

3. Number Operations

Your Task

Using **Mocha** and **Chai** write **JS Unit Tests** to test a variable named **numberOperations**, 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:

- **powNumber(num)** - A function that accepts a single parameter:
 - the function returns the power of the given number
 - there is no need of validation for the input
- **numberChecker(input)** - A function that accepts a single parameter:
 - the function parses the input to number, and validates it
 - If the input is not a number the function throws an error
 - if the input is a number, the function checks if it is lower than 100. If so the function returns the string: "The number is lower than 100!"
 - otherwise the function returns: "The number is greater or equal to 100!"
- **sumArrays(array1, array2)** - A function that should accepts two parameters (arrays):
 - the function loops through the arrays and sums the number on the first index in the first array with the number on the first index of the second array, then sums the number on the second index of the two arrays and so on.
 - The function returns new array, which represents the sum of the two arrays by indexes
 - The arrays will be always valid, there is no need to test the input arrays.

JS Code

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

numberOperations.js

```
Const numberOperations = {
  powNumber: function (num) {
    return num * num;
  },
  numberChecker: function (input) {
    input = Number(input);

    if (isNaN(input)) {
```

```

        throw new Error('The input is not a number!');
    }

    If (input < 100) {
        Return 'The number is lower than 100!';
    } else {
        Return 'The number is greater or equal to 100!';
    }
},
sumArrays: function (array1, array2) {

    const longerArr = array1.length > array2.length ? array1 : array2;
    const rounds = array1.length < array2.length ? array1.length : array2.length;

    const resultArr = [];

    for (let i = 0; i < rounds; i++) {
        resultArr.push(array1[i] + array2[i]);
    }

    resultArr.push(...longerArr.slice(rounds));

    return resultArr
}
};

```

Submission

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