Problem Statement:

Monk's best friend Micro, who happen to be an awesome programmer, got him an integer matrix M of size N×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 M. Number of inversions, in a matrix is defined as the number of unordered pairs of cells {(i,j),(p,q)} such that M[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

import java.util.\*;

class TestClass{

public static void main (String[] args)

{

Scanner sc = new Scanner(System.in);

int testcase = sc.nextInt();

while (testcase != 0){

int arrdy = sc.nextInt();

int[][] arr = new int[arrdy][arrdy];

for (int i = 0; i < arrdy ; i++){

for (int j = 0 ; j < arrdy ; j++){

arr[i][j] = sc.nextInt();

}

}

int inversions = 0;

for (int i = 0; i <arrdy; i++){

for (int l = 0 ; l < arrdy ; l++){

for (int j = 0 ; j <= i ; j++){

for (int k = 0 ; k <= l ; k++){

if (arr[i][l] < arr[j][k]){

inversions++;

}

}

}

}

}

System.out.println(inversions);

testcase--;

}

}

}