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

25-sep-2024 Coding Interview Questions


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

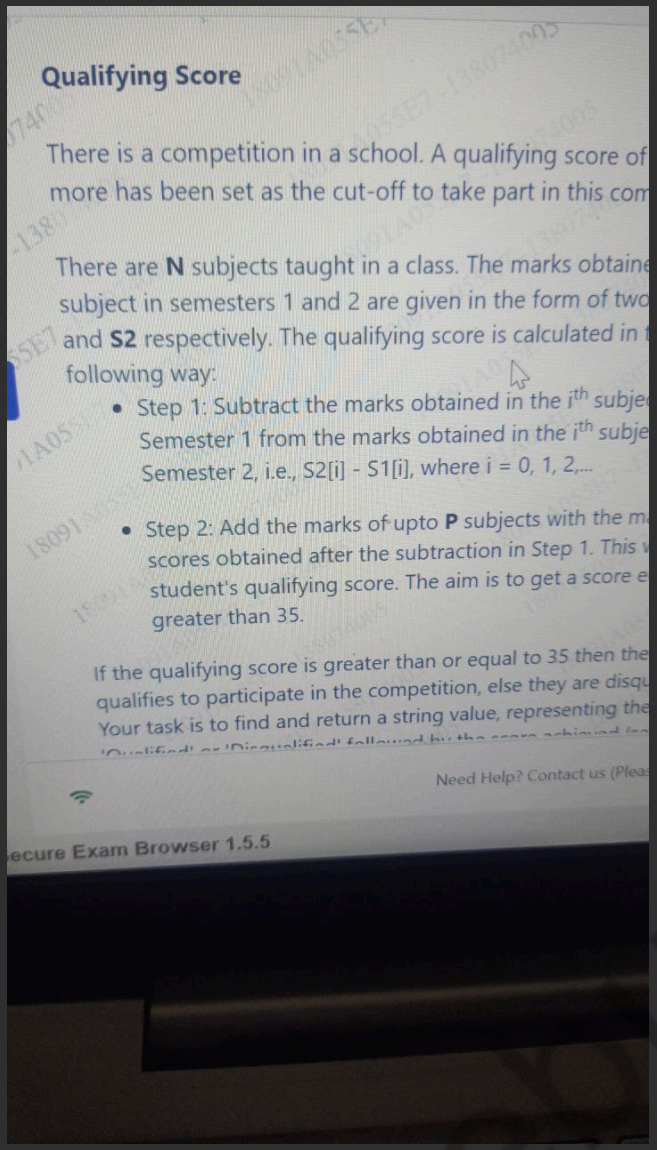

https://www.linkedin.com/in/debugwithshubham/


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


https://topmate.io/debugwithshubham


https://t.me/debugwithshubham

QUESTION-1



Qualifying Score

There is a competition in a school. A qualifying score of
more has been set as the cut-off to take part in this com

There are **N** subjects taught in a class. The marks obtaine
subject in semesters 1 and 2 are given in the form of two
and **S2** respectively. The qualifying score is calculated in t
following way:

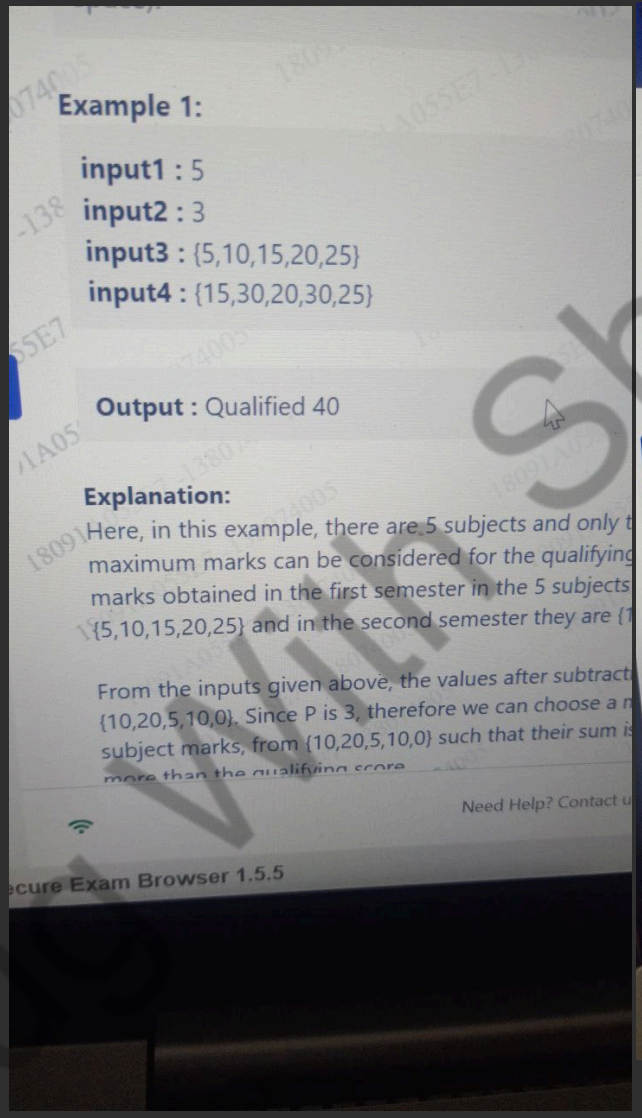
- Step 1: Subtract the marks obtained in the i^{th} subje
Semester 1 from the marks obtained in the i^{th} subje
Semester 2, i.e., S2[i] - S1[i], where i = 0, 1, 2,...

- Step 2: Add the marks of upto **P** subjects with the m
scores obtained after the subtraction in Step 1. This
student's qualifying score. The aim is to get a score e
greater than 35.

If the qualifying score is greater than or equal to 35 then the
qualifies to participate in the competition, else they are disqu
Your task is to find and return a string value, representing the

Need Help? Contact us (Plea

ecure Exam Browser 1.5.5



Example 1:

input1 : 5
input2 : 3
input3 : {5,10,15,20,25}
input4 : {15,30,20,30,25}

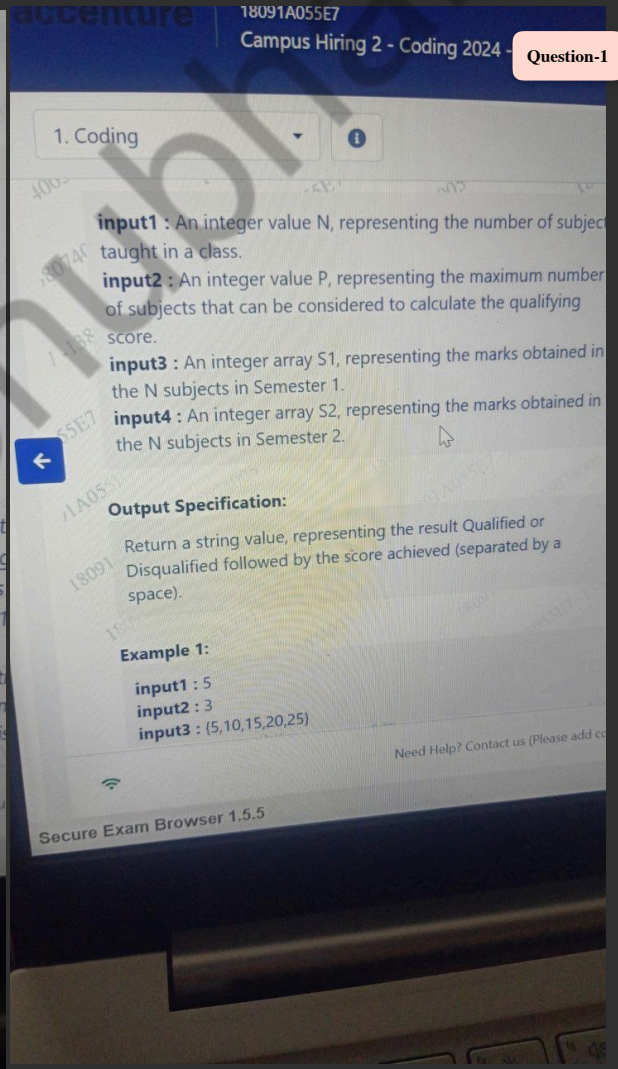
Output : Qualified 40

Explanation:
Here, in this example, there are 5 subjects and only t
maximum marks can be considered for the qualifying
marks obtained in the first semester in the 5 subjects
{5,10,15,20,25} and in the second semester they are {1

From the inputs given above, the values after subtract
{10,20,5,10,0}. Since P is 3, therefore we can choose a n
subject marks, from {10,20,5,10,0} such that their sum is

Need Help? Contact u

ecure Exam Browser 1.5.5



accenture 18091A055E7
Campus Hiring 2 - Coding 2024 -

Question-1

1. Coding

input1 : An integer value N, representing the number of subject
taught in a class.
input2 : An integer value P, representing the maximum number
of subjects that can be considered to calculate the qualifying
score.
input3 : An integer array S1, representing the marks obtained in
the N subjects in Semester 1.
input4 : An integer array S2, representing the marks obtained in
the N subjects in Semester 2.

Output Specification:

Return a string value, representing the result Qualified or
Disqualified followed by the score achieved (separated by a
space).

Example 1:

input1 : 5
input2 : 3
input3 : {5,10,15,20,25}

Need Help? Contact us (Please add c

Secure Exam Browser 1.5.5

input1: 5
input2: 3
input3: [5,10,15,20,25]
input4: [15,30,20,30,25]

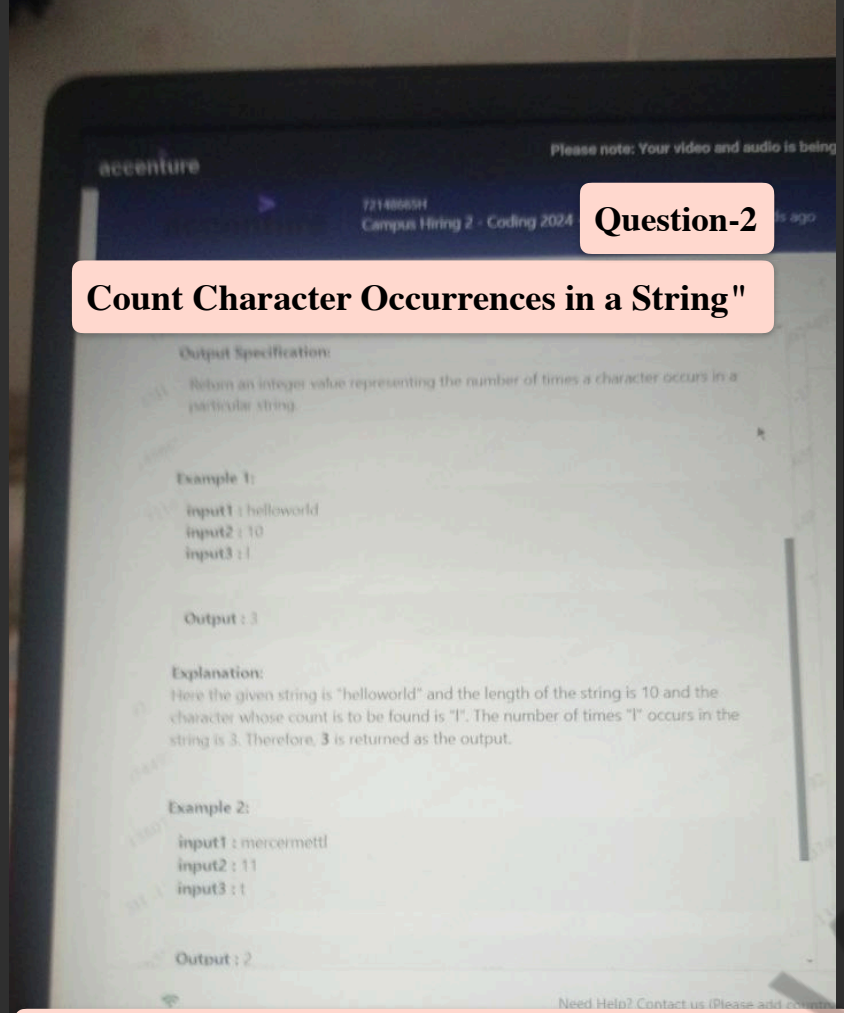
Problem Statement:

You are given a number of subjects N and marks obtained in two semesters, represented by arrays $S1$ (marks in semester 1) and $S2$ (marks in semester 2). To calculate the **qualifying score**, you need to:

1. For each subject, subtract the marks obtained in Semester 1 from the marks obtained in Semester 2: $S2[i] - S1[i]$, for $i = 0, 1, 2, ..., N - 1$.
2. Pick the top P subjects that have the maximum differences (from the above calculation).
3. Sum the differences of these top P subjects. This will be the student's **qualifying score**.
4. If the qualifying score is greater than or equal to 35, the student qualifies for the competition; otherwise, they are disqualified.

Input:

- **Input 1**: An integer N representing the number of subjects.
- **Input 2**: An integer P representing the maximum number of subjects that can be considered to calculate the qualifying score.
- **Input 3**: An integer array $S1$, representing the marks obtained in the N subjects in Semester 1.
- **Input 4**: An integer array $S2$, representing the marks obtained in the N subjects in Semester 2.



Input:
input1: “mercermettl:
input2: 11
input3: “t”

Output:
2

Explanation:
The string is "mercermettl" with a length of 11, and the character ’t” occurs 2 times in the string.

Problem Statement:

You are given a string, the length of the string, and a character. Your task is to determine how many times the given character occurs in the string.

Write a function that returns an integer representing the number of times the specified character appears in the string.

Input:

- **input1:** A string `s` of length `n` (1 ≤ n ≤ 10^5).
- **input2:** An integer `n`, representing the length of the string `s`.
- **input3:** A single character `c` that needs to be counted in the string.

Output:

- Return an integer representing the number of occurrences of the character `c` in the string `s`.

Solution of the First Problem

Python

main.py

```python
def greedy_solution(N, P, S1, S2):
    differences = [S2[i] - S1[i] for i in range(N)]
    differences.sort(reverse=True)
    qualifying_score = sum(differences[:P])
    if qualifying_score >= 35:
        return f"Qualified {qualifying_score}"
    else:
        return f"Disqualified {qualifying_score}"
N = 5
P = 3
S1 = [5, 10, 15, 20, 25]
S2 = [15, 30, 20, 30, 25]

print(greedy_solution(N, P, S1, S2))
```

C++

main.cpp

Share Run

```cpp
#include <iostream>
#include <vector>
#include <algorithm>
using namespace std;
string greedySolution(int N, int P, vector<int> &S1, vector<int> &S2) {
    vector<int> differences(N);
    for (int i = 0; i < N; i++) {
        differences[i] = S2[i] - S1[i];
    }
    sort(differences.begin(), differences.end(), greater<int>());
    int qualifyingScore = 0;
    for (int i = 0; i < P; i++) {
        qualifyingScore += differences[i];
    }
    if (qualifyingScore >= 35) {
        return "Qualified " + to_string(qualifyingScore);
    } else {
        return "Disqualified " + to_string(qualifyingScore);
    }
}
int main() {
    int N = 5;
    int P = 3;
    vector<int> S1 = {5, 10, 15, 20, 25};
    vector<int> S2 = {15, 30, 20, 30, 25};

    cout << greedySolution(N, P, S1, S2) << endl;
    return 0;
}
```


Java

Main.java

Share Run

```java
import java.util.Arrays;
public class Main {
    public static String greedySolution(int N, int P, int[] S1, int[] S2) {
        int[] differences = new int[N];
        for (int i = 0; i < N; i++) {
            differences[i] = S2[i] - S1[i];
        }
        Arrays.sort(differences);
        int qualifyingScore = 0;
        for (int i = N - 1; i >= N - P; i--) {
            qualifyingScore += differences[i];
        }
        if (qualifyingScore >= 35) {
            return "Qualified " + qualifyingScore;
        } else {
            return "Disqualified " + qualifyingScore;
        }
    }
    public static void main(String[] args) {
        int N = 5;
        int P = 3;
        int[] S1 = {5, 10, 15, 20, 25};
        int[] S2 = {15, 30, 20, 30, 25};

        System.out.println(greedySolution(N, P, S1, S2));
    }
}
```

Solution of the Second Problem

Using Count method

Python

main.py +

```python
def count_occurrences(string, length, char_to_find):
    if len(string) != length:
        return "Input length does not match the actual string length"
    count = string.count(char_to_find)
    return count

# Example 1
input1 = "helloworld"
input2 = 10
input3 = "l"
output = count_occurrences(input1, input2, input3)
print(f"Output: {output}")

# Example 2
input1 = "mercermettl"
input2 = 11
input3 = "t"
output = count_occurrences(input1, input2, input3)
print(f"Output: {output}")

```

Solution of the Second Problem

Python

main.py

```python
def count_occurrences(string, length, char_to_find):
    if len(string) != length:
        return "Input length does not match the actual string length"
    count = 0
    for char in string:
        if char == char_to_find:
            count += 1
    return count
input1 = "helloworld"
input2 = 10
input3 = "l"
output = count_occurrences(input1, input2, input3)
print(f"Output: {output}")
```

Java

Main.java

```java
public class Main {
    public static String countOccurrences(String string, int length, char
        charToFind) {
        if (string.length() != length) {
            return "Input length does not match the actual string length";
        }
        int count = 0;
        for (int i = 0; i < string.length(); i++) {
            if (string.charAt(i) == charToFind) {
                count++;
            }
        }
        return String.valueOf(count);
    }
    public static void main(String[] args) {
        String input1 = "helloworld";
        int input2 = 10;
        char input3 = 'l';
        String output = countOccurrences(input1, input2, input3);
        System.out.println("Output: " + output);
    }
}
```

C++

main.cpp

```cpp
#include <iostream>
#include <string>

using namespace std;

string countOccurrences(const string& str, int length, char charToFind) {
    if (str.length() != length) {
        return "Input length does not match the actual string length";
    }
    int count = 0;
    for (char ch : str) {
        if (ch == charToFind) {
            count++;
        }
    }
    return to_string(count);
}
int main() {
    string input1 = "helloworld";
    int input2 = 10;
    char input3 = 'l';
    string output = countOccurrences(input1, input2, input3);
    cout << "Output: " << output << endl;
    return 0;
}
```