There are multiple rooks placed on a chessboard.  
A rook can take another rook if they both stand on the same row or on the same column.  
If no rook can take another rook, then we call it a *safe configuration*.  
Determine whether the given configuration is *safe* or not.

**Example:**

* For

chessBoard = [[false,true],

[true,false]]

the answer is true.

* For

chessBoard = [[false,true],

[false,true]]

the answer is false.

The configuration is not *safe*, because rooks on the second column can take each other.

* **[input] array.array.boolean chessBoard**
  + The current chessboard configuration on a rectangular chessboard. chessBoard[i][j] = true if there is a rook at the cell(i, j), and false otherwise.
* **[output] boolean**
  + true if the current configuration is*safe*, false otherwise.

<https://codefights.com/challenge/BMy2WLAxNSD7r6z2b>

static bool rooksOnChessBoard(bool[][] chessBoard)

{

bool inseguro = false;

//rocorro fila por fila

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

{

bool en\_fila = false;

for (int j = 0; j < chessBoard[i].Length; j++)

{

if (en\_fila)

{

if (chessBoard[i][j] == true)

{

inseguro = true;

}

}

if (chessBoard[i][j] == true)

{

en\_fila = true;

}

}

}

for (int j = 0; j < chessBoard[0].Length; j++)

{

bool en\_col = false;

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

{

if (en\_col)

{

if (chessBoard[i][j] == true)

{

inseguro = true;

}

}

if (chessBoard[i][j] == true)

{

en\_col = true;

}

}

}

return !inseguro;

}