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

Coding Question


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


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

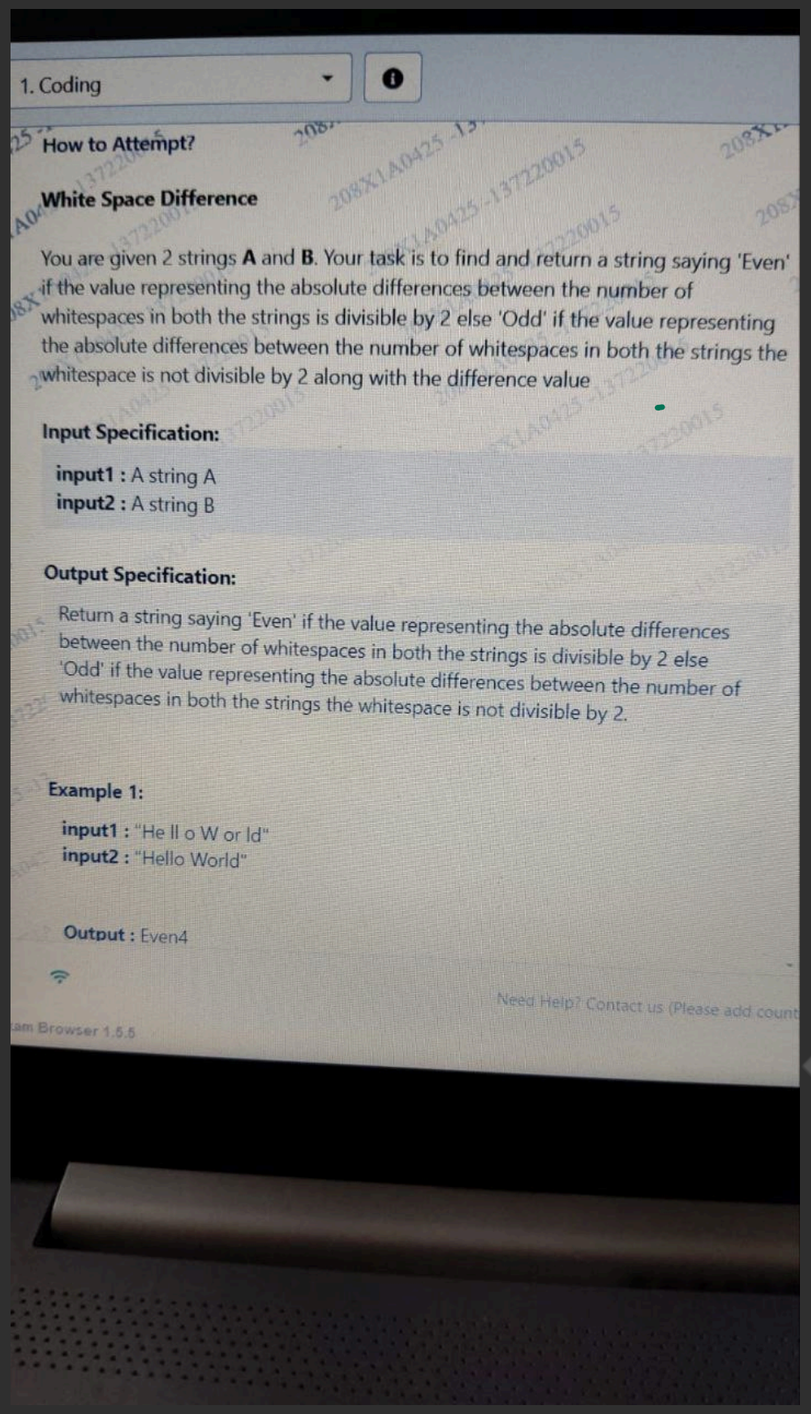

https://www.instagram.com/debugwithshubham/


https://topmate.io/debugwithshubham


https://t.me/debugwithshubham

Question-1

Python



freferer.p

```python
s1 ="He ll o W or id"
s2 = "Hello World"
count_s1  =s1.count(' ')
count_s2 = s2.count(' ')
diff = abs(count_s1 - count_s2)
if diff %2 == 0:
    print(f"Even{diff}")
else:
    print(f"Odd{diff}")
```

freferer.py

```python
s1 ="He ll o W or id"
s2 = "Hello World"
count_s1= 0
count_s2 = 0
for i in s1:
    if i == " ":
        count_s1 += 1
for j in s2:
    if j  == " ":
        count_s2 += 1
diff = abs(count_s1 - count_s2)
if diff %2 == 0:
    print(f"Even{diff}")
else:
    print(f"Odd{diff}")
```

C++

```cpp
#include <iostream>
#include <string>
#include <cmath>  // for abs()

int countSpaces(const std::string& str) {
    int count = 0;
    for (char ch : str) {
        if (ch == ' ') {
            count++;
        }
    }
    return count;
}
int main() {
    std::string s1 = "He ll o W or id";
    std::string s2 = "Hello World";
    int count_s1 = countSpaces(s1);
    int count_s2 = countSpaces(s2);
    int diff = std::abs(count_s1 - count_s2);
    if (diff % 2 == 0) {
        std::cout << "Even" << diff << std::endl;
    } else {
        std::cout << "Odd" << diff << std::endl;
    }

    return 0;
}
```

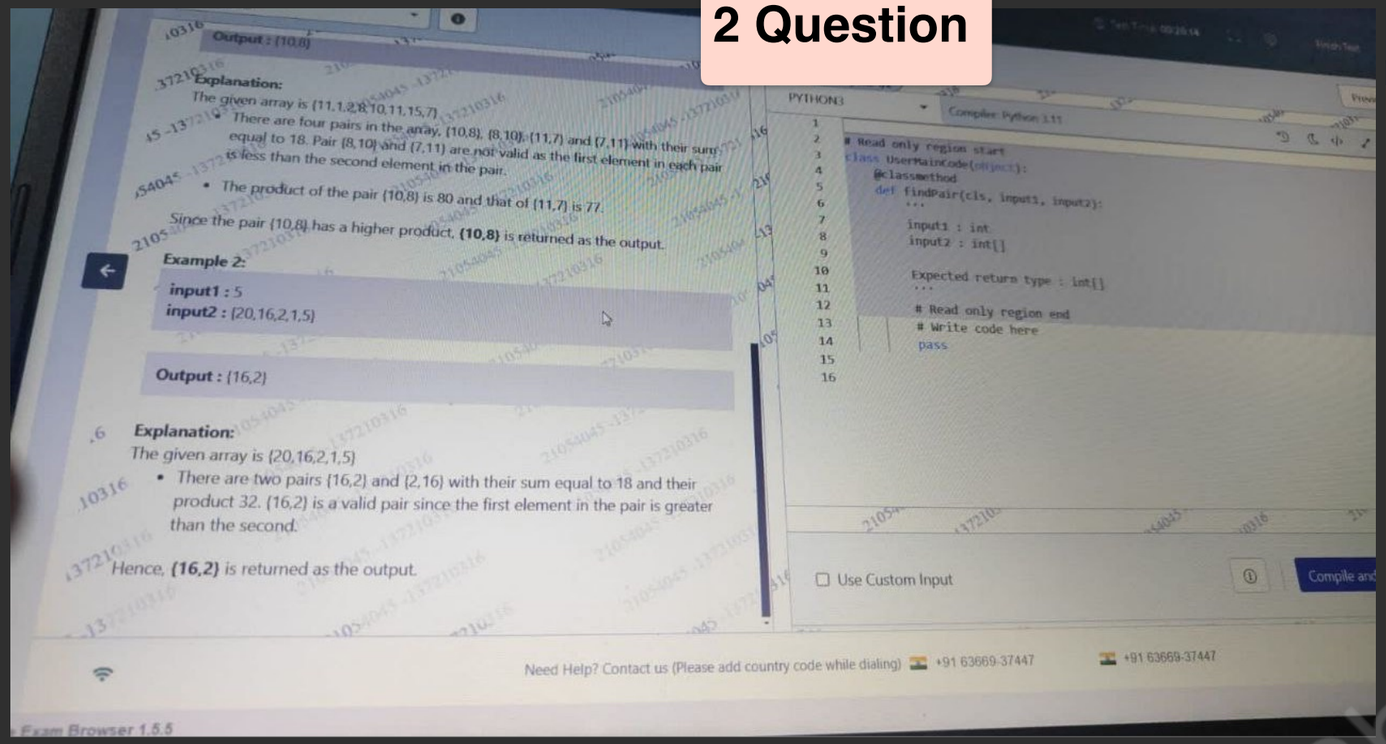
```cpp
#include <iostream>
#include <string>
#include <cmath>  // for abs()

int main() {
    std::string s1 = "He ll o W or id";
    std::string s2 = "Hello World";

    int count_s1 = 0, count_s2 = 0;
    for (char i : s1) {
        if (i == ' ') {
            count_s1++;
        }
    }
    for (char j : s2) {
        if (j == ' ') {
            count_s2++;
        }
    }
    int diff = std::abs(count_s1 - count_s2);
    if (diff % 2 == 0) {
        std::cout << "Even" << diff << std::endl;
    } else {
        std::cout << "Odd" << diff << std::endl;
    }

    return 0;
}
```

Java

```java
public class WhiteSpaceDifference {
    public static int countSpaces(String str) {
        int count = 0;
        for (char ch : str.toCharArray()) {
            if (ch == ' ') {
                count++;
            }
        }
        return count;
    }

    public static void main(String[] args) {
        String s1 = "He ll o W or id";
        String s2 = "Hello World";
        int count_s1 = countSpaces(s1);
        int count_s2 = countSpaces(s2);
        int diff = Math.abs(count_s1 -
count_s2);
        if (diff % 2 == 0) {
            System.out.println("Even" + diff);
        } else {
            System.out.println("Odd" + diff);
        }
    }
}
```

```java
public class WhiteSpaceDifference {

    public static void main(String[] args) {
        String s1 = "He ll o W or id";
        String s2 = "Hello World";

        int count_s1 = 0, count_s2 = 0;
        for (char i : s1.toCharArray()) {
            if (i == ' ') {
                count_s1++;
            }
        }
        for (char j : s2.toCharArray()) {
            if (j == ' ') {
                count_s2++;
            }
        }
        int diff = Math.abs(count_s1 - count_s2);
        if (diff % 2 == 0) {
            System.out.println("Even" + diff);
        } else {
            System.out.println("Odd" + diff);
        }
    }
}
```

2 Question



Python

```python
def find_pair(arr, target_sum):
    max_product = 0
    best_pair = None
    for i in range(len(arr)):
        for j in range(0, len(arr)):
            if arr[i] + arr[j] == target_sum and arr[i] > arr[j]:
                product = arr[i] * arr[j]
                if product > max_product:
                    max_product = product
                    best_pair = (arr[i], arr[j])

    return best_pair

arr = [11,1,2,8,10,11,15,7]
target = 18
print(find_pair(arr,target))
```

```python
def find_pair(arr, target):
    arr.sort(reverse=True)
    l = 0
    h = len(arr)-1
    maxP = 0
    pair = None
    while(l<h):
        curr = arr[l] + arr[h]
        if curr == target and arr[l] > arr[h]:
            pro = arr[l] * arr[h]
            if pro > maxP :
                maxP = pro
                pair = (arr[l],arr[h])
            l += 1
            h -= 1
        elif curr < target:
            h -=1
        else:
            l +=1
    return pair
arr = [11,1,2,8,10,11,15,7]
target = 18
print(find_pair(arr,target))
```

```python
def find_pair(arr, target):
    hashmap = {}
    maxP = 0
    pair = None
    for i in arr:
        comp = target - i
        if comp in hashmap and i > comp:

            pro = i * comp
            if pro > maxP :
                maxP = pro
                pair = (i, comp)

        hashmap[i] = True
    return pair
arr = [11,1,2,8,10,11,15,7]
target = 18
print(find_pair(arr,target))
```

C++

```cpp
#include <iostream>
#include <vector>

std::pair<int, int> findPair(const std::vector<int>& arr, int target_sum) {
    int max_product = 0;
    std::pair<int, int> best_pair = {-1, -1};
    for (size_t i = 0; i < arr.size(); ++i) {
        for (size_t j = 0; j < arr.size(); ++j) {
            if (arr[i] + arr[j] == target_sum && arr[i] > arr[j]) {
                int product = arr[i] * arr[j];
                if (product > max_product) {
                    max_product = product;
                    best_pair = {arr[i], arr[j]};
                }
            }
        }
    }

    return best_pair;
}

int main() {
    std::vector<int> arr = {11, 1, 2, 8, 10, 11, 15, 7};
    int target = 18;

    std::pair<int, int> result = findPair(arr, target);

    if (result.first != -1 && result.second != -1) {
        std::cout << "Pair: (" << result.first << ", " << result.second << ")" << std::endl;
    } else {
        std::cout << "No valid pair found." << std::endl;
    }
    return 0;
}
```

```cpp
#include <iostream>
#include <vector>
#include <algorithm>

using namespace std;

pair<int, int> find_pair(vector<int>& arr, int target_sum) {
    sort(arr.begin(), arr.end(), greater<int>());
    int low = 0, high = arr.size() - 1;
    int max_product = 0;
    pair<int, int> best_pair = {-1, -1};
    while (low < high) {
        int current_sum = arr[low] + arr[high];
        if (current_sum == target_sum && arr[low] > arr[high]) {
            int product = arr[low] * arr[high];
            if (product > max_product) {
                max_product = product;
                best_pair = {arr[low], arr[high]};
            }

            low++;
            high--;
        } else if (current_sum < target_sum) {
            high--;
        } else {
            low++;
        }
    }
    return best_pair;
}
int main() {
    vector<int> arr = {11, 1, 2, 8, 10, 11, 15, 7};
    int target_sum = 18;

    pair<int, int> result = find_pair(arr, target_sum);

    if (result.first != -1 && result.second != -1) {
        cout << "Pair: (" << result.first << ", " << result.second << ")" << endl;
    } else {
        cout << "No valid pair found." << endl;
    }
    return 0;
}
```

```cpp
#include <iostream>
#include <unordered_map>
#include <vector>
using namespace std;
pair<int, int> find_pair(const vector<int>& arr, int target) {
    unordered_map<int, bool> hashmap;
    int maxP = 0;
    pair<int, int> result(-1, -1);
    for (int i : arr) {
        int comp = target - i;
        if (hashmap.find(comp) != hashmap.end() && i > comp) {
            int pro = i * comp;
            if (pro > maxP) {
                maxP = pro;
                result = {i, comp};
            }
        }
        hashmap[i] = true;
    }
    return result;
}
int main() {
    vector<int> arr = {11, 1, 2, 8, 10, 11, 15, 7};
    int target = 18;
    pair<int, int> result = find_pair(arr, target);
    if (result.first != -1 && result.second != -1) {
        cout << "Pair: (" << result.first << ", " << result.second << ")" << endl;
    } else {
        cout << "No valid pair found." << endl;
    }
    return 0;
}
```

Java

```java
import java.util.Arrays;

public class FindPair {

    public static int[] findPair(int[] arr, int target_sum) {
        int maxProduct = 0;
        int[] bestPair = {-1, -1};

        // Iterate over all pairs
        for (int i = 0; i < arr.length; i++) {
            for (int j = 0; j < arr.length; j++) {
                if (arr[i] + arr[j] == target_sum && arr[i] > arr[j]) {
                    int product = arr[i] * arr[j];
                    if (product > maxProduct) {
                        maxProduct = product;
                        bestPair[0] = arr[i];
                        bestPair[1] = arr[j];
                    }
                }
            }
        }

        return bestPair;
    }
    public static void main(String[] args) {
        int[] arr = {11, 1, 2, 8, 10, 11, 15, 7};
        int target_sum = 18;

        int[] result = findPair(arr, target_sum);

        if (result[0]
```

```java
import java.util.Arrays;
import java.util.Comparator;
public class FindPair {
    public static int[] findPair(int[] arr, int target_sum) {
        Arrays.sort(arr);
        reverseArray(arr);
        int low = 0, high = arr.length - 1;
        int maxProduct = 0;
        int[] bestPair = {-1, -1};
        while (low < high) {
            int current_sum = arr[low] + arr[high];
            if (current_sum == target_sum && arr[low] > arr[high]) {
                int product = arr[low] * arr[high];
                if (product > maxProduct) {
                    maxProduct = product;
                    bestPair[0] = arr[low];
                    bestPair[1] = arr[high];
                }
                low++;
                high--;
            } else if (current_sum < target_sum) {
                high--;
            } else {
                low++;
            }
        }
        return bestPair;
    }
    private static void reverseArray(int[] arr) {
        int left = 0, right = arr.length - 1;
        while (left < right) {
            int temp = arr[left];
            arr[left] = arr[right];
            arr[right] = temp;
            left++;
            right--;
        }
    }
    public static void main(String[] args) {
        int[] arr = {11, 1, 2, 8, 10, 11, 15, 7};
        int target_sum = 18;
        int[] result = findPair(arr, target_sum);
        if (result[0] != -1 && result[1] != -1) {
            System.out.println("Pair: (" + result[0] + ", " + result[1] + ")");
        } else {
            System.out.println("No valid pair found.");
        }
    }
}
```

```java
import java.util.HashMap;
public class FindPair {
    public static int[] findPair(int[] arr, int target) {
        HashMap<Integer, Boolean> hashmap = new HashMap<>();
        int maxProduct = 0;
        int[] result = {-1, -1};
        for (int i : arr) {
            int comp = target - i;
            if (hashmap.containsKey(comp) && i > comp) {
                int product = i * comp;
                if (product > maxProduct) {
                    maxProduct = product;
                    result[0] = i;
                    result[1] = comp;
                }
            }
            hashmap.put(i, true);
        }
        return result;
    }
    public static void main(String[] args) {
        int[] arr = {11, 1, 2, 8, 10, 11, 15, 7};
        int target = 18;
        int[] result = findPair(arr, target);
        if (result[0] != -1 && result[1] != -1) {
            System.out.println("Pair: (" + result[0] + ", " + result[1] + ")");
        } else {
            System.out.println("No valid pair found.");
        }
    }
}
```