

Compte rendu TP1 - vitest

Wiem ben mlouka - eya raouine – fatma hamada (GL4)

Lien repo Github : <https://github.com/wiem2000/unit-testing-vitest>

Activité 1

File Edit Selection View ... tp1-vitest-js

EXPLORER ...

- OPEN EDITORS
 - GROUP 1
 - JS math.js src
 - GROUP 2
 - Welcome
 - package.json
 - JS math.test.js tests
- TP1-VITEST-JS
 - node_modules
 - src
 - JS math.js
 - tests
 - JS math.test.js
 - package-lock.json
 - package.json

src > JS math.js > add

```
1 export function add(a, b) {
2   return a + b;
3 }
```

tests > JS math.test.js > ...

```
1 import { test, expect } from 'vitest';
2 import { add } from '../src/math';
3
4 test('Test de la fonction add', () => {
5   const result = add(2, 3);
6   expect(result).toBe(5);
7 });
```

PROBLEMS OUTPUT TERMINAL PORTS DEBUG CONSOLE

node + -

press h to show help, press q to quit

PS C:\Users\MSI\Desktop\insat\GL4\2 eme semestre\test\tp1-vitest-js> npm run test

> test

> vitest

RERUN tests/math.test.js x1

✓ tests/math.test.js (1)

✓ Test de la fonction add

Test Files 1 passed (1)

Tests 1 passed (1)

Start at 10:45:24

Duration 21ms

PASS Waiting for file changes...

press h to show help, press q to quit

> OUTLINE

> TIMELINE

Exercice 1

```
JS ex4-async.test.js {} package.json JS ex1-stringManipulation.test.js X JS ex3-pricebo ... boundary.js JS ex1-stringManipulation.js X JS ex4-async.js

tests > JS ex1-stringManipulation.test.js > ...
1
2 import { test, expect, } from 'vitest';
3 import { transformer } from '../src/ex1-stringManipulation';
4
5
6 //it and test: Both it and test keywords are used to define individual
7
8 test('Transforme une chaîne vide', () => {
9   const result = transformer('');
10   expect(result).toBe('');
11   // toBe pour vérifier si le résultat est une chaîne vide
12 });
13
14
15 test('Transforme une chaîne en minuscules en majuscules', () => {
16   const result = transformer('hello');
17   expect(result).toEqual('HELLO');
18   //toEqual pour vérifier si le résultat est égal à 'HELLO'
19 });
20
21
22 test('Transforme une chaîne avec des caractères spéciaux en majuscules',
23   const result = transformer('Hello,@ world!');
24   expect(result).toMatch('HELLO,@ WORLD!');
25   // toMatch pour vérifier si le résultat correspond à 'HELLO,@ WORLD!'
26 });
27
28
29
```

```
src > JS ex1-stringManipulation.js > ...
1 export function transformer(inputString) {
2   return inputString.toUpperCase();
3 }
4
5
```

press h to show help, press q to quit

○ PS C:\Users\MSI\Desktop\insat\GL4\2 eme semestre\test\tp1-vitest-js> npm run test

> test

> vitest

DEV v1.5.0 C:/Users/MSI/Desktop/insat/GL4/2 eme semestre/test/tp1-vitest-js

✓ tests/ex1-stringManipulation.test.js (1)

✓ tests/math.test.js (1)

Test Files **2 passed** (2)

Tests **2 passed** (2)

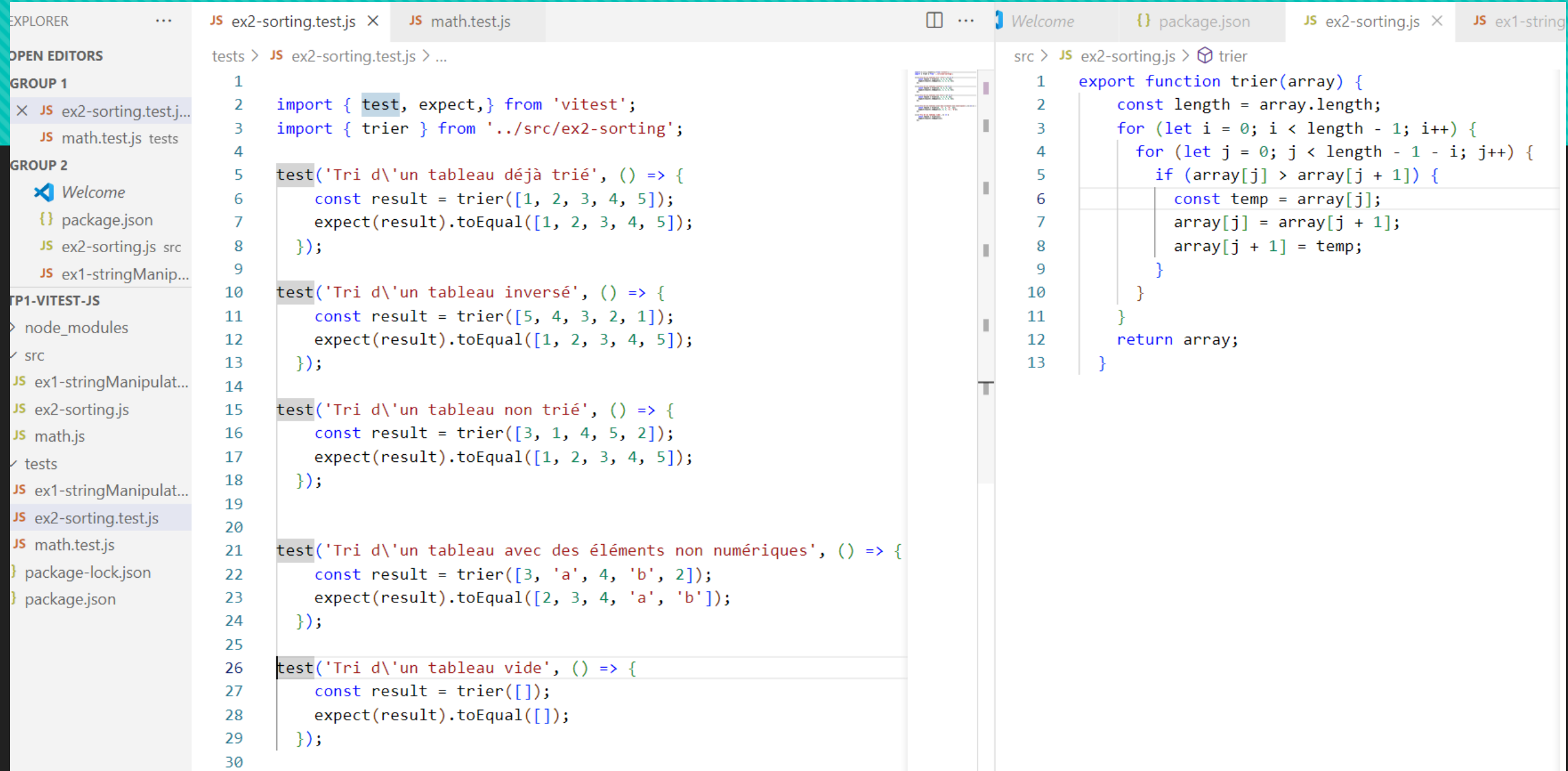
Start at 11:08:30

Duration 423ms (transform 34ms, setup 0ms, collect 46ms, tests 5ms, environment 0ms, prepare 309ms)

PASS Waiting for file changes...

press h to show help, press q to quit

Exercice 2



The image shows a VS Code editor with two open files. The left pane displays `ex2-sorting.test.js` with the following content:

```
1
2 import { test, expect, } from 'vitest';
3 import { trier } from '../src/ex2-sorting';
4
5 test('Tri d\'un tableau déjà trié', () => {
6   const result = trier([1, 2, 3, 4, 5]);
7   expect(result).toEqual([1, 2, 3, 4, 5]);
8 });
9
10 test('Tri d\'un tableau inversé', () => {
11   const result = trier([5, 4, 3, 2, 1]);
12   expect(result).toEqual([1, 2, 3, 4, 5]);
13 });
14
15 test('Tri d\'un tableau non trié', () => {
16   const result = trier([3, 1, 4, 5, 2]);
17   expect(result).toEqual([1, 2, 3, 4, 5]);
18 });
19
20 test('Tri d\'un tableau avec des éléments non numériques', () => {
21   const result = trier([3, 'a', 4, 'b', 2]);
22   expect(result).toEqual([2, 3, 4, 'a', 'b']);
23 });
24
25 test('Tri d\'un tableau vide', () => {
26   const result = trier([]);
27   expect(result).toEqual([]);
28 });
29
30
```

The right pane displays `ex2-sorting.js` with the following content:

```
1 export function trier(array) {
2   const length = array.length;
3   for (let i = 0; i < length - 1; i++) {
4     for (let j = 0; j < length - 1 - i; j++) {
5       if (array[j] > array[j + 1]) {
6         const temp = array[j];
7         array[j] = array[j + 1];
8         array[j + 1] = temp;
9       }
10    }
11  }
12  return array;
13 }
```

○ Avec erreur

PROBLEMS OUTPUT TERMINAL PORTS DEBUG CONSOLE

```
> tests/ex2-sorting.test.js (5)
  ✓ Tri d'un tableau déjà trié
  ✓ Tri d'un tableau inversé
  ✓ Tri d'un tableau non trié
  ✗ Tri d'un tableau avec des éléments non numériques
  ✓ Tri d'un tableau vide
✓ tests/math.test.js (1)
✓ tests/ex1-stringManipulation.test.js (1)
```

Failed Tests 1

FAIL tests/ex2-sorting.test.js > Tri d'un tableau avec des éléments non numériques
AssertionError: expected [3, 'a', 4, 'b', 2] to deeply equal [2, 3, 4, 'a', 'b']

- Expected
+ Received

```
Array [
-   2,
-   3,
-   4,
-   "a",
+   4,
+   "b",
+   2,
]
```

○ Sans erreur

```
JS ex2-sorting.test.js x JS math.test.js ... Welcome {} package.json JS ex2-sorting.js x JS ex1-stringManipulation.test.js

tests > JS ex2-sorting.test.js > test('Tri d\'un tableau avec des éléments non numériques', () => {
5   test('Tri d\'un tableau déjà trié', () => {
6     const result = trier([1, 2, 3, 4, 5]);
7     expect(result).toEqual([1, 2, 3, 4, 5]);
8   });
9
10  test('Tri d\'un tableau inversé', () => {
11    const result = trier([5, 4, 3, 2, 1]);
12    expect(result).toEqual([1, 2, 3, 4, 5]);
13  });
14
15  test('Tri d\'un tableau non trié', () => {
16    const result = trier([3, 1, 4, 5, 2]);
17    expect(result).toEqual([1, 2, 3, 4, 5]);
18  });
19
20
21  test('Tri d\'un tableau avec des éléments non numériques', () => {
22    const result = trier([3, 'a', 4, 'b', 2]);
23    expect(result).toEqual([2, 3, 4]);
24  });
25
26  test('Tri d\'un tableau vide', () => {
27    const result = trier([]);
28    expect(result).toEqual([]);
29  });
30
src > JS ex2-sorting.js > ...
1
2  function trierNotCorrected(array) {
3    const length = array.length;
4    for (let i = 0; i < length - 1; i++) {
5      for (let j = 0; j < length - 1 - i; j++) {
6        if (array[j] > array[j + 1]) {
7          const temp = array[j];
8          array[j] = array[j + 1];
9          array[j + 1] = temp;
10       }
11     }
12   }
13   return array;
14 }
15
16 export function trier(array) {
17
18   // Filtrer uniquement les nombres
19   const numbers = array.filter(item => typeof item === 'number');
20
21   // Trier les nombres
22   numbers.sort((a, b) => a - b);
23
24   return numbers;
25
26 }
```

RERUN src/ex2-sorting.js x43

✓ tests/ex2-sorting.test.js (5)
✓ Tri d'un tableau déjà trié
✓ Tri d'un tableau inversé
✓ Tri d'un tableau non trié
✓ Tri d'un tableau avec des éléments non numériques
✓ Tri d'un tableau vide

Test Files 1 passed (1)
Tests 5 passed (5)
Start at 12:13:27
Duration 24ms

Exercice 3

⦿ Avec erreur

```
JS ex2-sorting.test.js JS math.test.js JS ex3-priceboundary.test.js X ... {} package.json JS ex2-sorting.js JS ex3-priceboundary.js X

tests > JS ex3-priceboundary.test.js > ...
1 import { isPriceInRange } from '../src/ex3-priceboundary';
2 import { test, expect, } from 'vitest';
3
4 test('Prix égal à la limite inférieure', () => {
5   expect(isPriceInRange(100, 100, 200)).toBe(true);
6 });
7
8 test('Prix égal à la limite supérieure', () => {
9   expect(isPriceInRange(200, 100, 200)).toBe(true);
10 });
11
12 test('Prix en dessous de la limite inférieure', () => {
13   expect(isPriceInRange(99, 100, 200)).toBe(false);
14 });
15
16 test('Prix au-dessus de la limite supérieure', () => {
17   expect(isPriceInRange(201, 100, 200)).toBe(false);
18 });
19
20 test('Prix égal à la limite inférieure avec plage inversée',
21   expect(isPriceInRange(200, 200, 100)).toBe(true);
22 );
23
24 test('Prix égal à la limite supérieure avec plage inversée',
25   expect(isPriceInRange(100, 200, 100)).toBe(true);
26 );
27
```

src > JS ex3-priceboundary.js > isPriceInRange

```
1
2
3 export function isPriceInRange(price, min, max) {
4   return price >= min && price <= max;
5 }
```

PROBLEMS OUTPUT TERMINAL PORTS DEBUG CONSOLE

RERUN tests/ex3-priceboundary.test.js x6

```
> tests/ex3-priceboundary.test.js (6)
✓ Prix égal à la limite inférieure
✓ Prix égal à la limite supérieure
✓ Prix en dessous de la limite inférieure
✓ Prix au-dessus de la limite supérieure
✗ Prix égal à la limite inférieure avec plage inversée
✗ Prix égal à la limite supérieure avec plage inversée
```

Failed Tests 2

FAIL tests/ex3-priceboundary.test.js > Prix égal à la limite inférieure avec plage inversée
AssertionError: expected false to be true // Object.is equality

- Expected
+ Received

- true
+ false

🕒 Fonction corrigée

The screenshot shows a VS Code editor with several tabs: `JS ex2-sorting.test.js`, `JS math.test.js`, `JS ex3-priceboundary.test.js`, `package.json`, `JS ex2-sorting.js`, and `JS ex3-priceboundary.js`. The `ex3-priceboundary.test.js` file is active, showing a series of tests for the `isPriceInRange` function. The tests cover boundary conditions and a range inversion case. The `ex3-priceboundary.js` file is also visible, showing the implementation of `isPriceInRange` which handles the range inversion by swapping `min` and `max` if `min > max`.

```
tests > JS ex3-priceboundary.test.js > ...
1  import { isPriceInRange } from '../src/ex3-priceboundary';
2  import { test, expect, } from 'vitest';
3
4  test('Prix égal à la limite inférieure', () => {
5    expect(isPriceInRange(100, 100, 200)).toBe(true);
6  });
7
8  test('Prix égal à la limite supérieure', () => {
9    expect(isPriceInRange(200, 100, 200)).toBe(true);
10   });
11
12  test('Prix en dessous de la limite inférieure', () => {
13    expect(isPriceInRange(99, 100, 200)).toBe(false);
14  });
15
16  test('Prix au-dessus de la limite supérieure', () => {
```

```
src > JS ex3-priceboundary.js > isPriceInRangeNotCorrected
6
7  export function isPriceInRange(price, min, max) {
8    // Vérifie si la plage est inversée
9    if (min > max) {
10      // Si la plage est inversée, inverse les limites
11      const temp = min;
12      min = max;
13      max = temp;
14    }
15
16    // Vérifie si le prix est dans la plage entre min et max
17    return price >= min && price <= max;
18  }
19
```

PROBLEMS OUTPUT TERMINAL PORTS DEBUG CONSOLE

RERUN src/ex3-priceboundary.js x23

- ✓ tests/ex3-priceboundary.test.js (6)
 - ✓ Prix égal à la limite inférieure
 - ✓ Prix égal à la limite supérieure
 - ✓ Prix en dessous de la limite inférieure
 - ✓ Prix en dessous de la limite inférieure
 - ✓ Prix au-dessus de la limite supérieure
 - ✓ Prix égal à la limite inférieure avec plage inversée
 - ✓ Prix égal à la limite supérieure avec plage inversée

Test Files **1 passed** (1)
Tests **6 passed** (6)
Start at 12:38:53
Duration 24ms

Exercice 4

2-sorting.test.js

JS math.test.js

JS ex4-async.test.js X

JS ex3-pricebc

...

{ } package.json

JS ex2-sorting.js

JS ex3-priceboundary.js

JS ex4-async.js X

tests > JS ex4-async.test.js > test('Test de fetchRandomUser') callback

```
1 import { test, expect } from 'vitest';
2 import { fetchRandomUser } from '../src/ex4-async';
3
4 test('Test de fetchRandomUser', async () => {
5   // Attendre le resultat de la promesse retournée
6   // par fetchRandomUser
7   const userData = await fetchRandomUser();
8
9   // Vérifier que les données renvoyées sont bien un
10  // objet JSON
11  expect(typeof userData).toBe('object');
12  // Vérifier que l'objet contient les clés attendues
13  expect(userData).toHaveProperty('gender');
14  expect(userData).toHaveProperty('name');
15  expect(userData).toHaveProperty('location');
16  expect(userData).toHaveProperty('email');
17
18 });
```

src > JS ex4-async.js > fetchRandomUser

```
1
2
3 export async function fetchRandomUser() {
4   const response = await fetch('https://randomuser.me/api/');
5   if (!response.ok) {
6     throw new Error('Failed to fetch random user');
7   }
8   const data = await response.json();
9   return data.results[0]; // Renvoie les informations sur le
10 }
```

PROBLEMS

OUTPUT

TERMINAL

PORTS

DEBUG CONSOLE

RERUN

 tests/ex4-async.test.js x15

- tests/ex4-async.test.js (1)
 - Test de fetchRandomUser
- ✓ tests/ex4-async.test.js (1) 769ms
 - ✓ Test de fetchRandomUser 767ms

Test Files

 1 passed (1)

Tests

 1 passed (1)

Start at

 13:08:42

Duration

 783ms

PASS

 Waiting for file changes...

Couverture

package.json > {} devDependencies

1 {

2 "name": "tp1",

3 "scripts": {

4 "dev": "vite",

5 "build": "vite build",

6 "preview": "vite preview",

7 "test:coverage": "vitest --coverage"

8 },

9 "devDependencies": {

10 "@vitest/coverage-v8": "^1.5.0",

PROBLEMS OUTPUT TERMINAL PORTS DEBUG CONSOLE

✓ tests/ex4-async.test.js (1) 463ms
✓ tests/ex1-stringManipulation.test.js (3)
✓ tests/ex3-priceboundary.test.js (6)
✓ tests/ex2-sorting.test.js (5)
✓ tests/math.test.js (1)

Test Files **5 passed** (5)
 Tests **16 passed** (16)
 Start at 13:19:08
 Duration 899ms (transform 82ms, setup 1ms, collect 156ms, tests 475ms, environment 1ms, p

% Coverage report from v8

File	% Stmts	% Branch	% Funcs	% Lines	Uncovered Line #s
All files	73.77	91.66	71.42	73.77	
ex1-stringManipulation.js	100	100	100	100	
ex2-sorting.js	55.55	100	50	55.55	3-14
ex3-priceboundary.js	88.88	100	50	88.88	4-5
ex4-async.js	80	50	100	80	6-7
math.js	100	100	100	100	