

Trạng thái	Đã xong
Bắt đầu vào lúc	Thứ Sáu, 3 tháng 5 2024, 11:07 PM
Kết thúc lúc	Thứ Bảy, 4 tháng 5 2024, 2:13 AM
Thời gian thực hiện	3 giờ 5 phút

Câu hỏi 1

Đúng

Đạt điểm 1,00

Mô tả tiếng Việt:

Cho mảng 2 chiều chứa các số nguyên, kích thước $M \times N$.

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 $M \times N$.

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 HN1 = true;
5       for(int j = 0; j < col - 1 ; j++)
6       {
7           if(arr[i][j] > arr[i][j + 1])
8           {
9               HN1 = false;
10              break;
11          }
12      }
13      if(HN1) count++;
14  }
15  return count;
16 }

```

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

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 $M \times N$.

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 num) {
2     if (num <= 1) {
```

```

3         return false;
4     }
5     for (int i = 2; i * i <= num; ++i) {
6         if (num % i == 0) {
7             return false;
8         }
9     }
10    return true;
11 }
12 int primeColumns(int arr[][1000], int row, int col) {
13     int count = 0;
14     for(int j = 0; j < col ; j++)
15     {
16         int sumcol = 0;
17         for(int i = 0; i < row; i++)
18         {
19             sumcol += arr[i][j];
20         }
21         if(isPrime(sumcol)) count++;
22     }
23     return count;
24 }

```

	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! ✓

