**Functions**

**DEADLINE:** 03/05/2020

**FOLDER STRUCTURE**

|  |  |
| --- | --- |
| FL13\_HW8/\*     └─ task/      └─ FL13\_HW8.docx  └─ homework/\*     └─ js/\*  └─ isBigger.js\*  └─ stringToNumber.js\*  └─ storeNames.js\*  └─ getDifference.js\*  └─ positiveSum.js\*  └─ letterCount.js\*  └─ countPoints.js\*  └─ .eslintrc.js | \* ­­­- required |

**TASK**

**Task #1**

Write a function - *isBigger*

It should accept two arguments and returns **true** if first one has **greater** value than second one or false otherwise.

**Tip**: no need for if/else clause nor ternary operator  
**For example**:

isBigger(5, -1) // => true

**Task #2**

Write a function - *stringToNumber*

It should accept one argument as a string and return it as a number

**Tip**: Don’t worry about incoming string – it’s always valid

**For example**:

stringToNumber('1258') // => 1258

**Task #3**

Write a function - *storeNames*

It should accept an **arbitrary** number of strings and return an array of that strings

**For example**:

storeNames('Nick Fury', 'Iron Man', 'Doctor Strange')

// => ['Nick Fury', 'Iron Man', 'Doctor Strange']

**Task #4**

Write a function - *getDifference*

It should accept two arguments as numbers and return their difference. But the function *never returns a negative value*. If second parameter is greater than first one, function will change their order.

**Tip**: consider reusing *isBigger* function

For example:

getDifference(5, 3) // => 2

getDifference(5, 8) // => 3

**Task #5**

Write a function - *positiveSum*

It should accept an array of numbers and return a result of their addition. But you must calculate *only positive* numbers and omit negative if any presents.

**For example**:

positiveSum([2, 4, 6, 8])   // => 20

positiveSum([0, -3, 5, 7])  // => 12

**Task #6**

Write a function – *letterCount*

It accepts two string arguments and returns an integer of the count of occurrences the 2nd argument is found in the first one.

If no occurrences can be found, a count of 0 should be returned.  
**For example**:

letterCount("Maggy", "g") // => 2

letterCount("Barry", "b") // => 1

letterCount("", "z")  // => 0

**Task #7**

Our football team completed the championship. The result of each match look like **"x:y"**.

Results of all matches are recorded in the collection like this: **["3:1", "2:2", "0:1", ...]**

Write a function – *countPoints*

It should accept a collection of football games scores and count the points of our team in the championship.

Rules for counting points for each match:

* if x > y - 3 points
* if x < y - 0 point
* if x = y - 1 point

**Tip:** there are 10 matches in the championship

* 0 <= x <= 4
* 0 <= y <= 4
* Consider reusing of *isBigger* function

**For example:**

countPoints(['3:1', '1:0', '0:0', '1:2', '4:0', '2:3', '1:1', '0:1', '2:1', '1:0']) // => 17

countPoints(['1:1', '1:2', '2:0', '4:2', '0:1', '2:3', '1:1', '0:1', '1:1', '3:0']) // => 12

## RESTRICTIONS

* Usage of **Math object** is forbidden;

**BEFORE SUBMIT**

* Verify that all functionality is implemented according to requirements;
* Format your code (remove redundant spaces, lines of code etc.);
* Validate code via eslint;
* Add comments if necessary, delete non-relevant comments;
* In order to use npm you should install nodejs (https://nodejs.org/ );
* Install eslint to check your code (npm install -g eslint);
* open a terminal (or cmd);
* go to src folder;
* run eslint;
* Code should be without ‘errors’;

**SUBMIT**

* The folder should be uploaded to github repository '**FL13**' into **master** branch