

Trạng thái	Đã xong
Bắt đầu vào lúc	Thứ Sáu, 18 tháng 4 2025, 11:10 AM
Kết thúc lúc	Thứ Sáu, 18 tháng 4 2025, 11:22 AM
Thời gian thực hiện	12 phút 37 giây

Câu hỏi 1

Đúng

Đạt điểm 1,00

Mô tả tiếng Việt:

Cho mảng 2 chiều chứa các số nguyên, kích thước M x N.

Hiện thực hàm:

```
int ascendingRows(int arr[][1000], int row, int col);
```

Trong đó: **arr**, **row** và **col** lần lượt là mảng 2 chiều, số hàng và số cột của mảng. Một hàng trong mảng được gọi là **HN1** nếu trong hàng đó, mỗi phần tử đều có giá trị không lớn hơn các phần tử đứng sau nó. Tìm số hàng **HN1** có trong mảng.

Ghi chú: (Các) thư viện **iostream**, và **string** đã được khai báo, và **namespace std** đã được sử dụng.

English version:

Given a two-dimensional array whose each element is integer, its size is M x N.

Implement the following function:

```
int ascendingRows(int arr[][1000], int row, int col);
```

Where **arr**, **row** and **col** are the given two-dimensional array, its number of rows and its number of columns. A row of the given array is called as **HN1** if on this row, each element's value is not higher than all elements after it. Find the number of **HN1** rows of the given array.

Note: Libraries **iostream**, and **string** have been imported, and **namespace std** has been used.

For example:

Test	Result
<pre>int arr[][1000] = {{32,4,9},{-80,37,71}, {-91,-79,-55}}; cout << ascendingRows(arr,3, 3);</pre>	2
<pre>int arr[][1000] = {{-28,-8,-60,18}, {-100,44,-1,24},{-94,92,-70,75}}; cout << ascendingRows(arr,3,4);</pre>	0

Answer: (penalty regime: 0 %)

Reset answer

```
1 int ascendingRows(int arr[][1000], int row, int col) {
2     int count = 0;
3     for(int i = 0; i < row; ++i){
4         bool check = true;
5         for(int j = 0; j < col - 1; ++j){
6             if(arr[i][j] > arr[i][j+1])
7                 {
8                     check = false;
9                     break;
10                }
11        }
12        if(check) count++;
13    }
14    return count;
15 }
```



	Test	Expected	Got	
✓	<pre>int arr[][1000] = {{32,4,9}, {-80,37,71},{-91,-79,-55}}; cout << ascendingRows(arr,3, 3);</pre>	2	2	✓
✓	<pre>int arr[][1000] = {{-28,-8,-60,18}, {-100,44,-1,24}, {-94,92,-70,75}}; cout << ascendingRows(arr,3,4);</pre>	0	0	✓

Passed all tests! ✓

Câu hỏi 2

Đúng

Đạt điểm 1,00

Mô tả tiếng Việt:

Cho mảng 2 chiều chứa các số nguyên, kích thước M x N.

Hiện thực hàm:

```
int primeColumns(int arr[][1000], int row, int col);
```

Trong đó: `arr`, `row` và `col` lần lượt là mảng 2 chiều, số hàng và số cột của mảng. Một cột của mảng được gọi là **HN2** nếu tổng tất cả các phần tử trong cột đó là số nguyên tố. Tìm số cột **HN2** có trong mảng.

Ghi chú: (Các) thư viện `iostream`, `vector` và `string` đã được khai báo, và `namespace std` đã được sử dụng.

English version:

Given a two-dimensional array whose each element is integer, its size is M x N.

Implement the following function:

```
int primeColumns(int arr[][1000], int row, int col);
```

Where `arr`, `row` and `col` are the given two-dimensional array, its number of rows and its number of columns. A column of the given array is called as **HN2** if the sum of all elements on it is a prime number. Find the number of **HN2** columns in the given array.

Note: Libraries `iostream`, `vector`, and `string` have been imported, and `namespace std` has been used.

For example:

Test	Result
<pre>int arr[][1000] = {{-64,-28,-3,64}, {-56,90,57,-31}}; cout << primeColumns(arr,2,4);</pre>	0
<pre>int arr[][1000] = {{34,-15,11,-70,-23,24}, {-39,-90,63,-45,-52,48},{-42,92,55,92,82,81}}; cout << primeColumns(arr,3,6);</pre>	1

Answer: (penalty regime: 0 %)

Reset answer

```

1 bool isPrime(int n){
2     if(n < 2) return false;
3
4     for(int i = 2; i*i <= n; ++i){
5         if(n % i == 0) return false;
6     }
7     return true;
8 }
9 int primeColumns(int arr[][1000], int row, int col){
10    int count = 0;
11    for(int i = 0; i < col; ++i){
12        int sum = 0;
13        for(int j = 0; j < row; ++j){
14            sum += arr[j][i];
15        }
16        if(isPrime(sum)) count++;
17    }
18    return count;
19 }

```

	Test	Expected	Got	
✓	int arr[][1000] = { {-64,-28,-3,64}, {-56,90,57,-31}}; cout << primeColumns(arr,2,4);	0	0	✓
✓	int arr[][1000] = { {34,-15,11,-70,-23,24}, {-39,-90,63,-45,-52,48}, {-42,92,55,92,82,81}}; cout << primeColumns(arr,3,6);	1	1	✓

Passed all tests! ✓