Accenture Sections	Information	Questions and Time
Cognitive Ability	• English Ability • Critical Thinking and Problem Solving • Abstract Reasoning	50 Ques in 50 mins
Technical Assessment	• Common Application and MS Office • Pseudo Code • Fundamental of Networking, Security and Cloud	40 Ques in 40 mins
Coding Round	• C • C++ • Dot Net • JAVA • Python	2 Ques in 45 mins

DEBUG WITH SHUBHAM

Accenture Technical Assessment Detailed Overview

22-SEP-2024 Coding Question


https://www.youtube.com/@DebugWithShubham

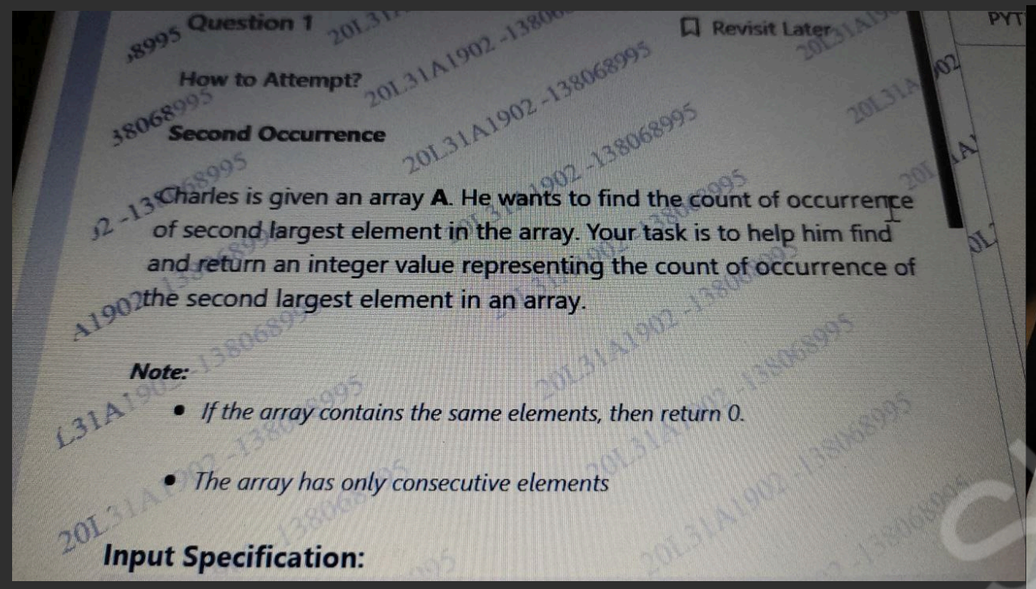

https://www.linkedin.com/in/debugwithshubham/


https://www.instagram.com/debugwithshubham/


https://topmate.io/debugwithshubham


https://t.me/debugwithshubham

Question-1



Question 1

Revisit Later

How to Attempt?

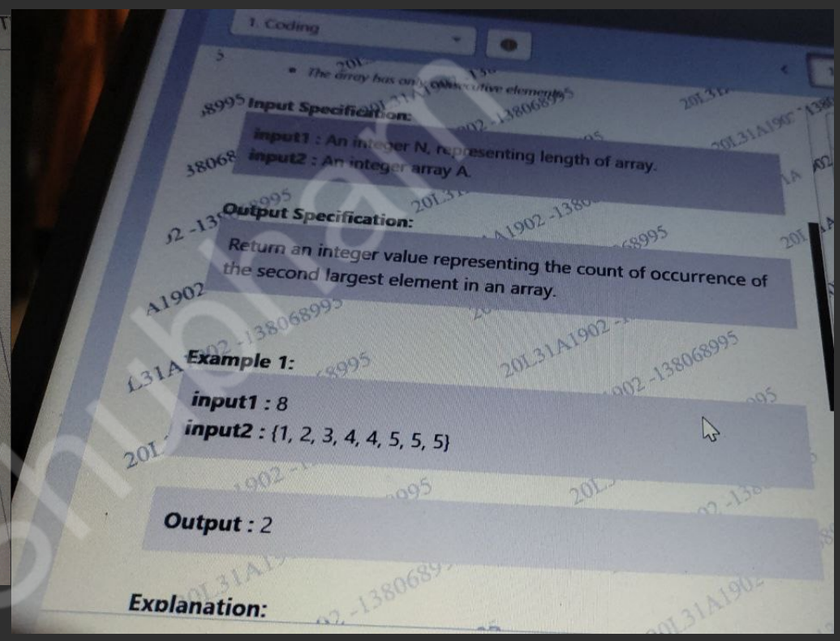
Second Occurrence

Charles is given an array **A**. He wants to find the count of occurrence of second largest element in the array. Your task is to help him find and return an integer value representing the count of occurrence of the second largest element in an array.

Note:

- *If the array contains the same elements, then return 0.*
- *The array has only consecutive elements*

Input Specification:



1. Coding

Input Specification:

input1 : An integer N, representing length of array.

input2 : An integer array A.

Output Specification:

Return an integer value representing the count of occurrence of the second largest element in an array.

Example 1:

input1 : *8*

input2 : *{1, 2, 3, 4, 4, 5, 5, 5}*

Output : *2*

Explanation:

Use a Set for Uniqueness

PYTHON

main.py +

```python
arr = [1, 2, 3, 4, 5, 5, 4]
n = len(arr)
s = list(set(arr))
maxa = s[-2]
cnt = 0
for i in range(n):
    if arr[i] == maxa:
        cnt +=1
print(cnt)
```

JAVA

Main.java

```java
import java.util.*;
public class Main {
    public static void main(String[] args) {
        int[] arr = {1, 2, 3, 4, 5, 5, 4};
        int n = arr.length;
        Set<Integer> set = new HashSet<>();
        for (int num : arr) {
            set.add(num);
        }
        List<Integer> s = new ArrayList<>(set);
        Collections.sort(s);
        int maxa = s.get(s.size() - 2);
        int cnt = 0;
        for (int i = 0; i < n; i++) {
            if (arr[i] == maxa) {
                cnt++;
            }
        }
        System.out.println(cnt);
    }
}

```

C++

main.cpp

```cpp
#include <iostream>
#include <vector>
#include <set>
#include <algorithm>
using namespace std;
int main() {
    vector<int> arr = {1, 2, 3, 4, 5, 5, 4};
    int n = arr.size();
    set<int> s(arr.begin(), arr.end());
    vector<int> sorted(s.begin(), s.end());
    int maxa = sorted[sorted.size() - 2];
    int cnt = 0;
    for (int i = 0; i < n; i++) {
        if (arr[i] == maxa) {
            cnt++;
        }
    }
    cout << cnt << endl;

    return 0;
}

```

Sort the Array

PYTHON

main.py

```python
def count_second_largest(A):
    A_sorted = sorted(A, reverse=True)
    largest = A_sorted[0]
    second_largest = None
    for num in A_sorted:
        if num < largest:
            second_largest = num
            break
    if second_largest is None:
        return 0
    return A.count(second_largest)

A = [1,2,3,4,5,5,4]
output = count_second_largest(A)
print(output)
```

JAVA

Main.java — Share — Run

```java
import java.util.*;

public class Main {
    public static int countSecondLargest(int[] A) {
        Integer[] A_sorted = Arrays.stream(A).boxed().toArray(Integer[]::new);
        Arrays.sort(A_sorted, Collections.reverseOrder());
        int largest = A_sorted[0];
        Integer second_largest = null;
        for (int num : A_sorted) {
            if (num < largest) {
                second_largest = num;
                break;
            }
        }
        if (second_largest == null) {
            return 0;
        }
        int count = 0;
        for (int num : A) {
            if (num == second_largest) {
                count++;
            }
        }
        return count;
    }
    public static void main(String[] args) {
        int[] A = {1, 2, 3, 4, 5, 5, 4};
        int output = countSecondLargest(A);
        System.out.println(output);
    }
}
```

C++

main.cpp

```cpp
#include <iostream>
#include <vector>
#include <algorithm>
using namespace std;
int countSecondLargest(vector<int>& A) {
    sort(A.begin(), A.end(), greater<int>());
    int largest = A[0];
    int second_largest = -1;
    for (int num : A) {
        if (num < largest) {
            second_largest = num;
            break;
        }
    }
    if (second_largest == -1) {
        return 0;
    }
    int count = 0;
    for (int num : A) {
        if (num == second_largest) {
            count++;
        }
    }
    return count;
}
int main() {
    vector<int> A = {1, 2, 3, 4, 5, 5, 4};
    int output = countSecondLargest(A);
    cout << output << endl;
    return 0;
}
```

Single Pass (Optimized Approach)

PYTHON

main.py

```python
def count_second_largest(A):
    largest = second_largest = float('-inf')
    count_second_largest = 0
    for num in A:
        if num > largest:
            second_largest = largest
            largest = num
        elif largest > num > second_largest:
            second_largest = num
    if second_largest == float('-inf'):
        return 0
    count_second_largest = A.count(second_largest)
    return count_second_largest
A = [1, 2, 3, 4, 5, 5, 4]
output = count_second_largest(A)
print(output)

```

JAVA

Main.java

Share

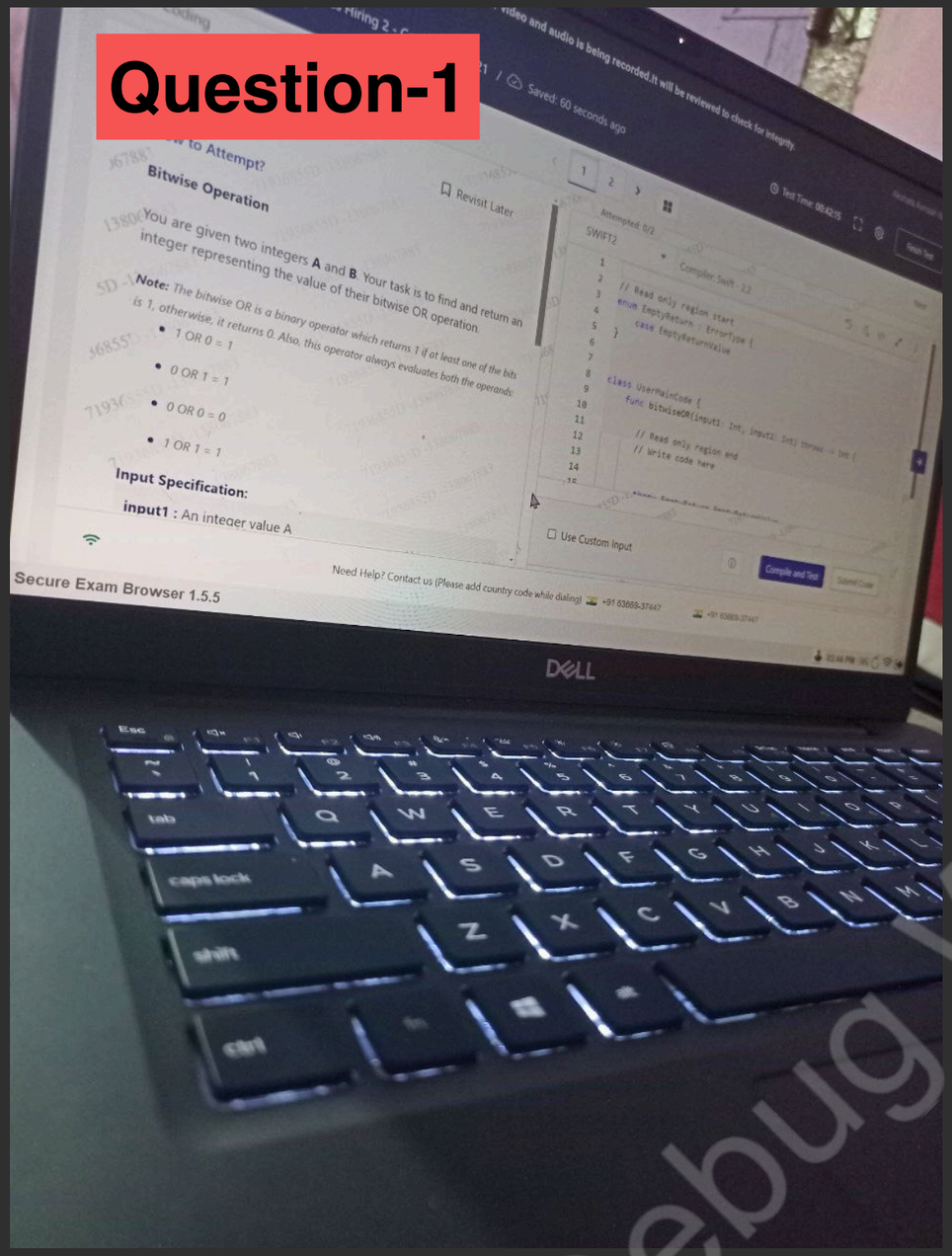
```java
import java.util.*;
public class Main {
    public static int countSecondLargest(int[] A) {
        int largest = Integer.MIN_VALUE;
        int second_largest = Integer.MIN_VALUE;
        int count_second_largest = 0;
        for (int num : A) {
            if (num > largest) {
                second_largest = largest;
                largest = num;
            } else if (num > second_largest && num < largest) {
                second_largest = num;
            }
        }
        if (second_largest == Integer.MIN_VALUE) {
            return 0;
        }
        for (int num : A) {
            if (num == second_largest) {
                count_second_largest++;
            }
        }
        return count_second_largest;
    }
    public static void main(String[] args) {
        int[] A = {1, 2, 3, 4, 5, 5, 4};
        int output = countSecondLargest(A);
        System.out.println(output);
    }
}
```

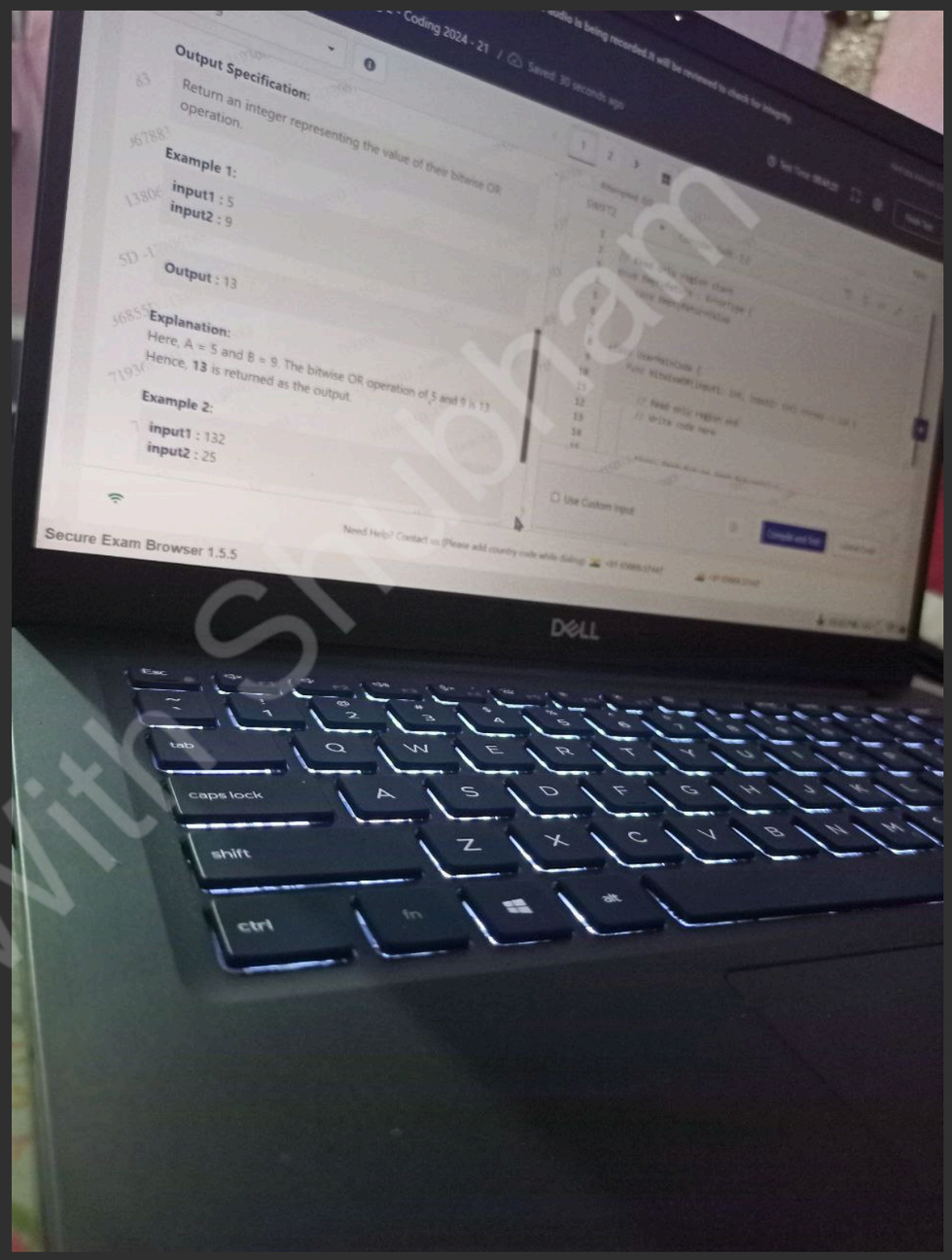
C++

main.cpp

```cpp
#include <iostream>
#include <vector>
#include <limits>
using namespace std;
int countSecondLargest(const vector<int>& A) {
    int largest = numeric_limits<int>::min();
    int second_largest = numeric_limits<int>::min();
    int count_second_largest = 0;
    for (int num : A) {
        if (num > largest) {
            second_largest = largest;
            largest = num;
        } else if (num > second_largest && num < largest) {
            second_largest = num;
        }
    }
    if (second_largest == numeric_limits<int>::min()) {
        return 0;
    }
    for (int num : A) {
        if (num == second_largest) {
            count_second_largest++;
        }
    }
    return count_second_largest;
}
int main() {
    vector<int> A = {1, 2, 3, 4, 5, 5, 4};
    int output = countSecondLargest(A);
    cout << output << endl;
    return 0;
}
```

Question-1





PYTHON

main.py +

```python
def add_using_or(a, b):
    return a | b

a = 5
b = 9
output = add_using_or(a, b)
print(output)

```

JAVA

Main.java

Shar

```java
public class Main {
    public static int addUsingOr(int a, int b) {
        return a | b;
    }
    public static void main(String[] args) {
        int a = 5;
        int b = 9;  /
        int output = addUsingOr(a, b);
        System.out.println(output);
    }
}

```

C++

main.cpp

```cpp
#include <iostream>
using namespace std;
int addUsingOr(int a, int b) {
    return a | b;
}
int main() {
    int a = 5;
    int b = 9;
    int output = addUsingOr(a, b);
    cout << output << endl;

    return 0;
}

```