

Trạng thái	Đã xong
Bắt đầu vào lúc	Thứ Tư, 16 tháng 4 2025, 4:27 PM
Kết thúc lúc	Thứ Tư, 16 tháng 4 2025, 4:35 PM
Thời gian thực hiện	8 phút 49 giây

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 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 $M \times N$.

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 << specialCells(arr,3,3);</pre>	2
<pre>int arr[][1000] = {{1,87,26},{97,95,88},{57,60,46}}; cout << specialCells(arr,3,3);</pre>	0

Answer: (penalty regime: 0 %)

Reset answer

```

1 // Kiểm tra số nguyên tố
2 bool isPrime(int n) {
3     if (n <= 1) return false;
4     if (n <= 3) return true;
5     if (n % 2 == 0 || n % 3 == 0) return false;
6
7     for (int i = 5; i * i <= n; i += 6) {
8         if (n % i == 0 || n % (i + 2) == 0) {
9             return false;
10        }
11    }
12    return true;
13 }
14
15 int specialCells(int arr[][1000], int row, int col) {
16     // Nếu mảng rỗng
17     if (row == 0 || col == 0) {
18         return 0;
19     }
20
21     // Tính tổng của từng hàng
22     vector<int> rowSum(row, 0);
23     for (int i = 0; i < row; i++) {
24         for (int j = 0; j < col; j++) {
25             rowSum[i] += arr[i][j];
26         }
27     }
28
29     // Tính tổng của từng cột
30     vector<int> colSum(col, 0);
31     for (int j = 0; j < col; j++) {
32         for (int i = 0; i < row; i++) {
33             colSum[j] += arr[i][j];

```

```

34     }
35 }
36
37 // Đếm số ô HN3
38 int count = 0;
39 for (int i = 0; i < row; i++) {
40     for (int j = 0; j < col; j++) {
41         // Kiểm tra nếu tổng hàng và tổng cột đều là số nguyên tố
42         if (isPrime(rowSum[i]) && isPrime(colSum[j])) {
43             count++;
44         }
45     }
46 }
47
48 return count;
49 }

```

	Test	Expected	Got	
✓	int arr[][1000] = {{37,53,74},{12,37,60},{98,13,15}}; cout << specialCells(arr,3,3);	2	2	✓
✓	int arr[][1000] = {{1,87,26},{97,95,88},{57,60,46}};; cout << specialCells(arr,3,3);	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 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 2×2 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 $M \times N$.

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 2×2 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 << subMatrix(arr,3, 3);</pre>	1
<pre>int arr[][1000] ={{44,45,89},{82,91,34},{83,87,33},{65,51,66}}; cout << subMatrix(arr,4, 3);</pre>	4

Answer: (penalty regime: 0 %)

Reset answer

```
1 int subMatrix(int arr[][1000], int row, int col) {
2     // Nếu mảng kích thước nhỏ hơn 2x2, không thể có mảng con 2x2
3     if (row < 2 || col < 2) {
4         return 0;
5     }
6
7     int count = 0;
8
9     // Duyệt qua từng vị trí có thể bắt đầu một mảng con 2x2
10    for (int i = 0; i <= row - 2; i++) {
11        for (int j = 0; j <= col - 2; j++) {
12            // Tính tổng của mảng con 2x2 bắt đầu tại vị trí (i,j)
13            int sum = arr[i][j] + arr[i][j+1] + arr[i+1][j] + arr[i+1][j+1];
14
15            // Kiểm tra nếu tổng là số lẻ
16            if (sum % 2 == 1) {
17                count++;
18            }
19        }
20    }
21
22    return count;
23 }
```

	Test	Expected	Got	
✓	<pre>int arr[][1000] = {{66,16,71},{25,81,61},{2,10,34}}; cout << subMatrix(arr,3, 3);</pre>	1	1	✓

	Test	Expected	Got	
✓	<pre>int arr[][1000] ={{44,45,89},{82,91,34},{83,87,33},{65,51,66}}; cout << subMatrix(arr,4, 3);</pre>	4	4	✓

Passed all tests! ✓

