Given two version strings composed of several non-negative decimal fields separated by periods ("."), both strings contain equal number of numeric fields. Return true if the first version is higher than the second version and false otherwise.

The syntax follows the regular *semver* ordering rules:

1.0.5 < 1.1.0 < 1.1.5 < 1.1.10 < 1.2.0 < 1.2.2

< 1.2.10 < 1.10.2 < 2.0.0 < 10.0.0

There are no leading zeros in any of the numeric fields, i.e. you do not have to handle inputs like 100.020.003(it would instead be given as 100.20.3).

**Example**

* For ver1 = "1.2.2" and ver2 = "1.2.0", the output should be  
  higherVersion(ver1, ver2) = true;
* For ver1 = "1.0.5" and ver2 = "1.1.0", the output should be  
  higherVersion(ver1, ver2) = false.

**Input/Output**

* **[time limit] 3000ms (cs)**
* **[input] string ver1**

*Constraints:*  
1 ≤ ver1.length ≤ 14.

* **[input] string ver2**

*Constraints:*  
1 ≤ ver2.length ≤ 14.

* **[output] boolean**

<https://codefights.com/arcade/code-arcade/lab-of-transformations/vsKRjYKv4SCjzJc8r>

static bool higherVersion(string ver1, string ver2)

{

int[] v1 = Array.ConvertAll(ver1.Split('.'), f => int.Parse(f));

int[] v2 = Array.ConvertAll(ver2.Split('.'), f => int.Parse(f));

for (int i = 0; i < v1.Length; i++)

{

if (v1[i] > v2[i])

{

return true;

}

else if(v2[i] > v1[i])

{

return false;

}

}

return false;

}