**Monk and Inversions**

Attempted by: **1345**

/

Accuracy: **63%**

/

Maximum Score: **20**

/

3 Votes

Tag(s):

Data Structures, Easy

**PROBLEM**

**EDITORIAL**

**MY SUBMISSIONS**

**ANALYTICS**

Monk's best friend Micro, who happen to be an awesome programmer, got him an integer matrix MM of size N×NN×N for his birthday. Monk is taking coding classes from Micro. They have just completed array inversions and Monk was successful in writing a program to count the number of inversions in an array. Now, Micro has asked Monk to find out the number of inversion in the matrix MM. Number of inversions, in a matrix is defined as the number of unordered pairs of cells {(i,j),(p,q)}{(i,j),(p,q)} such that M[i][j]>M[p][q] & i≤p & j≤qM[i][j]>M[p][q] & i≤p & j≤q.  
Monk is facing a little trouble with this task and since you did not got him any birthday gift, you need to help him with this task.

**Input:**  
First line consists of a single integer TT denoting the number of test cases.  
First line of each test case consists of one integer denoting NN. Following NN lines consists of NN space separated integers denoting the matrix MM.

**Output:**  
Print the answer to each test case in a new line.

**Constraints:**  
1≤T≤1001≤T≤100  
1≤N≤201≤N≤20  
1≤M[i][j]≤10001≤M[i][j]≤1000

**SAMPLE INPUT**

2

3

1 2 3

4 5 6

7 8 9

2

4 3

1 4

**SAMPLE OUTPUT**

0

2

**Explanation**

In first test case there is no pair of cells (x1,y1)(x1,y1), (x2,y2)(x2,y2), x1≤x2 & y1≤y2x1≤x2 & y1≤y2 having M[x1][y1]>M[x2][y2]M[x1][y1]>M[x2][y2], so the answer is 00.  
In second test case M[1][1]>M[1][2]M[1][1]>M[1][2] and M[1][1]>M[2][1]M[1][1]>M[2][1], so the answer is 22.

**Time Limit:**1.0 sec(s) for each input file.

**Memory Limit:**256 MB

**Source Limit:**1024 KB

**Marking Scheme:**Marks are awarded when all the testcases pass.

**Allowed Languages:**C, C++, Clojure, C#, D, Erlang, F#, Go, Groovy, Haskell, Java, Java 8, JavaScript(Rhino), JavaScript(Node.js), Lisp, Lisp (SBCL), Lua, Objective-C, OCaml, Octave, Pascal, Perl, PHP, Python, Python 3, R(RScript), Racket, Ruby, Rust, Scala, Scala 2.11.8, Swift, Visual Basic

<https://www.hackerearth.com/practice/data-structures/arrays/multi-dimensional/practice-problems/algorithm/monk-and-inversions-arrays-strings/>

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ConsoleApplication1

{

class Program

{

static void Main(string[] args)

{

int t = int.Parse(Console.ReadLine());

while (t-- > 0)

{

int n = int.Parse(Console.ReadLine());

int[][] matriz = new int[n][];

for (int i = 0; i < n; i++)

{

matriz[i] = Array.ConvertAll(Console.ReadLine().Split(' '), e => int.Parse(e));

}

//int[,] matriz =

//{

// {1, 2, 3},

// {4, 5, 6},

// {7, 8, 9}

//};

int cont = 0;

for (int i = 0; i < matriz.Length; i++)

{

for (int j = 0; j < matriz.Length; j++)

{

for (int p = i; p < matriz.Length; p++)

{

for (int q = j; q < matriz.Length; q++)

{

if (matriz[i][j] > matriz[p][q])

{

cont++;

}

}

}

}

}

Console.WriteLine(cont);

}

Console.ReadLine();

}

}

}