

<b>Trạng thái</b>	Đã xong
<b>Bắt đầu vào lúc</b>	Thứ Tư, 29 tháng 5 2024, 5:14 PM
<b>Kết thúc lúc</b>	Thứ Tư, 29 tháng 5 2024, 5:37 PM
<b>Thời gian thực hiện</b>	23 phút 3 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 &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 | int isPrime(int n){
```

```

2      if(n <= 1) return false;
3      if(n % 2 == 0) return false;
4
5      for(int i = 3; i*i <= n; i+=2)
6      {
7          if(n % i == 0) return false;
8      }
9      return true;
10 }
11
12 int specialCells(int arr[][1000], int row, int col) {
13
14     int count = 0;
15     for(int i = 0; i < row; i++)
16     {
17         for(int j = 0; j < col; j++)
18         {
19             int sumcol = 0;
20             int sumrow = 0;
21             for(int k = 0; k < col; k++ )
22                 sumrow += arr[i][k];
23
24             for(int k = 0; k < row; k++)
25                 sumcol += arr[k][j];
26
27             if(isPrime(sumrow) && isPrime(sumcol))
28                 count++;
29         }
30     }
31     return count;
32 }

```



	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	✓

	Test	Expected	Got	
✓	<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  $M \times N$ .

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

**Answer:** (penalty regime: 0 %)

Reset answer

```
1 | bool isOdd(int n)
```

```

2 {
3     if(n%2 != 0) return true;
4     return false;
5 }
6
7 int subMatrix(int arr[][1000], int row, int col) {
8     int count = 0;
9     for(int i=0; i < row - 1; i++)
10    {
11        for(int j=0; j < col - 1; j++)
12        {
13            int sum = 0;
14            sum = arr[i][j] + arr[i+1][j] + arr[i][j+1] + arr[i+1][j+1];
15            if(isOdd(sum) == true)
16                count++;
17        }
18    }
19    return count;
20 }

```

	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! ✓



