



Marwari University
Faculty of Technology
Department of Information and Communication Technology

**Subject: Design and Analysis
of Algorithms (01CT0512)**

Name: Nikhil Bhanderi

Class: 5EK1--C

Long Hour Coding

Date: 02-08-2025


Enrollment No: 92301733054

Company Name: Adobe

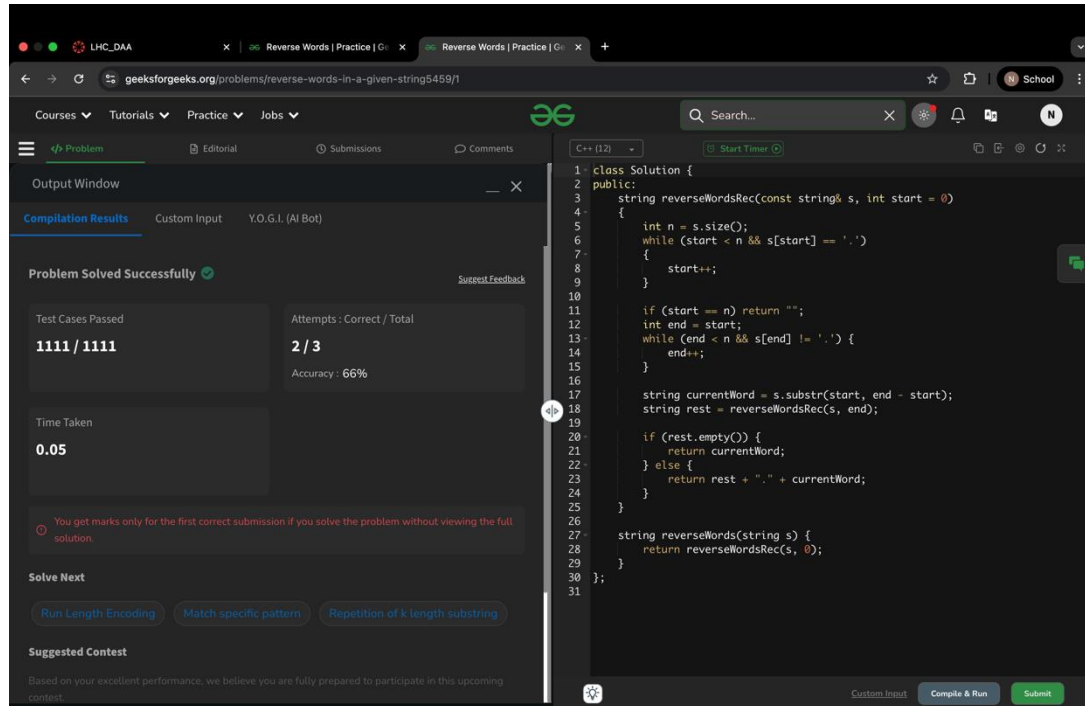
Easy Question 1

Code:

```
class Solution {  
public:  
    string reverseWordsRec(const string& s, int start = 0)  
    {  
        int n = s.size();  
        while (start < n && s[start] == '.')  
        {  
            start++;  
        }  
        if (start == n) return "";  
        int end = start;  
        while (end < n && s[end] != '.') {  
            end++;  
        }  
        string currentWord = s.substr(start, end - start);  
        string rest = reverseWordsRec(s, end);  
  
        if (rest.empty()) {  
            return currentWord;  
        } else {  
            return rest + "." + currentWord;  
        }  
    }  
    string reverseWords(string s) {  
        return reverseWordsRec(s, 0);  
    }  
};
```

 Marwadi University	Marwari University Faculty of Technology Department of Information and Communication Technology	
Subject: Design and Analysis of Algorithms (01CT0512)	Name: Nikhil Bhanderi	Class: 5EK1--C
Long Hour Coding	Date: 02-08-2025	Enrollment No: 92301733054

Output:



The screenshot shows a web browser with the URL <https://www.geeksforgeeks.org/problems/reverse-words-in-a-given-string5459/1>. The page displays the solution for the problem 'Reverse Words in a Given String'. The solution is implemented in C++ using a recursive function. The output window shows 'Problem Solved Successfully' with 11/11 test cases passed, 2/3 attempts correct, and an accuracy of 66%. The time taken is 0.05 seconds.

```

1 class Solution {
2 public:
3     string reverseWordsRec(const string& s, int start = 0)
4     {
5         int n = s.size();
6         while (start < n && s[start] == ' ')
7             start++;
8     }
9
10    if (start == n) return "";
11    int end = start;
12    while (end < n && s[end] != ' ')
13        end++;
14
15    string currentWord = s.substr(start, end - start);
16    string rest = reverseWordsRec(s, end);
17
18    if (rest.empty()) {
19        return currentWord;
20    } else {
21        return rest + " " + currentWord;
22    }
23 }
24
25 string reverseWords(string s) {
26     return reverseWordsRec(s, 0);
27 }
28
29 };

```

Easy Question 2

Code:


```

class Solution {
public:
    bool hasPathSum(Node* root, int targetSum) {
        if (root == nullptr) {
            return false;
        }

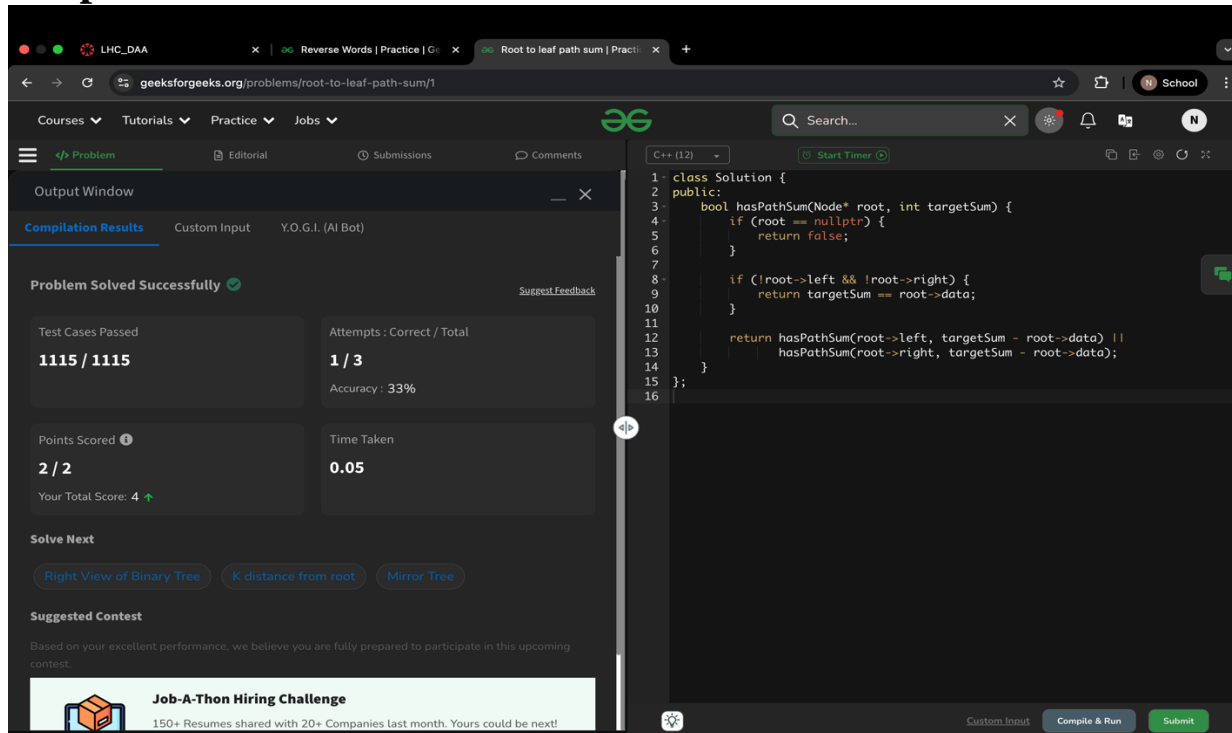
        if (!root->left && !root->right) {
            return targetSum == root->data;
        }

        return hasPathSum(root->left, targetSum - root->data) ||
               hasPathSum(root->right, targetSum - root->data);
    }
};

```

 Marwadi University	Marwari University Faculty of Technology Department of Information and Communication Technology	
Subject: Design and Analysis of Algorithms (01CT0512)	Name: Nikhil Bhanderi	Class: 5EK1--C
Long Hour Coding	Date: 02-08-2025	Enrollment No: 92301733054

Output:




Easy Question 3

Code:

```

class Solution {
public:
    bool isBSTUtil(Node* node, int minValue, int maxValue) {
        if (node == nullptr) {
            return true;
        }
        if (node->data <= minValue || node->data >= maxValue) {
            return false;
        }
        return isBSTUtil(node->left, minValue, node->data) &&
            isBSTUtil(node->right, node->data, maxValue);
    }
    bool isBST(Node* root) {
        const int MIN_VALUE = 0;
        const int MAX_VALUE = 1000000001;
        return isBSTUtil(root, MIN_VALUE, MAX_VALUE);
    }
};

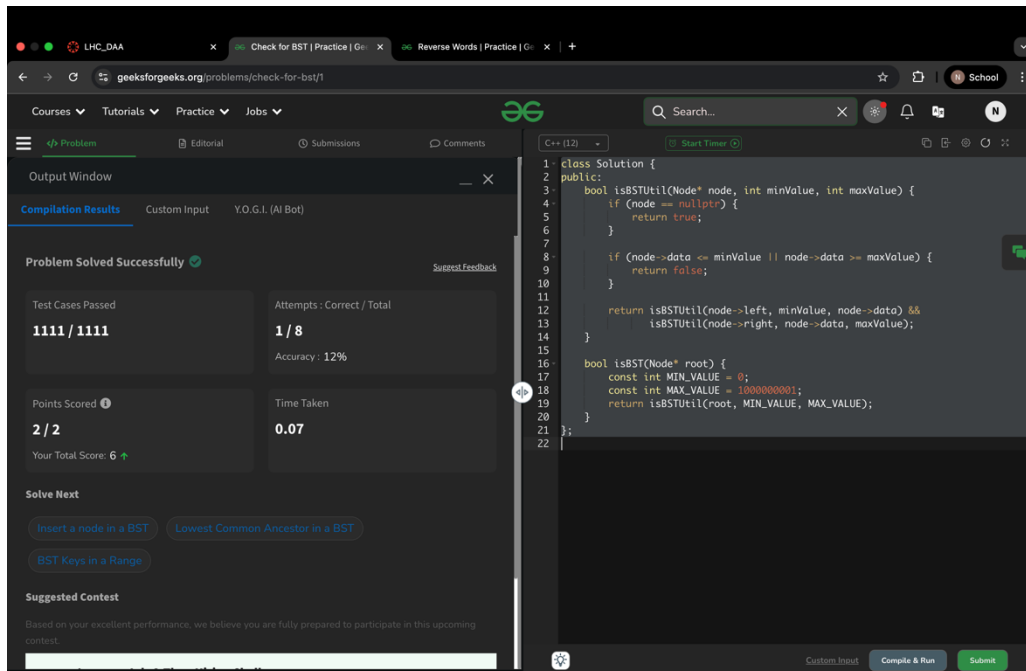
```

 Marwadi University	Marwari University Faculty of Technology Department of Information and Communication Technology	
Subject: Design and Analysis of Algorithms (01CT0512)	Name: Nikhil Bhanderi	Class: 5EK1--C
Long Hour Coding	Date: 02-08-2025	Enrollment No: 92301733054

```

}
};
Output:

```



Easy Question 4


Code:

class Solution

```

{
public:
    int search(vector<int>& arr, int key)
    {
        int low = 0, high = arr.size() - 1;
        while (low <= high)
        {
            int mid = low + (high - low) / 2;
            if (arr[mid] == key)
                return mid;
            if (arr[low] <= arr[mid])
            {
                if (arr[low] <= key && key < arr[mid])

```

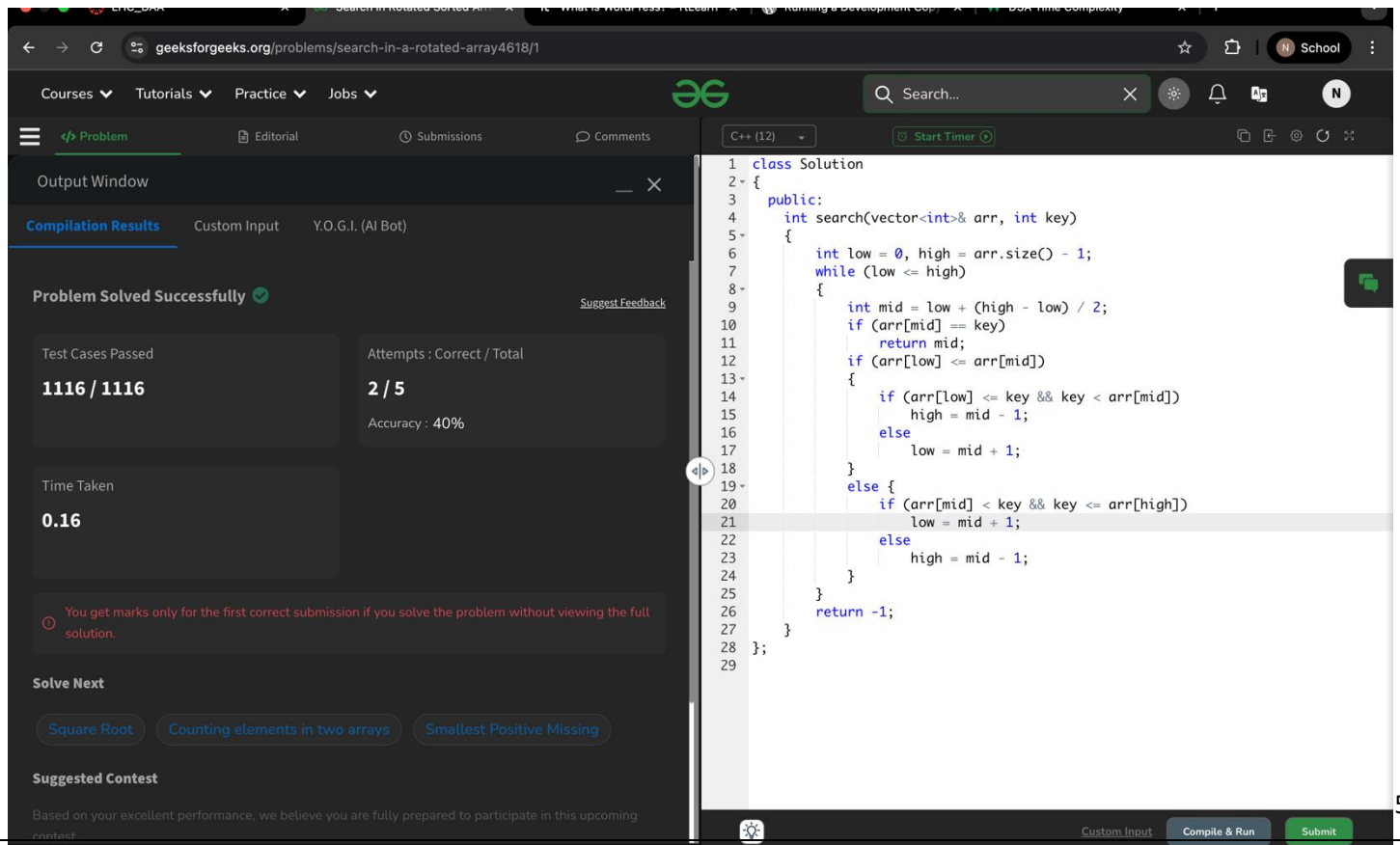
 Marwadi University	Marwari University Faculty of Technology Department of Information and Communication Technology	
Subject: Design and Analysis of Algorithms (01CT0512)	Name: Nikhil Bhanderi	Class: 5EK1--C
Long Hour Coding	Date: 02-08-2025	Enrollment No: 92301733054

```

        high = mid - 1;
    else
        low = mid + 1;
    }
    else {
        if (arr[mid] < key && key <= arr[high])
            low = mid + 1;
        else
            high = mid - 1;
    }
}
return -1;
}
};

```

Output:



The screenshot shows a web browser displaying the GeeksforGeeks problem page for "Search in a rotated array" (ID: 4618/1). The page includes navigation links for Courses, Tutorials, Practice, and Jobs. The problem status is "Problem Solved Successfully" with 1116/1116 test cases passed, 2/5 attempts, and an accuracy of 40%. The time taken is 0.16 seconds. A message states: "You get marks only for the first correct submission if you solve the problem without viewing the full solution." Below this, there are buttons for "Solve Next" and "Suggested Contest".

The code editor on the right shows a C++ solution for the problem. The code defines a class Solution with a public method search that takes a vector of integers and a key. It implements a binary search algorithm to find the key in the rotated array. The code is as follows:

```

1 class Solution
2 {
3 public:
4     int search(vector<int>& arr, int key)
5     {
6         int low = 0, high = arr.size() - 1;
7         while (low <= high)
8         {
9             int mid = low + (high - low) / 2;
10            if (arr[mid] == key)
11                return mid;
12            if (arr[low] <= arr[mid])
13            {
14                if (arr[low] <= key && key < arr[mid])
15                    high = mid - 1;
16                else
17                    low = mid + 1;
18            }
19            else {
20                if (arr[mid] < key && key <= arr[high])
21                    low = mid + 1;
22                else
23                    high = mid - 1;
24            }
25        }
26        return -1;
27    }
28 };

```



Marwari University
Faculty of Technology
Department of Information and Communication Technology

**Subject: Design and Analysis
of Algorithms (01CT0512)**

Name: Nikhil Bhanderi

Class: 5EK1--C

Long Hour Coding


Date: 02-08-2025

Enrollment No: 92301733054

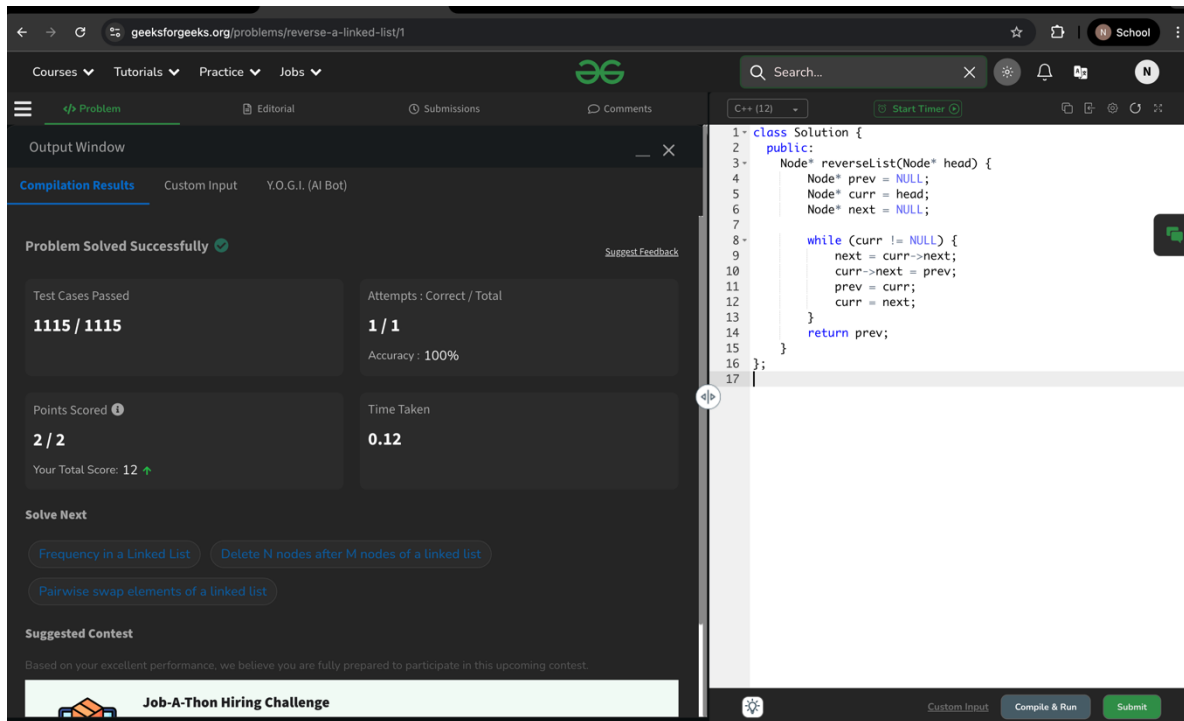
Easy Question 5

Code:

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

 Marwadi University	Marwari University Faculty of Technology Department of Information and Communication Technology	
Subject: Design and Analysis of Algorithms (01CT0512)	Name: Nikhil Bhanderi	Class: 5EK1--C
Long Hour Coding	Date: 02-08-2025	Enrollment No: 92301733054

Output:




The screenshot shows a web-based coding environment. On the left, a sidebar contains navigation links like 'Courses', 'Tutorials', 'Practice', and 'Jobs'. The main area is divided into a 'Problem' section on the left and a code editor on the right. The 'Problem' section displays the status 'Problem Solved Successfully' and various performance metrics: 1115/1115 test cases passed, 1/1 attempts, 100% accuracy, 2/2 points scored, and a time taken of 0.12 seconds. Below these metrics are suggestions for the next problem and a contest recommendation. The code editor on the right shows a C++ solution for reversing a linked list, with line numbers 1 through 17 visible. The code defines a 'Solution' class with a 'reverseList' method that uses a while loop to reverse the linked list by changing the 'next' pointers of each node.

Easy Question 6

Code:

```
class Solution {
public:
    vector<int> findSpiral(Node* root) {
        vector<int> result;
        if (!root) return result;
        stack<Node*> s1;
        stack<Node*> s2;
        s1.push(root);
        while (!s1.empty() || !s2.empty()) {
            while (!s1.empty()) {
                Node* curr = s1.top();
                s1.pop();
                result.push_back(curr->data);
                if (curr->right) s2.push(curr->right);
            }
            while (!s2.empty()) {
                Node* curr = s2.top();
                s2.pop();
                result.push_back(curr->data);
                if (curr->left) s1.push(curr->left);
            }
        }
        return result;
    }
};
```

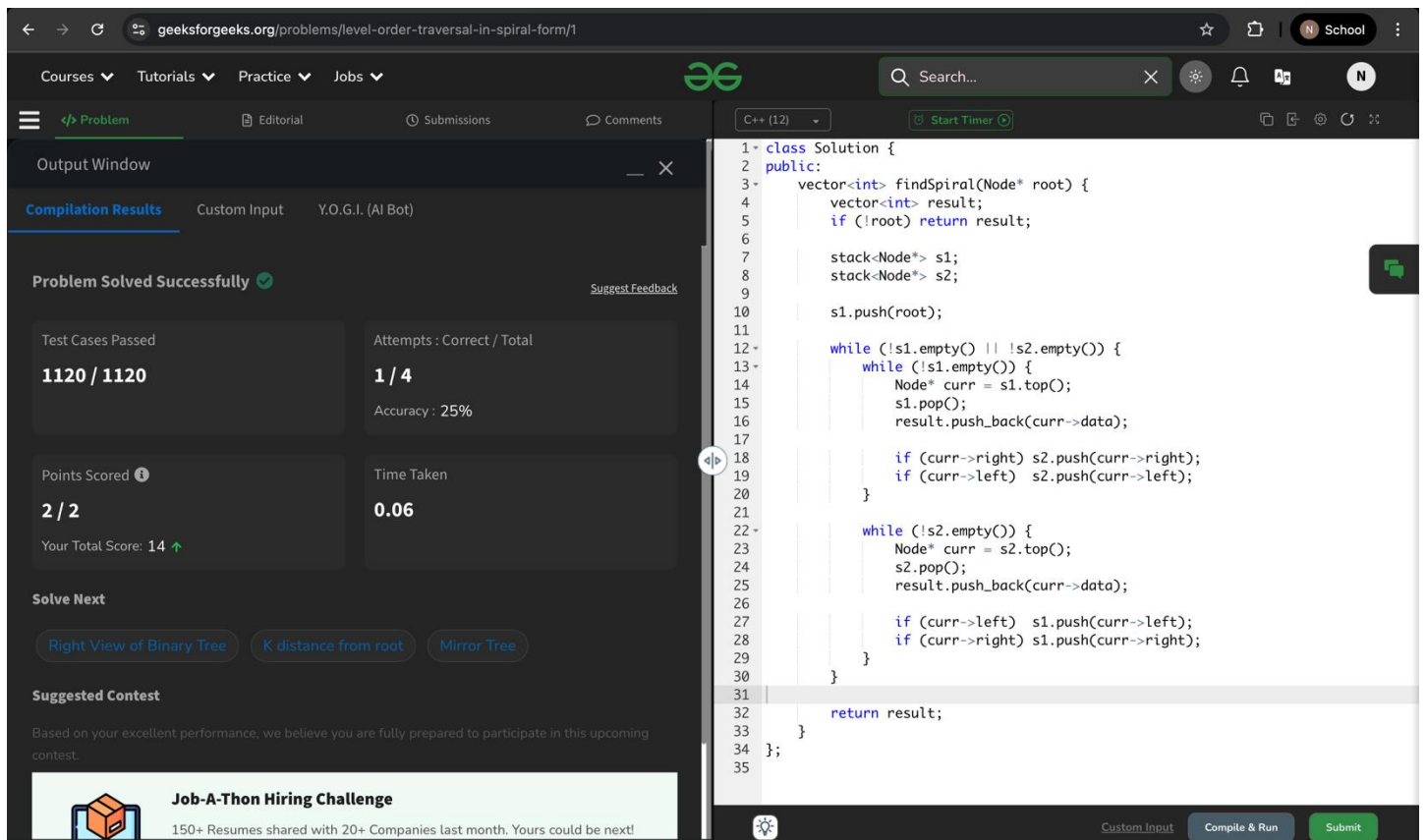
 Marwadi University	Marwari University Faculty of Technology Department of Information and Communication Technology	
Subject: Design and Analysis of Algorithms (01CT0512)	Name: Nikhil Bhanderi	Class: 5EK1--C
Long Hour Coding	Date: 02-08-2025	Enrollment No: 92301733054

```

        if (curr->left) s2.push(curr->left);
    }
    while (!s2.empty()) {
        Node* curr = s2.top();
        s2.pop();
        result.push_back(curr->data);
        if (curr->left) s1.push(curr->left);
        if (curr->right) s1.push(curr->right);
    }
}
return result;
}
};

```

OUTPUT:



The screenshot displays the GeeksforGeeks website interface for a coding problem titled "Level-order-traversal-in-spiral-form/1". The user has successfully solved the problem, as indicated by the "Problem Solved Successfully" message and the green checkmark. The solution is written in C++ and is shown in the editor on the right. The output window on the left shows the following statistics:

- Test Cases Passed: 1120 / 1120
- Attempts: Correct / Total: 1 / 4
- Accuracy: 25%
- Points Scored: 2 / 2
- Time Taken: 0.06
- Your Total Score: 14

The suggested contest is "Job-A-Thon Hiring Challenge", which is described as a contest where 150+ resumes were shared with 20+ companies last month. The user is encouraged to participate in this upcoming contest.



Marwari University
Faculty of Technology
Department of Information and Communication Technology

**Subject: Design and Analysis
of Algorithms (01CT0512)**

Name: Nikhil Bhandari

Class: 5EK1--C

Long Hour Coding

Date: 02-08-2025

Enrollment No: 92301733054