Accenture Sections	Information	Questions and Time
Cognitive Ability	English AbilityCritical Thinking and Problem SolvingAbstract 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	CC++Dot NetJAVAPython	2 Ques in 45 mins

DEBUG WITH SHUBHAM

Accenture Technical Assessment Detailed Overview

20-SEP-2024 Coding Question



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



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



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



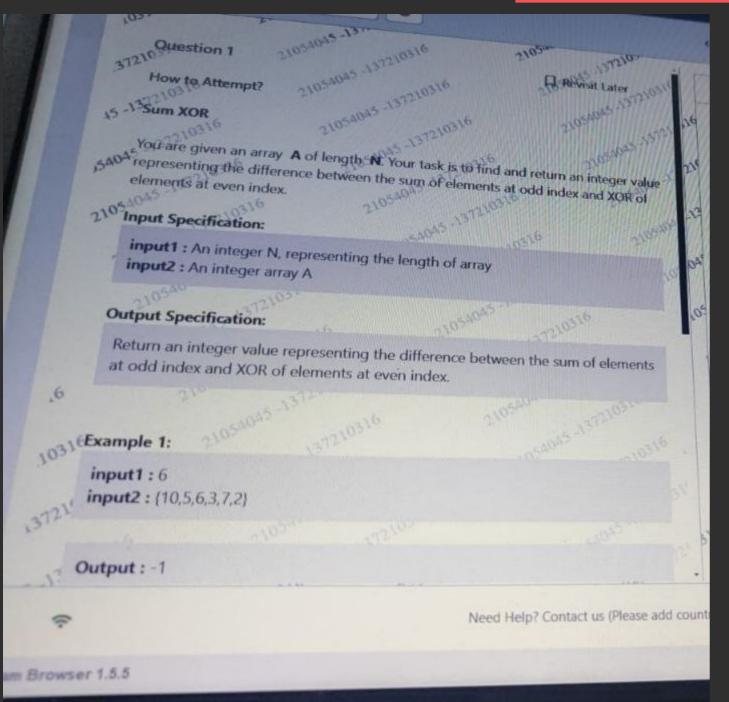
https://topmate.io/debugwithshubham

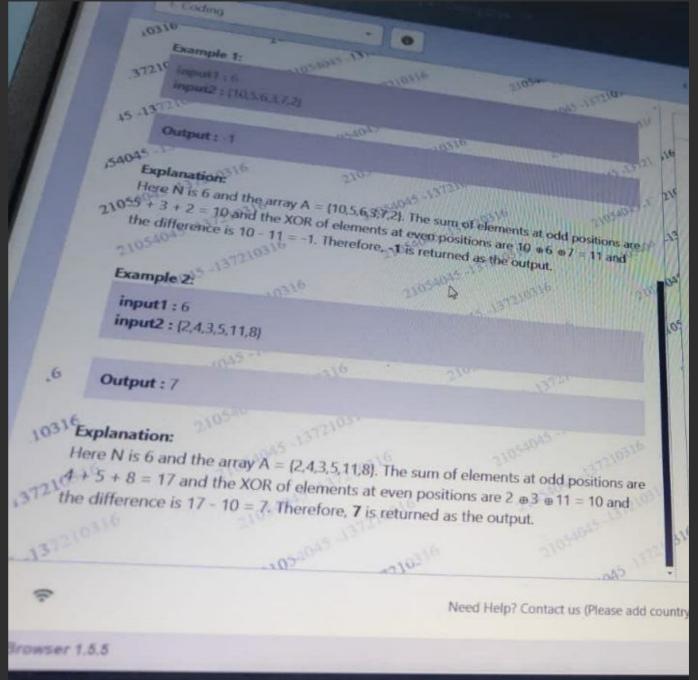


https://t.me/debugwithshubham



Question-1





Python

```
main.py
 1 def difference_sum_xor(N, A):
        odd_sum = 0
 3
        even_xor = 0
 4 -
        for i in range(N):
            if i % 2 == 0:
 5 -
                even_xor ^= A[i]
            else:
                 odd_sum += A[i]
 8
        return odd_sum - even_xor
 9
    input1 = 6
10
    input2 = [10,5,6,3,7,2]
12
    result = difference_sum_xor(input1, input2)
13
    print(result)
14
15
```

Main.java

JAVA

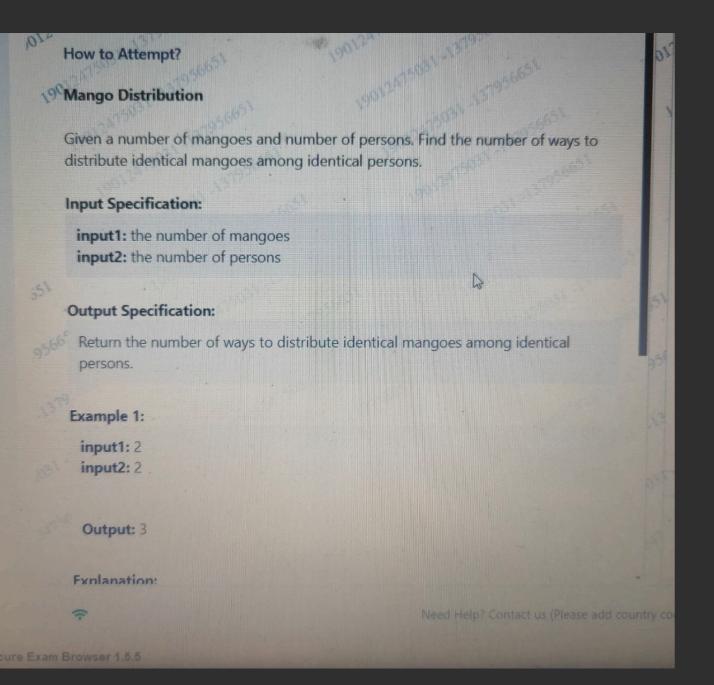


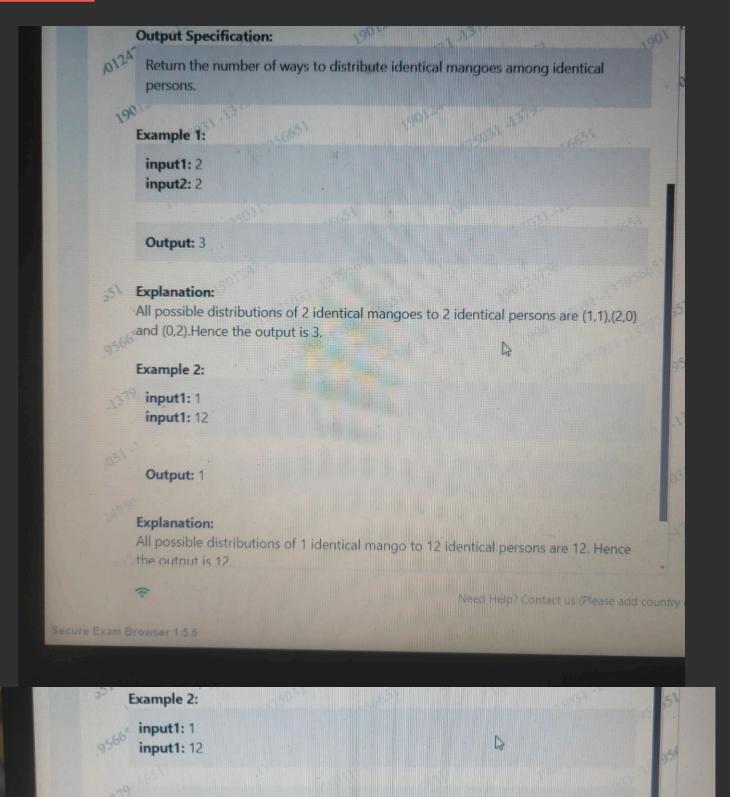
```
1 public class DifferenceSumXor {
        public static int differenceSumXor(int N, int[] A) {
 3
            int oddSum = 0;
 4
            int evenXor = 0;
            for (int i = 0; i < N; i++) {
 6
 7
                if (i \% 2 == 0) {
 8
                    evenXor ^= A[i];
 9
                } else {
                    oddSum += A[i];
10
                }
11
12
13
            return oddSum - evenXor;
14
15
        public static void main(String[] args) {
16
            int input1 = 6;
17
            int[] input2 = {10, 5, 6, 3, 7, 2};
18
19
            int result = differenceSumXor(input1, input2);
20
            System.out.println(result);
21
22 }
22
```

C++

```
main.cpp
 1 #include <iostream>
 2 #include <vector>
3 using namespace std;
4 int differenceSumXor(int N, vector<int>& A) {
 5
        int oddSum = 0;
        int evenXor = 0;
 6
 7 -
        for (int i = 0; i < N; i++) {
            if (i % 2 == 0) {
 8 -
                evenXor ^= A[i];
 9
            } else {
10
                oddSum += A[i];
11
12
            }
13
        }
14
        return oddSum - evenXor;
15 }
16 int main() {
17
        int input1 = 6;
18
        vector<int> input2 = {10, 5, 6, 3, 7, 2};
19
20
        int result = differenceSumXor(input1, input2);
        cout << result << endl;</pre>
21
22
        return 0;
23 }
21
```

Question-2





All possible distributions of 1 identical mango to 12 identical persons are 12. Hence

Output: 1

the output is 12.



First Approach

∝ Share



```
main.cpp
```

```
1 #include <iostream>
   using namespace std;
   long long factorial(int n) {
        long long result = 1;
        for (int i = 2; i \le n; i++) {
            result *= i;
        return result;
9
   long long combination(int n, int k) {
11
        if (k > n) return 0;
12
        return factorial(n) / (factorial(k) * factorial(n - k));
13 }
14 long long mangoDistribution(int mangoes, int persons) {
15
        return combination(mangoes + persons - 1, persons - 1);
16 }
17 int main() {
18
        int input1 = 2;
19
        int input2 = 2;
        cout << mangoDistribution(input1, input2) << endl;</pre>
20
        input1 = 1; input2 = 12;
21
22
        cout << mangoDistribution(input1, input2) << endl;</pre>
23
        return 0;
24 }
```

main.py

Main.java

```
1 import java.math.BigInteger;
   public class MangoDistribution {
       public static BigInteger combination(int n, int k) {
            BigInteger numerator = BigInteger.ONE;
           BigInteger denominator = BigInteger.ONE;
            for (int i = 0; i < k; i++) {
               numerator = numerator.multiply(BigInteger.valueOf(n - i));
 8
                denominator = denominator.multiply(BigInteger.valueOf(i + 1));
9
10
            return numerator.divide(denominator);
11
12
       public static BigInteger mangoDistribution(int mangoes, int persons) {
13
            return combination(mangoes + persons - 1, persons - 1);
14
15
       public static void main(String[] args) {
16
            int input1 = 2;
17
            int input2 = 2;
18
            System.out.println(mangoDistribution(input1, input2));
19
20
            input1 = 1;
21
            input2 = 12;
22
            System.out.println(mangoDistribution(input1, input2));
23
24 }
```

Python

```
1 import math
2 - def mango_distribution(mangoes, persons):
        return math.comb(mangoes + persons - 1, persons - 1)
3
   input1 = 2
   input2 = 2
   print(mango_distribution(input1, input2))
8
   input1 = 1
   input2 = 12
   print(mango_distribution(input1, input2))
12
```

```
#include <iostream>
using namespace std;
long long binomialCoefficients(int n, int m) {
   long long res = 1;
   if (m > n - m) {
       m = n - m;
   }
   for (int i = 0; i < m; i++) {
       res *= (n - i);
       res = (i + 1);
   }
    return res;
int calcWays(int m, int n) {
   if (m == 1) {
        return n; // Changed to return n when m == 1
   }
   if (m < n) {
        return 0;
   long long ways = binomialCoefficients(n + m - 1, n - 1);
   return static_cast<int>(ways);
int main() {
   int m = 2;
   int n = 2;
   cout << calcWays(m, n) << endl; // Output: 2</pre>
    return 0;
```

without built in function



```
Python
main.py
 1 def binomial_coefficients(n, m):
       res = 1
       if m > n - m:
          m = n - m
       for i in range(m):
          res *= (n - i)
          res /= (i + 1)
       return res
10 def calc_ways(m, n):
       if m == 1:
11
12
           return n
13
       if m < n:
14
          return 0
15
       ways = binomial_coefficients(n + m - 1, n - 1)
16
       return int(ways)
17
18 m = 2
19 n = 2
20 print(calc_ways(m, n))
```

```
import java.util.Scanner;
public class BinomialCoefficients {
   public static long binomialCoefficients(int n, int m) {
       long res = 1;
       if (m > n - m) {
           m = n - m;
       }
       for (int i = 0; i < m; i++) {
           res *= (n - i);
           res = (i + 1);
       }
       return res;
   }
   public static int calcWays(int m, int n) {
       if (m == 1) {
           return n; // Changed to return n when m == 1
       if (m < n) {
           return 0;
       long ways = binomialCoefficients(n + m - 1, n - 1);
       return (int) ways;
   }
   public static void main(String[] args) {
       int m = 2;
       int n = 2;
       System.out.println(calcWays(m, n)); // Output: 2
   }
```

Best Case

```
main.py +

Python

def distribute_mangoes(mangoes, persons):
    ways = (mangoes + persons - 1) // (persons - 1)
    return ways
    mangoes = 1
    persons = 12
    result = distribute_mangoes(mangoes, persons)
    print(result)
```

```
JAVA
                                                          ∞ Share
Main.java
                                                                       Run
 1 - public class MangoDistribution {
       public static int distributeMangoes(int mangoes, int persons) {
           int ways = (mangoes + persons - 1) / persons;
            return ways;
 5
       }
 6
       public static void main(String[] args) {
           int mangoes = 1;
           int persons = 12;
           int result = distributeMangoes(mangoes, persons);
10
           System.out.println(result);
11
12
13 }
14
```

C++

```
main.cpp
 1 #include <iostream>
   using namespace std;
 3
   int distributeMangoes(int mangoes, int persons) {
        int ways = (mangoes + persons - 1) / persons;
 5
 6
        return ways;
 7 }
 8
9 int main() {
        int mangoes = 1;
10
        int persons = 12;
11
        int result = distributeMangoes(mangoes, persons);
12
        cout << result << endl;</pre>
13
14
        return 0;
15 }
16
```