

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

- [execution time limit] 4 seconds (js)
- [input] string ver1

Guaranteed constraints:
`1 ≤ ver1.length ≤ 15 .`

- [input] string ver2

Guaranteed constraints:
`1 ≤ ver2.length ≤ 15 .`

- [output] boolean

[JavaScript (ES6)] Syntax Tips

```
// Prints help message to the console
// Returns a string
function helloWorld(name) {
    console.log("This prints to the console when you Run Tests");
    return "Hello, " + name;
}
```