

Question 1

Not answered

Mark 0.00 out of 10.00

[TwoDimensionArray]

Ở [bài trước](#), các bạn đã được thực hành với [mảng một chiều](#) trên ngôn ngữ C++. Bây giờ các bạn thử tưởng tượng nếu kiểu dữ liệu của [mảng một chiều](#) là [mảng một chiều](#). Hay nói cách khác, chúng ta có một mảng chứa các [mảng một chiều](#). Lúc này, chúng trở thành mảng hai chiều.

Cũng như [mảng một chiều](#), ta có thể khai báo mảng hai chiều bằng cách dùng mảng tĩnh hoặc vector như sau

```
// Mảng tĩnh
<tên kiểu phần tử> <tên mảng>ước [<số hàng>][ <số cột>];
```

```
// Vector
vector < vector < <tên kiểu phần tử> > > <tên mảng>(<số hàng>, vector < <tên kiểu phần tử> >(<số cột>));
```

Ví dụ, để lưu một ma trận số nguyên có kích thước 3×5 , ta khai báo như sau,

```
// Mảng tĩnh
int a[3][5];
```

```
// Vector
vector < vector<int> > a(3,vector<int>(5));
```

Các hàng và cột trong mảng hai chiều cũng được đánh số từ 0. Ví dụ, một mảng hai chiều 3×5 được tạo thành như sau.

	Cột1	Cột2	Cột3	Cột4	Cột5
Hàng1	[0][0]	[0][1]	[0][2]	[0][3]	[0][4]
Hàng2	[1][0]	[1][1]	[1][2]	[1][3]	[1][4]
Hàng3	[2][0]	[2][1]	[2][2]	[2][3]	[2][4]

Để đọc một mảng hai chiều 3×5 từ một đầu vào, ta có thể sử dụng hai [vòng lặp lồng nhau](#) như sau,

```
for (int i = 0; i < 3; ++i)
    for (int j = 0; j < 5; ++j)
        cin >> a[i][j];
```

Bài tập

Cho ma trận kích thước $m \times n$ chứa các số nguyên, các hàng được đánh số từ 1 đến m , các cột được đánh số từ 1 đến n . Có Q câu hỏi, mỗi câu hỏi đưa ra 2 số i, j . Nhiệm vụ của bạn là in ra giá trị của phần tử ở hàng thứ i và cột thứ j của ma trận.

Đầu vào

Đầu vào từ bàn phím gồm $m + Q + 1$ dòng.

- Dòng đầu tiên chứa 3 số nguyên m, n, Q ($m, n, Q \leq 500$).
- m dòng tiếp theo mỗi dòng chứa n số nguyên biểu diễn ma trận đã cho, các số liên tiếp trên một dòng cách nhau bởi một dấu cách.
- Q dòng tiếp theo, mỗi dòng là một câu hỏi chứa 2 số nguyên i, j cách nhau bởi một dấu cách.

Đầu ra

In ra màn hình Q dòng, mỗi dòng tương ứng với câu trả lời của một câu hỏi.

For example:

Input	Result
3 3 2	5
1 3 5	4
2 4 7	
3 4 2	
1 3	
2 2	