JS Advanced Regular Exam – 19 Feb 2022

3. Flower Shop

Your Task

Using **Mocha** and **Chai** write **JS Unit Tests** to test a variable named **flowerShop**, which represents an object. You may use the following code as a template:

```
describe("Tests …", function() {
    describe("TODO …", function() {
        it("TODO …", function() {
            // TODO: …
        });
     });
     // TODO: …
});
```

The object that should have the following functionality:

- **`calcPriceOfFlowers(flower, price, quantity)`** - A function that accepts three parameters (one string and two numbers):
    - The function returns the multiplies price and quantity;
    - There is a need for validation of the input, a flower, a price and a quantity may not always be valid. In case of submitted invalid parameters, **throw** an error **`"Invalid input!"`;**
    - The result is rounded to the second digits after the decimal point.
- **`checkFlowersAvailable(flower, gardenArr)`** - A function that accepts two parameters (string and array):
    - There is no need for validation for input, you will always be given a string and an array;
    - The array (**`gardenArr`**) includes all available flowers (example: `[`**`"Rose", "Lily", "Orchid"`**`]`);
    - The function checks whether the submitted string **`flower`** is present in the array **`gardenArr`**;
    - If present in the array, the function **return:** **`` `The ${flower} are available!` ``**;
    - Otherwise, the function **return:** **`` `The ${flower} are sold! You need to purchase more!` ``**.
- **`sellFlowers(gardenArr, space)`** - A function that accepts two parameters (array and number):
    - The **`gardenArr`** array will store the names of flowers`(["`**`Rose`**`", "`**`Lily`**`", "`**`Orchid`**`"])`;
    - You must **remove** an **element** from the array that is located on the **`space`** specified as a parameter;
    - There is a need for validation for the input, an array and space may not always be valid. In case of submitted **invalid** parameters, **throw** an error **`"Invalid input!"`**;
    - Finally, **return** the changed array of flowers as a string, joined by **`" / "`**.

JS Code

To ease you in the process, you are provided with an implementation that meets all of the specification requirements for the **flowerShop** object:

flowerShop.js

```
const flowerShop = {
     calcPriceOfFlowers(flower, price, quantity) {
          if (typeof flower != 'string' || !Number.isInteger(price) ||
!Number.isInteger(quantity)) {
              throw new Error('Invalid input!');
          } else {
              let result = price * quantity;
              return `You need $${result.toFixed(2)} to buy ${flower}!`;
        }
    }
     checkFlowersAvailable(flower, gardenArr) {
        if (gardenArr.indexOf(flower) >= 0) {
            return `The ${flower} are available!`;
        } else {
            return `The ${flower} are sold! You need to purchase more!`;
        }
    },
     sellFlowers(gardenArr, space) {
        let shop = [];
        let i = 0;
        if (!Array.isArray(gardenArr) || !Number.isInteger(space) || space < 0 ||
space >= gardenArr.length) {
            throw new Error('Invalid input!');
        } else {
            while (gardenArr.length > i) {
                if (i != space) {
                    shop.push(gardenArr[i]);
                }
                i++
            }
        }
        return shop.join(' / ');
    }
}
```

Submission

Submit your tests inside a **`describe()`** statement, as shown above.