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

Chocolate Distribution Problem

Given an array of N integers where each value represents the number of chocolates in a packet. Each packet can have a variable number of chocolates. There are m students, the task is to distribute chocolate packets such that:

- **Each student gets one packet.**
- **The difference between the number of chocolates in the packet with maximum chocolates and the packet with minimum chocolates given to the students is minimum.**

Input : arr[] = {7, 3, 2, 4, 9, 12, 56} , m = 3
Output: Minimum Difference is 2

Input : arr[] = {3, 4, 1, 9, 56, 7, 9, 12} , m = 5
Output: Minimum Difference is 6

C++

```cpp
#include <bits/stdc++.h>
using namespace std;
int findMinDiff(int arr[], int n, int m)
{
    if (m == 0 || n == 0)
        return 0;
    sort(arr, arr + n);
    if (n < m)
        return -1;
    int min_diff = INT_MAX;
    for (int i = 0; i + m - 1 < n; i++) {
        int diff = arr[i + m - 1] - arr[i];
        if (diff < min_diff)
            min_diff = diff;
    }
    return min_diff;
}
int main()
{
    int arr[] = { 12, 4,  7,  9,  2,  23, 25, 41, 30,
                  40, 28, 42, 30, 44, 48, 43, 50 };
    int m = 7;
    int n = sizeof(arr) / sizeof(arr[0]);
    cout << "Minimum difference is "
         << findMinDiff(arr, n, m);
    return 0;
}
```

JAVA

```java
import java.util.Arrays;
public class ChocolateDistribution {
    public static int chocolateDistribution(int arr[], int m) {
        if (arr.length == 0 || m == 0) {
            return 0;
        }
        Arrays.sort(arr);
        if (arr.length - 1 < m) {
            return -1;
        }
        int min_diff = Integer.MAX_VALUE;
        for (int i = 0; i < arr.length; i++) {
            int nextWindow = i + m - 1;
            if (nextWindow >= arr.length)
                break;
            int diff = arr[nextWindow] - arr[i];
            min_diff = Math.min(min_diff, diff);
        }
        return min_diff;
    }

    public static void main(String[] args) {
        int arr[] = {12, 4, 7, 9, 2, 23, 25, 41, 30, 40, 28, 42, 30, 44, 48, 43, 50};
        int m = 7;
        int result = chocolateDistribution(arr, m);
        if (result != -1) {
            System.out.println("Minimum difference is " + result);
        } else {
            System.out.println("Invalid input");
        }
    }
}
```

Python

```python
def Chocolate(arr, m, n):
    if (m==0 or n ==0 ):
        return 0
    if(n<m):
        return -1
    arr.sort()
    mina = arr[n-1] - arr[0]
    for i in range(n-m+1):
        mina = min(mina,arr[i+m-1] - arr[i])
    return mina

arr = list(map(int,input().split()))
m = int(input())
n = len(arr)
print(Chocolate(arr,m,n))
```

Given two strings s1 and s2 consisting of lowercase characters, the task is to check whether the two given strings are anagrams of each other or not. An anagram of a string is another string that contains the same characters, only the order of characters can be different. For example, “act” and “tac” are anagrams of each other.

Input:
Input 1: 1st string
Input 2: 2nd string
Output:
(If they are anagrams, the function will return ‘yes’. Otherwise, it will return ‘no’.)
Example
Input 1: listen
Input 2: silent
Output:
Yes
Explanation
Listen and Silent are anagrams (an anagram is a word formed by rearranging the letters of the other word).

C++

```cpp
#include <algorithm>
#include <iostream>
using namespace std;
bool areAnagrams(string s1, string s2)
{
    sort(s1.begin(), s1.end());
    sort(s2.begin(), s2.end());
    return s1 == s2;
}

int main()
{
    string str1 = "abcd";
    string str2 = "adce";

    if (areAnagrams(str1, str2)) {
        cout << "True" << endl;
    }
    else {
        cout << "False" << endl;
    }

    return 0;
}
```

JAVA

```java
import java.util.Arrays;

public class AnagramChecker {
    public static boolean areAnagrams(String s1, String s2)
    {
        char[] charArray1 = s1.toCharArray();
        char[] charArray2 = s2.toCharArray();
        Arrays.sort(charArray1);
        Arrays.sort(charArray2);
        return Arrays.equals(charArray1, charArray2);
    }

    public static void main(String[] args)
    {
        String str1 = "abcd";
        String str2 = "adce";

        if (areAnagrams(str1, str2)) {
            System.out.println("True");
        }
        else {
            System.out.println("False");
        }
    }
}
```

Python

```python
def anagrams(s1, s2):
    if len(s1) != len(s2):
        return False
    count = [0]*26

    for i in s1:
        count[ord(i)-ord('a')] += 1
    for i in s2:
        count[ord(i)-ord('a')] -= 1
    for cnt in count:
        if cnt != 0:
            return False
    return True
s1 = "abcd"
s2= "dacb"
print(anagrams(s1,s2))
```