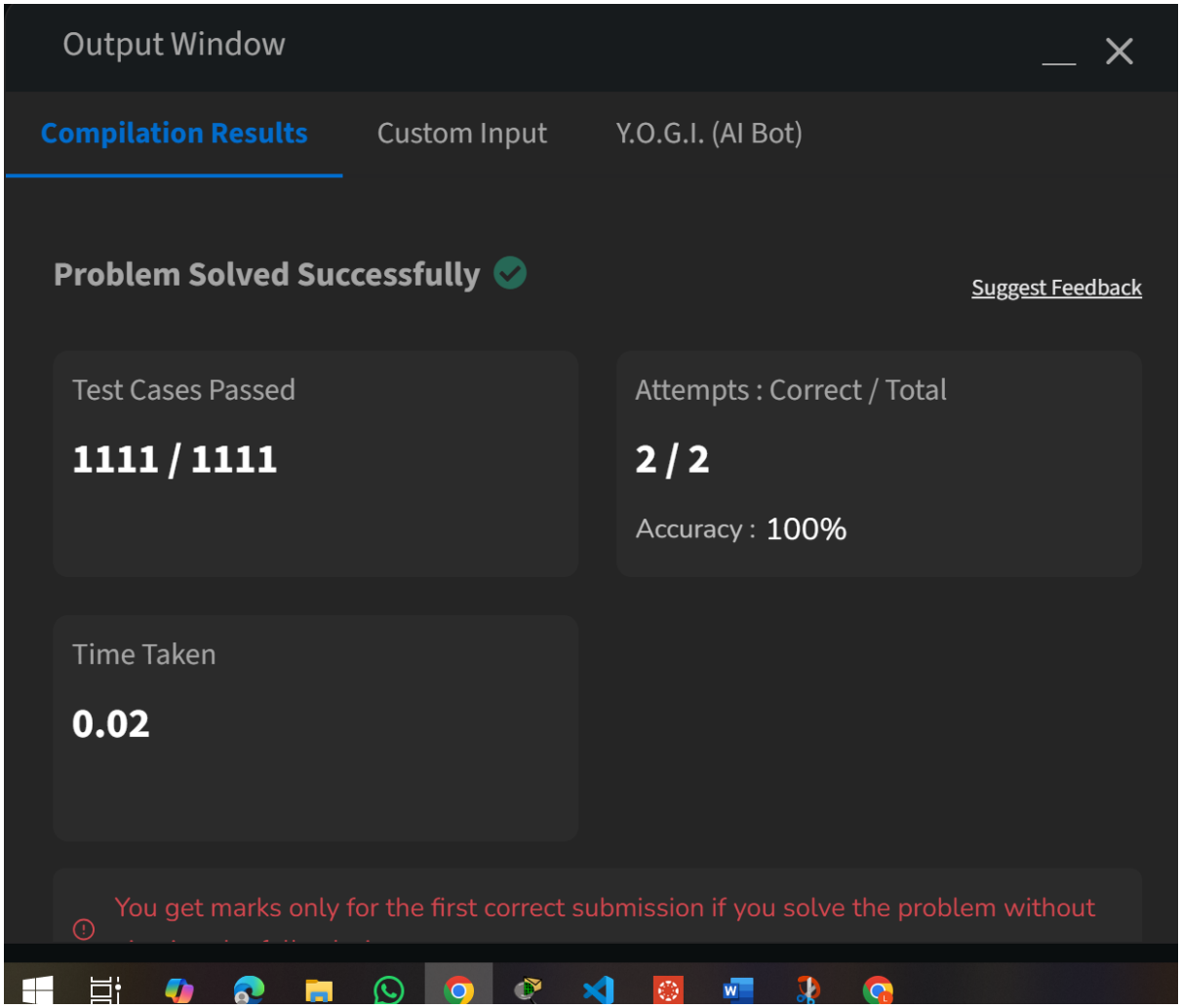
Qualcomm Interview Coding Questions

- Code 7:- Array Subset of another array

```
#include <bitset>
using namespace std;


class Solution {
 public:
  int setBits(int n) {
    int result[32];
    int temp = 0, i = 0, count = 0;



    while (n != 0) {
      temp = n % 2;
      result[i] = temp;
      i++;
      n = n / 2;
    }


    for (int j = 0; j < i; j++) {
      if (result[j] == 1) {
        count++;
      }
    }
    return count;
  }
};
```

Qualcomm Interview Coding Questions

Output:-



- Code 8 :- Set Bits

```
class Solution {
 public:
  int missingNum(vector<int>& arr) {
     int n = arr.size() + 1;
     int xor_all = 0;
     int xor_arr = 0;

     for (int i = 1; i <= n; i++) {
```

Qualcomm Interview Coding Questions

```
        xor_all ^= i;
      }

      for (int x : arr) {
        xor_arr ^= x;
      }

      return xor_all ^ xor_arr;
    }
};
```

Output:-