DEBUG WITH SHUBHAM

Cognizant Technical Assessment Detailed Overview

09-Oct-2024 Coding Interview Questions


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


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

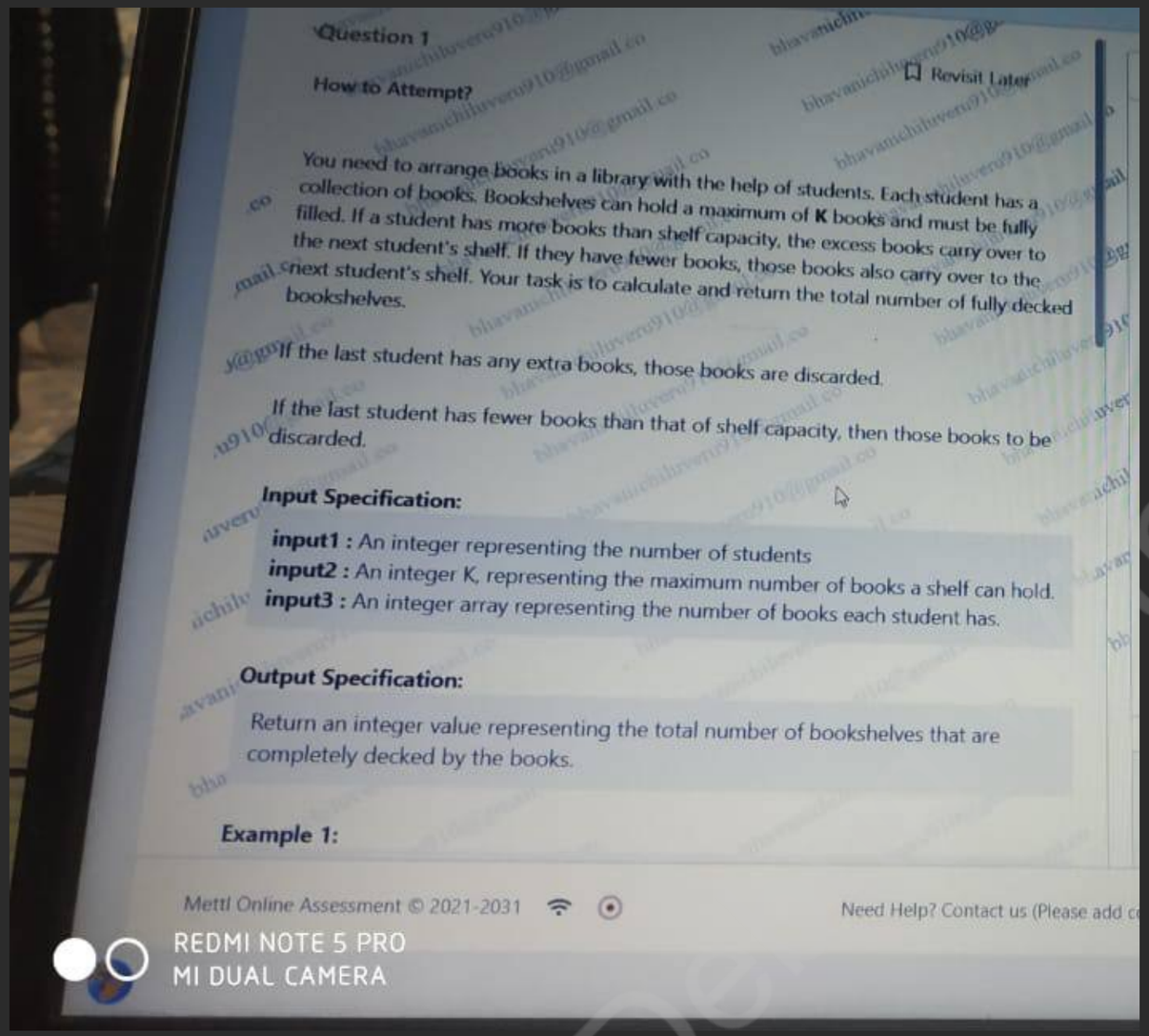

https://www.instagram.com/debugwithshubham/


https://topmate.io/debugwithshubham


https://t.me/debugwithshubham

ALL SOLUTION (C++, JAVA, PYTHON) UPLOADED IN GITHUB WITH QUESTION NAME

QUESTION NAME: Fully Filled Bookshelves with Student Book CollectioN



Question 1

How to Attempt?

Revisit Later

You need to arrange books in a library with the help of students. Each student has a collection of books. Bookshelves can hold a maximum of **K** books and must be fully filled. If a student has more books than shelf capacity, the excess books carry over to the next student's shelf. If they have fewer books, those books also carry over to the next student's shelf. Your task is to calculate and return the total number of fully decked bookshelves.

If the last student has any extra books, those books are discarded.

If the last student has fewer books than that of shelf capacity, then those books to be discarded.

Input Specification:

input1 : An integer representing the number of students
input2 : An integer K, representing the maximum number of books a shelf can hold.
input3 : An integer array representing the number of books each student has.

Output Specification:

Return an integer value representing the total number of bookshelves that are completely decked by the books.

Example 1:

Mettl Online Assessment © 2021-2031

Need Help? Contact us (Please add c

REDMI NOTE 5 PRO
MI DUAL CAMERA

```
num_students = 5,
K = 10,
books = [15, 8, 17, 12, 5]
output= 5
```

QUESTION NAME:Fully Filled Bookshelves with Student Book Collections

main.py

```python
def fully_decked_shelves(num_students: int, K: int, books: list) -> int:
    total_shelves = 0
    carry_over = 0
    for student_books in books:
        student_books += carry_over
        full_shelves = student_books // K
        total_shelves += full_shelves
        carry_over = student_books % K
    return total_shelves
num_students = 5
K = 10
books = [15, 8, 17, 12, 5]
print(fully_decked_shelves(num_students, K, books))

```

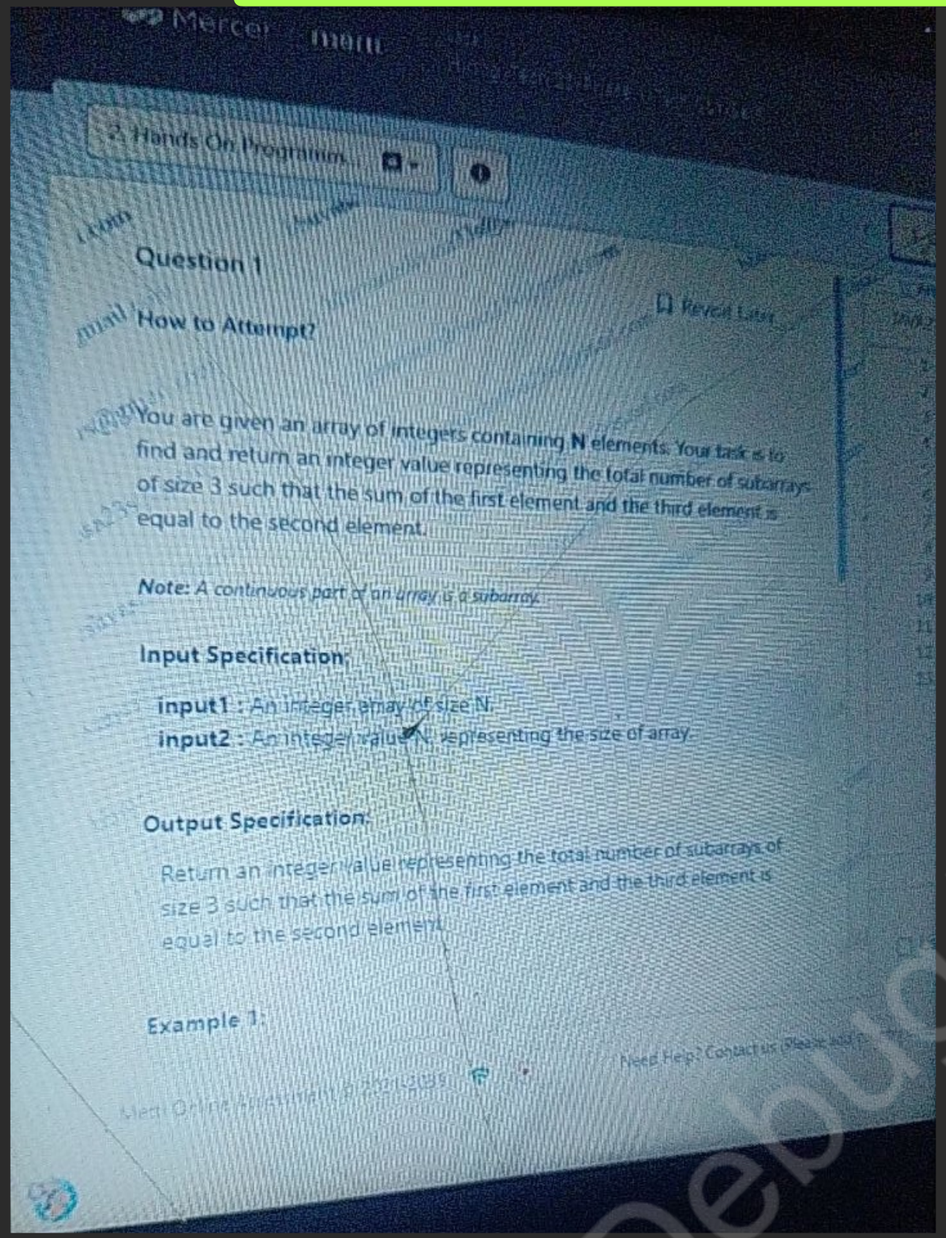
main.cpp

```cpp
#include <iostream>
#include <vector>

using namespace std;

int fullyDeckedShelves(int numStudents, int K, vector<int>& books) {
    int totalShelves = 0;
    int carryOver = 0;

    for (int studentBooks : books) {
        studentBooks += carryOver;
        int fullShelves = studentBooks / K;
        totalShelves += fullShelves;
        carryOver = studentBooks % K;
    }

    return totalShelves;
}

int main() {
    int numStudents = 5;
    int K = 10;
    vector<int> books = {15, 8, 17, 12, 5};

    cout << fullyDeckedShelves(numStudents, K, books) << endl;  // Output: 5

    return 0;
}

```

Main.java

```java
public class FullyDeckedShelves {
    public static int fullyDeckedShelves(int numStudents, int K, int[] books) {
        int totalShelves = 0;
        int carryOver = 0;

        for (int studentBooks : books) {
            studentBooks += carryOver;
            int fullShelves = studentBooks / K;
            totalShelves += fullShelves;
            carryOver = studentBooks % K;
        }

        return totalShelves;
    }

    public static void main(String[] args) {
        int numStudents = 5;
        int K = 10;
        int[] books = {15, 8, 17, 12, 5};

        System.out.println(fullyDeckedShelves(numStudents, K, books));  // Output: 5
    }
}

```

Question name: Count Special Subarrays of Size



N= 6
Arr = [1, 2, 1, 5, 6, 5]
output= 1

Question name: Count Special Subarrays of Size

main.py

```python
def count_special_subarrays(arr: list) -> int:
    count = 0
    n = len(arr)
    for i in range(n - 2):
        if arr[i] + arr[i + 2] == arr[i + 1]:
            count += 1

    return count
arr = [1, 2, 1, 6, 5, 5]

print(count_special_subarrays(arr))

```

Main.java

Share

```java
public class SpecialSubarrays {
    public static int countSpecialSubarrays(int[] arr) {
        int count = 0;
        int n = arr.length;

        for (int i = 0; i < n - 2; i++) {
            if (arr[i] + arr[i + 2] == arr[i + 1]) {
                count++;
            }
        }

        return count;
    }

    public static void main(String[] args) {
        int[] arr = {1, 2, 1, 6, 5, 5};

        System.out.println(countSpecialSubarrays(arr));  // Output will be 2
    }
}
```

main.cpp

```cpp
#include <iostream>
#include <vector>

using namespace std;

int countSpecialSubarrays(const vector<int>& arr) {
    int count = 0;
    int n = arr.size();

    for (int i = 0; i < n - 2; i++) {
        if (arr[i] + arr[i + 2] == arr[i + 1]) {
            count++;
        }
    }

    return count;
}

int main() {
    vector<int> arr = {1, 2, 1, 6, 5, 5};

    cout << countSpecialSubarrays(arr) << endl;  // Output: 2

    return 0;
}
```