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

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

Question 1

How to Attempt?

Revisit Later

Sum XOR

You are given an array **A** of length **N**. Your task is to find and return an integer value representing the difference between the sum of elements at odd index and XOR of elements at even index.

Input Specification:

input1 : An integer N, representing the length of array
input2 : An integer array A

Output Specification:

Return an integer value representing the difference between the sum of elements at odd index and XOR of elements at even index.

Example 1:

input1 : 6
input2 : {10,5,6,3,7,2}

Output : -1

Need Help? Contact us (Please add count

Browser 1.5.5

Example 1:

input1 : 6
input2 : {10,5,6,3,7,2}

Output : -1

Explanation:

Here N is 6 and the array A = {10,5,6,3,7,2}. The sum of elements at odd positions are 5 + 3 + 2 = 10 and the XOR of elements at even positions are 10 ⊕ 6 ⊕ 7 = 11 and the difference is 10 - 11 = -1. Therefore, **-1** is returned as the output.

Example 2:

input1 : 6
input2 : {2,4,3,5,11,8}

Output : 7

Explanation:

Here N is 6 and the array A = {2,4,3,5,11,8}. The sum of elements at odd positions are 4 + 5 + 8 = 17 and the XOR of elements at even positions are 2 ⊕ 3 ⊕ 11 = 10 and the difference is 17 - 10 = 7. Therefore, **7** is returned as the output.

Need Help? Contact us (Please add country

Browser 1.5.5

Python

main.py

```python
def difference_sum_xor(N, A):
    odd_sum = 0
    even_xor = 0
    for i in range(N):
        if i % 2 == 0:
            even_xor ^= A[i]
        else:
            odd_sum += A[i]
    return odd_sum - even_xor
input1 = 6
input2 = [10,5,6,3,7,2]

result = difference_sum_xor(input1, input2)
print(result)

```

JAVA

Main.java

```java
public class DifferenceSumXor {
    public static int differenceSumXor(int N, int[] A) {
        int oddSum = 0;
        int evenXor = 0;

        for (int i = 0; i < N; i++) {
            if (i % 2 == 0) {
                evenXor ^= A[i];
            } else {
                oddSum += A[i];
            }
        }
        return oddSum - evenXor;
    }
    public static void main(String[] args) {
        int input1 = 6;
        int[] input2 = {10, 5, 6, 3, 7, 2};

        int result = differenceSumXor(input1, input2);
        System.out.println(result);
    }
}
```

C++

main.cpp

```cpp
#include <iostream>
#include <vector>
using namespace std;
int differenceSumXor(int N, vector<int>& A) {
    int oddSum = 0;
    int evenXor = 0;
    for (int i = 0; i < N; i++) {
        if (i % 2 == 0) {
            evenXor ^= A[i];
        } else {
            oddSum += A[i];
        }
    }
    return oddSum - evenXor;
}
int main() {
    int input1 = 6;
    vector<int> input2 = {10, 5, 6, 3, 7, 2};

    int result = differenceSumXor(input1, input2);
    cout << result << endl;
    return 0;
}
```

Question-2

How to Attempt?

Mango Distribution

Given a number of mangoes and number of persons. Find the number of ways to distribute identical mangoes among identical persons.

Input Specification:

input1: the number of mangoes
input2: the number of persons

Output Specification:

Return the number of ways to distribute identical mangoes among identical persons.

Example 1:

input1: 2
input2: 2

Output: 3

Explanation:

Need Help? Contact us (Please add country co

ecure Exam Browser 1.5.5

Output Specification:

Return the number of ways to distribute identical mangoes among identical persons.

Example 1:

input1: 2
input2: 2

Output: 3

Explanation:
All possible distributions of 2 identical mangoes to 2 identical persons are (1,1),(2,0) and (0,2).Hence the output is 3.

Example 2:

input1: 1
input1: 12

Output: 1

Explanation:
All possible distributions of 1 identical mango to 12 identical persons are 12. Hence the output is 12.

Need Help? Contact us (Please add country

Secure Exam Browser 1.5.5

Example 2:

input1: 1
input1: 12

Output: 1

Explanation:
All possible distributions of 1 identical mango to 12 identical persons are 12. Hence the output is 12.

!

First Approach

C++

main.cpp

Share Run

```cpp
#include <iostream>
using namespace std;
long long factorial(int n) {
    long long result = 1;
    for (int i = 2; i <= n; i++) {
        result *= i;
    }
    return result;
}
long long combination(int n, int k) {
    if (k > n) return 0;
    return factorial(n) / (factorial(k) * factorial(n - k));
}
long long mangoDistribution(int mangoes, int persons) {
    return combination(mangoes + persons - 1, persons - 1);
}
int main() {
    int input1 = 2;
    int input2 = 2;
    cout << mangoDistribution(input1, input2) << endl;
    input1 = 1; input2 = 12;
    cout << mangoDistribution(input1, input2) << endl;
    return 0;
}
```

JAVA

Main.java

Share Run

```java
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));
            denominator = denominator.multiply(BigInteger.valueOf(i + 1));
        }
        return numerator.divide(denominator);
    }
    public static BigInteger mangoDistribution(int mangoes, int persons) {
        return combination(mangoes + persons - 1, persons - 1);
    }
    public static void main(String[] args) {
        int input1 = 2;
        int input2 = 2;
        System.out.println(mangoDistribution(input1, input2));

        input1 = 1;
        input2 = 12;
        System.out.println(mangoDistribution(input1, input2));
    }
}
```

Python

main.py +

```python
import math
def mango_distribution(mangoes, persons):
    return math.comb(mangoes + persons - 1, persons - 1)

input1 = 2
input2 = 2
print(mango_distribution(input1, input2))

input1 = 1
input2 = 12
print(mango_distribution(input1, input2))

```

without built in function

C++

```cpp
#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
    return 0;
}
```

Python

main.py

```python
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

def calc_ways(m, n):
    if m == 1:
        return n
    if m < n:
        return 0
    ways = binomial_coefficients(n + m - 1, n - 1)
    return int(ways)

m = 2
n = 2
print(calc_ways(m, n))
```

JAVA

```java
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

Python

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)

```

C++

main.cpp

```cpp
#include <iostream>
using namespace std;

int distributeMangoes(int mangoes, int persons) {
    int ways = (mangoes + persons - 1) / persons;
    return ways;
}

int main() {
    int mangoes = 1;
    int persons = 12;
    int result = distributeMangoes(mangoes, persons);
    cout << result << endl;
    return 0;
}

```

JAVA

Main.java Share Run

```java
public class MangoDistribution {
    public static int distributeMangoes(int mangoes, int persons) {
        int ways = (mangoes + persons - 1) / persons;
        return ways;
    }

    public static void main(String[] args) {
        int mangoes = 1;
        int persons = 12;
        int result = distributeMangoes(mangoes, persons);
        System.out.println(result);
    }
}

```