

Trạng thái	Đã xong
Bắt đầu vào lúc	Thứ Tư, 7 tháng 5 2025, 12:26 PM
Kết thúc lúc	Thứ Tư, 7 tháng 5 2025, 12:44 PM
Thời gian thực hiện	18 phút 31 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 specialCells(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 ô trong mảng được gọi là **HN3** nếu tổng tất cả các phần tử trong hàng chứa ô đó và tổng tất cả các phần tử trong cột chứa ô đó đều là số nguyên tố. Tìm số ô **HN3** 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 specialCells(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 cell is called as **HN3** if the sum of all elements of the row containing it and the sum of all elements of the column containing it are prime numbers. Find the number of **HN3** cells of 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] = {{37,53,74},{12,37,60},{98,13,15}}; cout &lt;&lt; specialCells(arr,3,3);</pre>	2
<pre>int arr[][1000] = {{1,87,26},{97,95,88},{57,60,46}}; cout &lt;&lt; specialCells(arr,3,3);</pre>	0

Answer: (penalty regime: 0 %)

Reset answer

```
1 bool isPrime(int num){
2     if(num < 2) return false;
3
4     for(int i = 2; i*i <= num; ++i){
5         if(num % i == 0) return false;
6     }
7     return true;
8 }
9
10 int specialCells(int arr[][1000], int row, int col){
11     vector<int> rowSum(row, 0);
12     vector<int> colSum(col, 0);
13     int count = 0;
14
15     for(int i = 0; i < row; ++i){
```

```

16   for(int j = 0; j < col; ++j){
17       rowSum[i] += arr[i][j];
18       colSum[j] += arr[i][j];
19   }
20 }
21 for(int i = 0; i < row; i++){
22     for(int j = 0; j < col; ++j){
23         if(isPrime(rowSum[i]) && isPrime(colSum[j]))
24             ++count;
25     }
26 }
27 return count;
28 }
29

```

	Test	Expected	Got	
✓	<pre>int arr[][1000] = {{37,53,74},{12,37,60}, {98,13,15}}; cout &lt;&lt; specialCells(arr,3,3);</pre>	2	2	✓
✓	<pre>int arr[][1000] = {{1,87,26}, {97,95,88},{57,60,46}};; cout &lt;&lt; specialCells(arr,3,3);</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 subMatrix(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 mảng con kích thước 2x2 thuộc mảng đã cho được gọi là **HN4** nếu tổng tất cả các phần tử trong nó là một số lẻ. Tìm số mảng con **HN4** trong mảng đã cho.

**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 subMatrix(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 sub-array whose size is 2x2 is called as **HN4** if the sum of all elements of it is an odd number. Find the number of sub-array of 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] = {{66,16,71},{25,81,61}, {2,10,34}}; cout &lt;&lt; subMatrix(arr,3, 3);</pre>	1
	4

```
int arr[][1000] = {{44,45,89},{82,91,34},
{83,87,33},{65,51,66}};
cout << subMatrix(arr,4, 3);
```

Answer: (penalty regime: 0 %)

Reset answer

```
1 bool isOdd(int n){
2     return (n % 2 == 0)? false : true;
3 }
4 int subMatrix(int arr[][1000], int row, int col)
5 {
6     int count = 0;
7     for(int i = 0; i < row - 1; ++i){
8         int sum = 0;
9         for(int j = 0; j < col - 1; ++j){
10             sum = arr[i][j] + arr[i][j+1] + arr[i][j+2];
11             if(isOdd(sum))
12                 ++count;
13         }
14     }
15     return count;
16 }
```

	Test	Expected	Got	
✓	int arr[][1000] = { {66,16,71},{25,81,61}, {2,10,34}}; cout << subMatrix(arr,3, 3);	1	1	✓
✓	int arr[][1000] = {{44,45,89}, {82,91,34},{83,87,33}, {65,51,66}}; cout << subMatrix(arr,4, 3);	4	4	✓

Passed all tests! ✓