JavaScript Unit Testing

DEADLINE: 16/02/2022

FOLDER STRUCTURE

```
FL18_HW11/ *                                          * - required
    ├── task/
    ├── homework/ *
        ├── spec/ *
            ├── index.spec.js
            ├── get-bigest-number.spec.js *
            ├── birthday.service.spec.js *
        ├── src/ *
            ├── index.js
            ├── get-bigest-number.js *
            ├── birthday.service.js *
        ├── .gitignore *
        ├── jasmine.js *
        ├── karma.conf.js *
        ├── package.json *
        ├── .eslintrc.js *
```

DESCRIPTION

Your task is:

- implement functionality described in the Task section
- cover your code by unit tests using *Jasmine* framework

TASK

1. Function **getBigestNumber()**:
    - accepts [2-10] arguments
    - returns the biggest value
    - if at least one argument's type is not a number throw Error *'Wrong argument type'*
    - if amount of arguments is not respective throw Error *'Not enough arguments'* or `*Too many arguments'*

2. Class or plain object **BirthdayService**
    - **howLongToMyBirthday(date)** method, accepts date/timestamp
    - asynchronously compute time left to the expectant date (should return Promise or thenable, resolve/reject after 100ms)
    - in case of wrong argument – throw Error `Wrong argument!`
    - if it is today – call method congratulateWithBirthday (just simple console log with 'Hooray!!! It is today!' would be enough)
    - if it will be in the next half year call function notifyWaitingTime(waitingTime) which inform a user (just console log) 'Soon...Please, wait just xxx day/days'
    - if it has already happened less than 6 month ago notify by calling
    - notifyWaitingTime(time) 'Oh, you have celebrated it xxxx day/s ago, don't you remember?'

    NIT: You might implement your own logger method and use it instead of console.log

HOW TO

Setup env

- In order to use npm package manager you should install Node.js (https://nodejs.org/)
- go to the homework folder
- run **npm install** (or **npm i**) command
- **package.json** has two commands to run tests under the script section:
    - **test:jasmine** – run tests in the Node.js environment
    - **test:karma** – run tests in the Chrome Browser (includes code coverage)
- to run any kind of test just execute **npm run test:jasmine** or **npm run test:karma** (prefer the second command)

TIP: npm run test:karma runs tests in the *watch* mode (no need to execute command over and over after any changes in the code) and generates **coverage** folder as well with respective information about you code coverage. Go to that folder and open index.html file from the root, verify how many lines of code still need to be covered.

Please, note: running **npm run test:jasmine** will execute tests in the Node.js environment, it's slightly different from the browser (Chrome in your case) env, there are not such entities like window object, fetch method, xhr, storages etc.

IMPORTANT: Currently testing environment configuration allows using of CommonJS modules only whereas ES6 imports doesn't work. (find in the code use case example)

RESTRICTIONS

- Adding **task/** folder is forbidden. Do not push it to repository. (Only **homework/** folder should be pushed)
- Do not use any external libraries

BEFORE SUBMIT

- Code should be clean, without comments, readable, and tested

- Make sure your GitLab folder structure meets folder structure from this document (without **task** folder)
- Check your Javscript tasks with Eslint

SUBMIT

- The **FL18_HW11** folder without **task** folder should be uploaded to GitLab repository "**FL-18**" into **main** branch.

USEFUL LINKS

- https://jasmine.github.io/index.html - Jasmine documentation
- https://nodejs.org/api/modules.html - CommonJS modules
- https://developer.mozilla.org/ru/docs/Web/JavaScript/Guide/Modules - JavaScript modules