# Another Version of Inversion

DCE Coders admins are way much geekier than they actually seem! Kartik has been following that tradition lately. How? Well, he took the inversion count problem to a whole new level!

Sounds pretty normal to you, huh? Wanna challenge him? Try solving his version of inversion count then!

You are given a 2-d array of integers. You need to find out the inversion count of that array. A pair of integers in the 2-d array counts as an inversion pair (A,B) if and only if:

- There exists a valid path from top-left corner (0,0) to bottom right corner (r, c) such that A and B integers lie on that path.
- A occurs before B on that path.
- And, A > B.

A valid path is defined as a path that can be traversed from top-left corner (0, 0) to bottom-right corner (r, c) by moving only in right or downwards direction, without moving out of the grid.

Are you geekier than Kartik?

**Constraints:**

0 < R, C <= 300

0 < Ai <= 10^5, where Ai stands for an integer in the array.

## Input

First line contains space separated 2 integers, R and C, denoting the number of rows and columns.

Next R lines contain C space separated integers representing the 2-d array.

## Output

Output the number of inversion pairs as described in the problem statement.

## Example

**Input:**

```
4 4
3 4 2 5
1 7 11 16
8 9 6 12
10 13 15 14
```

**Output:**

```
10
```