Yesterday you found some shoes in your room. Each shoe is described by two values:

* *type* indicates if it's a left or a right shoe;
* *size* is the size of the shoe.

Your task is to check whether it is possible to pair the shoes you found in such a way that each pair consists of a right and a left shoe of an equal size.

**Example**

* For
* shoes = [[0, 21],
* [1, 23],
* [1, 21],
* [0, 23]]

the output should be  
pairOfShoes(shoes) = true;

* For
* shoes = [[0, 21],
* [1, 23],
* [1, 21],
* [1, 23]]

the output should be  
pairOfShoes(shoes) = false.

**Input/Output**

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

Array of shoes. Each shoe is given in the format [*type*, *size*], where *type* is either 0 or 1 for left and right respectively, and *size* is a positive integer.

*Constraints:*  
1 ≤ shoes.length ≤ 25,  
1 ≤ shoes[i][1] ≤ 100.

* **[output] boolean**

true if it is possible to pair the shoes, falseotherwise.

<https://codefights.com/arcade/code-arcade/spring-of-integration/picP24ieQnuqR4kxJ>

static bool pairOfShoes(int[][] shoes)

{

var frec = new Dictionary<int, int>();

/\*la idea es asociar a cada size la cantidad de veces

que aparece un uno, y la cantidad de veces que aparece un cero

si la cantidad de unos y de ceros es la misma, entonces la

cantidad frec[size] es cero, que significa que no hay ningun

zapato de ese tamaño suelto\*/

/\*cargo todo el diccionario con a cada size con cero\*/

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

{

frec[ shoes[i][1]]=0;

}

/\*cuento cuantos unos y ceros hay, (de type) \*/

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

{

if (shoes[i][0] == 0) frec[shoes[i][1]]--;

else frec[shoes[i][1]]++;

}

/\*si hay algun value distinto de cero es porque hay algun zapato

\* que no pertenece a ningun par \*/

foreach (KeyValuePair<int, int> kvp in frec)

{

if (kvp.Value != 0)

{

return false;

}

}

return true;

}