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

Coding Question 1

The function def differenceofSum(n. m) accepts two integers n, m as arguments Find the sum of all numbers in range from 1 to m(both inclusive) that are not divisible by n. Return difference between sum of integers not divisible by n with sum of numbers divisible by n.

<u>Assumption:</u>

- n>0 and m>0
- Sum lies between integral range

<u>Example</u>

Input
n:4
m:20
Output
90

<u>Explanation</u>

- Sum of numbers divisible by 4 are 4 + 8 + 12 + 16 + 20 = 60
- Sum of numbers not divisible by 4 are 1 +2 + 3 + 5 + 6 + 7 + 9 + 10 + 11 + 13 + 14 + 15 + 17 + 18 + 19 = 150
- Difference 150 – 60 = 90

Sample Input
n:3
m:10
Sample Output
19

Python

```python
def differenceofSum(n,m):
    sum_div= 0
    sum_not_div= 0
    for i in range(1, m+1):
        if i % n == 0:
            sum_div += i
        else:
            sum_not_div += i

    return abs(sum_not_div - sum_div)

n = int(input())
m = int(input())
print(differenceofSum(n,m))
```

JAVA

```java
import java.util.*;
class Solution
{
    public static int differenceOfSum (int m, int n)
    {
        int sum1 = 0, sum2 = 0;
        for (int i = 1; i <= m; i++)
        {
            if (i % n == 0)
                sum1 = sum1 + i;
            else
                sum2 = sum2 + i;
        }
        return Math.abs (sum1 - sum2);
    }
public static void main (String[]args)
{
Scanner sc = new Scanner (System.in);
int n = sc.nextInt ();
int m = sc.nextInt ();
System.out.println (differenceOfSum (m, n));
}
}
```

C++

```cpp
#includ<bits/stdc++.h>
using namespace std;
int differenceofSum(int n, int m)
{
int i, sum1 = 0, sum2 = 0;
for(i=1; i<=m; i++)
{
if(i%n==0)
{
sum1 = sum1 + i;
}
else
{
sum2 = sum2 + i;
}
}
return sum2 - sum1;
}

int main()
{
int n, m;
int result;
cin>>n>>m;
result = differenceofSum(n, m);
cout<<result;
return 0;
}
```

Coding Question 2

You are required to implement the following Function def LargeSmallSum(arr).

The function accepts an integers arr of size ’length’ as its arguments you are required to return the sum of second largest largest element from the even positions and second smallest from the odd position of given ‘arr’.

Assumption:

- All array elements are unique
- Treat the 0th position a seven

NOTE

- Return 0 if array is empty
- Return 0, if array length is 3 or less than 3

Example:-

Input

arr:3 2 1 7 5 4

Output

7

Explanation

- Second largest among even position elements(1 3 5) is 3
- Second largest among odd position element is 4
- Thus output is 3+4 = 7

Sample Input

arr:1 8 0 2 3 5 6

Sample Output

8

freferer.py - /Users/shubhammaurya/Documents/freferer.py (3.10.10)

Python

```python
def LargeSmallSum(arr,n):
    arr.sort()
    even_arr =[]
    odd_arr = []
    for i in range(n):
        if i % 2 == 0:
            even_arr.append(arr[i])
        else:
            odd_arr.append(arr[i])
    return even_arr[len(even_arr)-2] + odd_arr[len(odd_arr)-2]

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

JAVA

```java
import java.util.*;
class Solution
{
    public static int largeSmallSum (int[]arr)
    {
        ArrayList < Integer > even = new ArrayList < Integer > ();
        ArrayList < Integer > odd = new ArrayList < Integer > ();
        even.add (arr[0]);

        for (int i = 1; i < arr.length; i++)
        {
            if (i % 2 == 0)
                even.add (arr[i]);
            else
                odd.add (arr[i]);
        }
        Collections.sort (even);
        Collections.sort (odd);

        return even.get (even.size () - 2) + odd.get (1);
    }

    public static void main (String[]args)
    {
        Scanner sc = new Scanner (System.in);
        int n = sc.nextInt ();
        int arr[] = new int[n];
        for (int i = 0; i < n; i++)
            arr[i] = sc.nextInt ();

        System.out.println (largeSmallSum (arr));
    }
}
```


C++

```cpp
#include <bits/stdc++.h>
using namespace std;

int largeSmallSum(int *array, int n)
{
    int answer, i, j, temp;;
    int even[n], odd[n];
    int evencount = 0, oddcount = 0;
    if(n<=3)
    {
        answer = 0;
    }
    else
    {
        even[0] = array[0];
        evencount = 1;
        for(i=1; i<n; i++)
        {
            if(i%2==0)
            {
                even[evencount] = array[i];
                evencount++;
            }
            else
            {
                odd[oddcount] = array[i];
                oddcount++;
            }
        }

        sort(even, even + evencount);

        sort(odd, odd+oddcount);

        answer = even[evencount-2] + odd[oddcount-2];
    }
    return answer;
}

int main()
{
    int n, result, i;

    cin>>n;

    int array[n];

    for(i=0; i<n; i++)
    cin>>array[i];

    result = largeSmallSum(array, n);

    cout<<result;

    return 0;
}
```

Coding Question 3

Implement the following Function

The function def ProductSmallestPair(sum, arr) accepts an integers sum and an integer array arr of size n. Implement the function to find the pair, (arr[j], arr[k]) where j!=k, Such that arr[j] and arr[k] are the least two elements of array (arr[j] + arr[k] <= sum) and return the product of element of this pair

<u>NOTE</u>

- Return -1 if array is empty or if n<2
- Return 0, if no such pairs found
- All computed values lie within integer range

<u>Example</u>

Input

sum:9

Arr:5 2 4 3 9 7 1

Output

2

<u>Explanation</u>

Pair of least two element is (2, 1) 2 + 1 = 3 < 9, Product of (2, 1) 2*1 = 2. Thus, output is 2

Sample Input

sum:4

Arr:9 8 3 -7 3 9

Sample Output

-21

C++

```cpp
#include<bits/stdc++.h>
using namespace std;

int productSmallestPair(int *array, int n, int sum)
{
    int answer, temp, i, j, check;
    if(n<=2)
    {
        answer = -1;
    }
    else
    {
        sort(array, array+n);
        check = array[0] + array[1];
        if(check<=sum)
        {
            answer = array[0] * array[1];
        }
        else
        {
            answer = 0;
        }
    }
    return answer;
}
int main()
{
    int n, sum, result;

    cin>>sum>>n;
    int array[n];

    for(int i=0; i<n; i++)
    cin>>array[i];

    result = productSmallestPair(array, n, sum);

    cout<<result;

    return 0;
}
```


JAVA

```java
import java.util.*;
class Solution
{
    public static int productSmallestPair (int arr[], int n, int sum)
    {
        if (n <= 2)
            return -1;
        int ans, temp, check;
        for (int i = 0; i < n; i++)
        {
            for (int j = i + 1; j < n; j++)
            {
                if (arr[i] > arr[j])
                {
                    temp = arr[i];
                    arr[i] = arr[j];
                    arr[j] = temp;
                }
            }
        }
        check = arr[0] + arr[1];

        if (check <= sum)
            return arr[0] * arr[1];
        else
            return 0;
    }

    public static void main (String[]args)
    {
        Scanner sc = new Scanner (System.in);
        int sum = sc.nextInt ();
        int n = sc.nextInt ();
        int arr[] = new int[n];

        for (int i = 0; i < n; i++)
            arr[i] = sc.nextInt ();
        System.out.println (productSmallestPair (arr, n, sum));
    }
}
```

Python

*freferer.py - /Users/shub

```python
def ProductSmallestPair(sum,arr):

    n = len(arr)
    if not arr or n < 2 :
        return -1

    arr.sort()
    if arr[0] + arr[1] <= sum :
        return arr[0]* arr[1]
    else:
        return 0
sum = int(input())
arr = list(map(int, input().split()))
print(ProductSmallestPair(sum, arr))
```

Coding Question 4

A carry is a digit that is transferred to left if sum of digits exceeds 9 while adding two numbers from right-to-left one digit at a time

You are required to implement the following function, Int NumberOfCarries(int num1 , int num2);

The functions accepts two numbers ‘num1’ and ‘num2’ as its arguments. You are required to calculate and return the total number of carries generated while adding digits of two numbers ‘num1’ and ‘ num2’.

Assumption: num1, num2>=0

Example:

- Input
  - Num 1: 451
  - Num 2: 349
- Output
  - 2

Explanation:

Adding ‘num 1’ and ‘num 2’ right-to-left results in 2 carries since (1+9) is 10. 1 is carried and (5+4=1) is 10, again 1 is carried. Hence 2 is returned.

Sample Input

Num 1: 23

Num 2: 563

Sample Output

0

C++

```cpp
#include <bits/stdc++.h>
using namespace std;
int count_carry(string a, string b)
{
    int carry = 0;
    int count = 0;
    int len_a = a.length(), len_b = b.length();

    while (len_a != 0 || len_b != 0) {
        int x = 0, y = 0;
        if (len_a > 0) {
            x = a[len_a - 1]-'0';
            len_a--;
        }
        if (len_b > 0) {
            y = b[len_b - 1]-'0';
            len_b--;
        }
        int sum = x + y + carry;
        if (sum >= 10) {
            carry = 1;
            count++;
        }

        else
            carry = 0;
    }
    return count;
}

int main()
{

    string a = "23", b = "563";

    int count = count_carry(a, b);
    if (count == 0)
        cout << "0\n";
    else if (count == 1)
        cout << "1\n";
    else
        cout << count << "\n";

    return 0;
}
```

JAVA

```java
import java.io.*;
class DEBUG
{
static int count_carry(String a, String b)
{
    int carry = 0;
    int count = 0;
    int len_a = a.length(),
        len_b = b.length();

    while (len_a != 0 || len_b != 0)
    {
        int x = 0, y = 0;
        if (len_a > 0)
        {
            x = a.charAt(len_a - 1) - '0';
            len_a--;
        }
        if (len_b > 0)
        {
            y = b.charAt(len_b - 1) - '0';
            len_b--;
        }
        int sum = x + y + carry;
        if (sum >= 10)
        {
            carry = 1;
            count++;
        }
        else
            carry = 0;
    }

    return count;
}
public static void main (String[] args)
{
    String a = "23", b = "563";
    int count = count_carry(a, b);
    if (count == 0)
        System.out.println("0\n");
    else if (count == 1)
        System.out.println("1\n");
    else
        System.out.println(count);
}
}
```

Python

```python
def NumberOfCarries(num1, num2):
    l1 = len(num1)
    l2 = len(num2)
    carry =0
    count =0
    while(l1 != 0 or l2 !=0 ):
        x = 0
        y = 0
        if (l1>0):
            x = int(num1[l1-1])
            l1 -= 1

        if (l2>0):
            y = int(num2[l2-1])
            l2 -= 1
        sum= x + y + carry
        if sum >= 10:
            carry =1
            count += 1
        else:
            carry = 0
    return count

num1 = input()
num2 = input()
print(NumberOfCarries(num1,num2))
```