You are given a 2D matrix. Every diagonal in the matrix, including the main diagonal that runs from its top left corner to its bottom right corner and all of the diagonals running parallel to it, must be composed of same element. Each diagonal can contain a different element. A matrix is considered *good* if each of its diagonals meets these requirements.

**Example**

For matrix = [

[7, 3, 5, 1],

[5, 7, 3, 5],

[1, 5, 7, 3],

[2, 1, 5, 7]

], the output should be

isGoodMatrix(matrix) = true.

Let's collect numbers in each diagonal:

* 1
* 5 5
* 3 3 3
* 7 7 7 7
* 5 5 5
* 1 1
* 2

Each diagonal is composed of the same number, meaning that the matrix is *good* and the answer is true.

For matrix = [

[1, 2, 3, 4],

[0, 1, 4, 3],

[4, 0, 2, 2],

[4, 2, 0, 1]

], the output should be

isGoodMatrix(matrix) = false.

There are three diagonals in which every cell does not have the same element:

* 2 4 2
* 1 1 2 1
* 4 2

The matrix is not *good* and the answer is false.

* **[time limit] 3000ms (cs)**
* **[input] array.array.integer matrix**

1 ≤ matrix.length ≤ 10  
matrix[i].length = matrix.length  
-100 ≤ matrix[i][j] ≤ 100

* **[output] boolean**

<https://codefights.com/challenge/teS5pB7RPT6kJchgW?utm_source=featuredChallenge&utm_medium=email&utm_campaign=email_notification>

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ConsoleApplication1

{

class Program

{

static bool isGoodMatrix(int[][] matrix)

{

int n = matrix[0].Length;

int col\_ini = n - 1;

int fila\_ini = 0;

while (col\_ini >= 0)

{

int col = col\_ini, fila = fila\_ini;

int actual = matrix[fila][col];

while (col < n && fila < n)

{

// Console.Write(matrix[fila][col] + " ");

if (matrix[fila][col] != actual)

{

return false;

}

col++;

fila++;

}

col\_ini--;

// Console.WriteLine();

}

col\_ini = 0;

fila\_ini = 1;

while (fila\_ini < n)

{

int col = col\_ini, fila = fila\_ini;

int actual = matrix[fila][col];

while (col < n && fila < n)

{

// Console.Write(matrix[fila][col] + " ");

if (matrix[fila][col] != actual)

{

return false;

}

col++;

fila++;

}

fila\_ini++;

//Console.WriteLine();

}

return true;

}

static void Main(string[] args)

{

//int[][] matrix = {

// new int[] {7, 3, 5, 1},

// new int[] {5, 7, 3, 5},

// new int[] {1, 5, 7, 3},

// new int[] {2, 1, 5, 7}

// };

int[][] matrix = {

new int[]{1, 2, 3, 4},

new int[]{0, 1, 4, 3},

new int[]{ 4, 0, 2, 2},

new int[] {4, 2, 0, 1}

};

Console.WriteLine( isGoodMatrix(matrix));

Console.ReadLine();

}

}

}

SOLUTION BY CHUCKWILLWER

bool isGoodMatrix(int[][] m)

{

for (int i = 1; i < m.Length; i++)

for (int j = 1; j < m.Length; j++)

if (m[i][j] != m[i - 1][j - 1])

return false;

return true;

}