

AVISO: Libere todo el potencial de su cuenta. ¡Confirme su correo electrónico para habilitar la experiencia dojo.com.



54

8 kyu Potenciación

☆ 21 🏆 dieciséis 🏆 90% de 905 🏆 4.346 de 4.345 🏆 JapaX

⚠ 2 problemas reportados

Instrucciones

Producción

✓ Test Passed: Value == 248832

▼ Log

6^5 => ... ?

✓ Test Passed: Value == 7776

▼ Log

1^7 => ... ?

✓ Test Passed: Value == 1

▼ Log

9^4 => ... ?

✓ Test Passed: Value == 6561

▼ Log

5^1 => ... ?

✓ Test Passed: Value == 5

▼ Log

12^5 => ... ?

✓ Test Passed: Value == 248832

▼ Log

6^12 => ... ?

js javascript

Nodo v8.1.3



EMPUJE EMACS



Solución

```
1 function power(x, y) {
2     if (y === 0) {
3         return 1;
4     } else {
5         let result = 1;
6         for (let i = 1; i <= y; i++) {
7             result *= x;
8         }
9         return result;
10    }
11}
12
13 // Pruebas
14 describe("Tests", function() {
15     it("should return 1 when x is raised to 701270", function() {
16         Test.assertEquals(power(1, 701270), 1);
17     });
18});
```

Pruebas de muestra

```
1 describe("Tests", function(){
2
3     Test.assertEquals(power(1,701270), 1);
4     Test.assertEquals(power(2,2), 4);
5     Test.assertEquals(power(3,2), 9);
6     Test.assertEquals(power(-1,40), 1);
7
8});
```

▶ SALTAR

🔒 DESBLOQUEAR SOLUCIONES

💬 DISCUTIR (14)

REINICIAR

PRUEBA

ENTREGAR



Entrenamiento de Kata



56

¡Aviso! Libere todo el potencial de su cuenta: ¡confirme su correo electrónico para habilitar la experiencia dojo completa!**8 kyu** Dólares y centavos

☆ 265 ⛳ 106 🏃 92% de 2.588 ⌂ 10.640 de 22.590

kolohelios

⚠ 2 problemas reportados

Instrucciones

Producción

Time: 633ms Passed: 43 Failed: 0

Test Results:

▼ Tests

> test

Completed in 4ms

▼ Fixed tests

✓ Test Passed: Value == '\$39.99'

✓ Test Passed: Value == '\$3.00'

✓ Test Passed: Value == '\$3.10'

✓ Test Passed: Value == '\$314.16'

▼ Random tests

> Testing 72

> Testing 175

> Testing 189

> Testing 262

javascript

Node v8.1.3

EMPUJE EMACS

**Solución**

```
1 function formatMoney(amount) {  
2     const formattedAmount = '$' + amount.toFixed(2);  
3     return formattedAmount;  
4 }  
5  
6 // Prueba  
7 describe("Tests", () => {  
8     it("test", () => {  
9         Test.assertEquals(formatMoney(39.99), '$39.99', 'That\'s not formatted the way we expected.  
10    });  
11});  
12
```

✓ ¡Buen trabajo! Puede tomarse su tiempo para refactorizar/comentar su solución. Enviar cuando esté listo.**Pruebas de muestra**

```
1 describe("Tests", () => {  
2     it("test", () => {  
3         Test.assertEquals(formatMoney(39.99), '$39.99', 'That\'s not formatted the way we expected.  
4     });  
5});  
6
```




Heads Up! Unlock the full potential of your account - confirm your email to enable the full dojo experience!



7 kyu

58

8 kyu Flick Switch

☆ 129 🏆 22 🏆 92% of 650 🏆 3,406 of 7,906

saudiGuy

Instructions

Output

Time: 915ms Passed: 51 Failed: 0

Test Results:

Fixed tests

Basic tests

Completed in 2ms

Random tests

> Testing for flickSwitch(["flick","qot"])

> Testing for flickSwitch(["tqmrw","flick"])

> Testing for
flickSwitch(["kxwwbkw","cxpczn","flick","cfv","ypkpzt","jtzz","sceitsv","flick",> Testing for
flickSwitch(["uarc","flick","rouvekz","flick","moj","mjbaew","jygy","flick","hxh

> Testing for flickSwitch(["dzfskc","cmrvj"])

> Testing for
flickSwitch(["kwz","mybh","xep","drasmh","flick","lhkuvvk","ygbgf"])> Testing for
flickSwitch(["flick","caedkyf","flick","flick","rfug","flick","dfg","bvqv","ckkkcpo

> Testing for flickSwitch(["whyteif","ock"])

JavaScript

Node v18.x

VIM

EMACS



Solution

```
1 function flickSwitch(arr){  
2     let flick = true  
3  
4     return arr.map(item => item === 'flick'? flick = !flick: flick)  
5 }
```

Impressive! You may take your time to refactor/comment your solution. Submit when ready.

Sample Tests

```
1 const {assert} = require("chai");  
2  
3 describe("Sample tests", function() {  
4     it("Basic tests", function() {  
5         assert.deepEqual(flickSwitch(["codewars", "flick", "code", "wars"]), [true, false, false,  
6         assert.deepEqual(flickSwitch(["flick", "chocolate", "adventure", "sunshine"]), [false, fa  
7         assert.deepEqual(flickSwitch(["bicycle", "jarmony", "flick", "sheep", "flick"]), [true, t  
8         assert.deepEqual(flickSwitch(["flick", "flick", "flick", "flick", "flick"]), [false, true  
9         assert.deepEqual(flickSwitch(["john, smith, susan", "flick"])), [true, false]);  
10        assert.deepEqual(flickSwitch(["bicycle"])), [true]);  
11        assert.deepEqual(flickSwitch(["flick"])), [false]).  
12    })  
13})
```

SKIP

VIEW SOLUTIONS

DISCUSS (26)

RESET

TEST

SUBMIT



¡Aviso! Libere todo el potencial de su cuenta: ¡confirme su correo electrónico para habilitar la experiencia dojo completa.

8 kyu ¿Lo lograrás?

☆ 561 🏆 141 🚗 90% de 8.515 ⌂ 51.801 de 126.017 🎙 usuario2514386

⚠ 1 problema reportado

Instrucciones

Producción

Time: 755ms Passed: 3 Failed: 0

Test Results:

Log

true

zeroFill

> Sample Tests

> Basic Tests

> Random Tests

Completed in 1ms

You have passed all of the tests! :)

javascript

Nodo v18.x

EMPUJE EMACS

Solución

```
1 const zeroFuel = (distanceToPump, mpg, fuelLeft) => {
2   const distanciaPossible = mpg * fuelLeft;
3
4   if (distanciaPossible >= distanceToPump) {
5     return true;
6   } else {
7     return false;
8   }
9 }
10
11 console.log(zeroFuel(50, 25, 2));
```

✓ ¡Buen trabajo! Puede tomarse su tiempo para refactorizar/comentar su solución. Enviar cuando esté listo.

Pruebas de muestra

```
1 const assert = require("chai").assert;
2
3 describe("zeroFill", function() {
4   it("Sample Tests", function() {
5     assert.equal(zeroFuel(50, 25, 2), true);
6     assert.equal(zeroFuel(100, 50, 1), false);
7   });
8});
```



Entrenamiento de Kata



7 kyu

62

¡Aviso! Libere todo el potencial de su cuenta: ¡confirme su correo electrónico para habilitar la experiencia dojo completa!**8 kyu** Solo necesitas uno - Principiante

☆ 640 ⚡ 222 🏆 91% de 8.956 ⚙ 54.457 de 128.062 🏁 PG1

⚠ 2 problemas reportados

Instrucciones

Producción

Time: 1017ms Passed: 2 Failed: 0

Test Results:

Basic tests

> Testing for fixed tests

Completed in 2ms

Random tests

> Testing for 40 random tests

Completed in 1ms

You have passed all of the tests! :)

javascript

Nodo v18.x

EMPUJE

EMACS

Solución

```
1 function check(a, x) {  
2     return a.includes(x);  
3 }  
4
```

 ¡Excelente! Puede tomarse su tiempo para refactorizar/comentar su solución. Enviar cuando esté listo. x

Pruebas de muestra

```
1 const chai = require("chai");  
2 const assert = chai.assert;  
3 chai.config.truncateThreshold=0;  
4  
5 describe("Basic tests", () =>{  
6     it("Testing for fixed tests", () => {  
7         assert.strictEqual(check([66, 101], 66), true);  
8         assert.strictEqual(check([101, 45, 75, 105, 99, 107], 107), true);  
9         assert.strictEqual(check(['t', 'e', 's', 't'], 'e'), true);  
10        assert.strictEqual(check(['what', 'a', 'great', 'kata'], 'kat'), false);  
11    })  
12})
```



¡Aviso! Libere todo el potencial de su cuenta: ¡confirme su correo electrónico para habilitar la experiencia dojo com



8 kyu Generar rango de números enteros.

☆ 258 ⚡ 84 🏆 93% de 3.011 ⌂ 15.964 de 24.177 🏁 XxxX88

⚠ 1 problema reportado



Instrucciones

Producción

Time: 595ms Passed: 15 Failed: 0

Test Results:

generateRange(2, 10, 2)

✓ Test Passed: Value == '[2, 4, 6, 8, 10]'

Completed in 4ms

generateRange(1, 10, 3)

✓ Test Passed: Value == '[1, 4, 7, 10]'

generateRange(1, 10, 1)

✓ Test Passed: Value == '[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]'

generateRange(1, 10, 4)

✓ Test Passed: Value == '[1, 5, 9]'

generateRange(1, 10, 5)

✓ Test Passed: Value == '[1, 6]'

generateRange for random

✓ Test Passed: Value == '[15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60]

javascript

Nodo v8.1.3

EMPUJE EMACS



Solución

```
1 function generateRange(min, max, step){  
2     let result = [];  
3     for (let i = min; i <= max; i += step) {  
4         result.push(i);  
5     }  
6     return result;  
7 }  
8 
```

✓ ¡Excelente! Puede tomarse su tiempo para refactorizar/comentar su solución. Enviar cuando esté listo.

Pruebas de muestra

```
1 describe("generateRange(2, 10, 2)", function() {  
2     Test.assertSimilar(generateRange(2, 10, 2), [2,4,6,8,10]);  
3 });
```





Entrenamiento de Kata



7 kyu

66

¡Aviso! Libere todo el potencial de su cuenta: ¡confirme su correo electrónico para habilitar la experiencia dojo completa!**8 kyu** Encuentra números que son divisibles por un número dado.

☆ 310 ⚡ 116 🏆 93% de 4.524 ⌂ 18.082 de 37.785 🏁 XxxX88

⚠ 1 problema reportado

Instrucciones

Producción

Time: 767ms Passed: 2 Failed: 0

Test Results:

Fixed tests

> should pass fixed tests

Completed in 2ms

Random tests

> should pass random tests

Completed in 3ms

> STDRR

You have passed all of the tests! :)

javascript

Nodo v18.x

EMPUJE

EMACS

Solución

```
1 function divisibleBy(numbers, divisor){  
2     return numbers.filter(number => number % divisor === 0);  
3 }  
4
```

✓ ¡Correcto! Puede tomarse su tiempo para refactorizar/comentar su solución. Enviar cuando esté listo. x

Pruebas de muestra

```
1 const Test = require('@codewars/test-compat');  
2  
3 describe("Sample tests", function() {  
4     it("should pass sample tests", function() {  
5         Test.assertDeepEquals(divisibleBy([1,2,3,4,5,6], 2), [2,4,6]);  
6         Test.assertDeepEquals(divisibleBy([1,2,3,4,5,6], 3), [3,6]);  
7         Test.assertDeepEquals(divisibleBy([0,1,2,3,4,5,6], 4), [0,4]);  
8         Test.assertDeepEquals(divisibleBy([0], 4), [0]);  
9         Test.assertDeepEquals(divisibleBy([1,3,5], 2), []);  
10    })  
11})
```



Entrenamiento de Kata



7 kyu

68

¡Aviso! Libere todo el potencial de su cuenta: ¡confirme su correo electrónico para habilitar la experiencia dojo completa!**8 kyu** Giro por gravedad

☆ 620 ⚡ 114 🏃 91% de 3.000 ⌂ 12.186 de 32.525 LógicoX

Instrucciones

Producción

Time: 786ms Passed: 2 Failed: 0

Test Results:

Log

```
[ 1, 2, 2, 3 ]  
[ 5, 5, 4, 3, 1 ]
```

Solution

> fixed tests

> random tests

Completed in 4ms

You have passed all of the tests! :)

javascript

Nodo v18.x

EMPUJE EMACS

Solución

```
1 const flip = (d, a) => {  
2   if (d === 'R') {  
3     return a.sort((x, y) => x - y);  
4   } else if (d === 'L') {  
5     return a.sort((x, y) => y - x);  
6   }  
7 };  
8  
9 // Pruebas de muestra  
10 console.log(flip('R', [3, 2, 1, 2])); // Devuelve: [1, 2, 2, 3]  
11 console.log(flip('L', [1, 4, 5, 3, 2])); // Devuelve: [5, 4, 3, 2, 1]
```

✓ ¡Pendiente! Puede tomarse su tiempo para refactorizar/comentar su solución. Enviar cuando esté listo.

Pruebas de muestra

```
1 // Since Node 10, we're using Mocha.  
2 // You can use `chai` for assertions.  
3 const chai = require("chai");  
4 const assert = chai.assert;  
5 // Uncomment the following line to disable truncating failure messages for deep equals, do:  
6 // chai.config.truncateThreshold = 0;  
7 // Since Node 12, we no longer include assertions from our deprecated custom test framework by  
8 // Uncomment the following to use the old assertions:  
9 // const Test = require("@codewars/test-compat");  
10  
11 describe("Solution", function() {
```

Entrenamiento de Kata



70

¡Aviso! Libere todo el potencial de su cuenta: ¡confirme su correo electrónico para habilitar la experiencia dojo completa!

8 kyu Agregar longitud

☆ 293 🏆 98 🏆 94% de 3.070 ⚡ 16.414 de 31.306 🛡️ escombros



Instrucciones

Producción

Time: 794ms Passed: 50 Failed: 0

Test Results:

Log

```
[ 'apple 5', 'ban 3' ]  
[ 'you 3', 'will 4', 'win 3' ]
```

Basic tests

- > Testing for "apple ban"
- > Testing for "you will win"
- > Testing for "you"
- > Testing for "y"
- > Testing for "x y z"
- > Testing for "xyz"
- > Testing for "xyz x y z xyz"
- > Testing for "a bc cde"
- > Testing for "a ab abc"
- > Testing for "a ab abc ab a"

javascript

Nodo v18.x

EMPUJE

EMACS

Solución

```
1 function addLength(str) {  
2   return str.split(' ').map(word => `${word} ${word.length}`);  
3 }  
4  
5 // Ejemplo de uso  
6 console.log(addLength("apple ban")); // Devuelve: ["apple 5", "ban 3"]  
7 console.log(addLength("you will win")); // Devuelve: ["you 3", "will 4", "win 3"]  
8
```

✓ ¡Correctomundo! Puede tomarse su tiempo para refactorizar/comentar su solución. Enviar cuando esté listo. ✎

Pruebas de muestra

```
1 const { assert } = require('chai');  
2 describe("Sample Tests", () => {  
3   it(`Testing for "apple ban"`, () => {  
4     const actual = addLength('apple ban');  
5     assert.isDefined(actual, "You need to return an array. Did you log to the console instead?  
6     assert.deepEqual(actual, ["apple 5", "ban 3"]);  
7   });  
8   it(`Testing for "you will win"`, () => assert.deepEqual(addLength('you will win'), ["you 3",  
9 ]);  
10});
```

¡Aviso! Libere todo el potencial de su cuenta: ¡confirme su correo electrónico para habilitar la experiencia dojo como profesional.



8 kyu Pozo de Ideas - Versión Fácil

☆ 309 🏆 125 🛡 93% de 3.723 ⌂ 13.834 de 31.445 🏆 PG1

⚠ 4 problemas reportados

Instrucciones

Producción

Time: 774ms Passed: 2 Failed: 0

Test Results:

Log

```
Publish!
I smell a series!
Fail!
I smell a series!
```

Basic tests

Testing for fixed tests

Completed in 1ms

Random tests

Testing for 100 random tests

Completed in 2ms

You have passed all of the tests! :)

javascript

Nodo v18.x

EMPUJE EMACS

Solución

```
1 function well(x) {
2     const goodIdeasCount = x.filter(idea => idea === 'good').length;
3
4     if (goodIdeasCount === 1 || goodIdeasCount === 2) {
5         return 'Publish!';
6     } else if (goodIdeasCount > 2) {
7         return 'I smell a series!';
8     } else {
9         return 'Fail!';
10    }
11}
12
13 // Ejemplo de uso
14 console.log(well(['good', 'bad', 'good'])); // Devuelve: 'Publish!'
15 console.log(well(['good', 'bad', 'good', 'good'])); // Devuelve: 'Publish!'
16 console.log(well(['bad', 'bad', 'bad'])); // Devuelve: 'Fail!'
17 console.log(well(['good', 'good', 'good', 'good'])); // Devuelve: 'I smell a series!'
```

Impresionante! Puede tomarse su tiempo para refactorizar/comentar su solución. Enviar cuando esté listo.

Pruebas de muestra

```
1 // Pruebas de muestra
2
3 // Pruebas de muestra
4
5 describe("Basic tests", () => {
6     it("Testing for fixed tests", () => {
7         assert.strictEqual(well(['bad', 'bad', 'bad']), 'Fail!');
8         assert.strictEqual(well(['good', 'bad', 'bad', 'bad', 'bad']), 'Publish!');
9         assert.strictEqual(well(['good', 'bad', 'bad', 'bad', 'good', 'bad', 'bad', 'good']), 'I smell a series!');
10    });
11});
```

▶ SALTAR

🔒 DESBLOQUEAR SOLUCIONES

💬 DISCUTIR (42)

REINICIAR

PRUEBA

ENTREGAR



¡Aviso! Libere todo el potencial de su cuenta: ¡confirme su correo electrónico para habilitar la experiencia dojo com



8 kyu Abreviar un nombre de dos palabras

☆ 1349 🏆 395 🏆 92% de 11,521 ⚙ 65.517 de 155.420

⚠ 6 problemas reportados



Instrucciones

Producción

Time: 783ms Passed: 2 Failed: 0

Test Results:

Log

S.H
P.F

Basic tests

> Testing for fixed tests

Completed in 1ms

Random tests

> Testing for 200 random tests

Completed in 5ms

You have passed all of the tests! :)

javascript

Nodo v18.x

EMPUJE EMACS

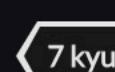
Solución

```
1 function abbrevName(name) {  
2     const [firstName, lastName] = name.split(' ');  
3     return `${firstName[0].toUpperCase()}.${lastName[0].toUpperCase()}`;  
4 }  
5 // Ejemplo de uso  
6 console.log(abbrevName('Sam Harris')) // Devuelve: 'S.H'  
7 console.log(abbrevName('patrick feeney')) // Devuelve: 'P.F'  
8  
9
```

Correcto! Puede tomarse su tiempo para refactorizar/comentar su solución. Enviar cuando esté listo.

Pruebas de muestra

```
1 const chai = require("chai");  
2 const assert = chai.assert;  
3 chai.config.truncateThreshold=0;  
4  
5 describe("Basic tests", () => {  
6     it("Testing for fixed tests", () => {  
7         assert.strictEqual(abbrevName("Sam Harris"), "S.H");  
8         assert.strictEqual(abbrevName("Patrick Feenan"), "P.F");  
9         assert.strictEqual(abbrevName("Evan Cole"), "E.C");  
10        assert.strictEqual(abbrevName("P Favuzzi"), "P.F");  
11        assert.strictEqual(abbrevName("David Mendieta"), "D.M");  
12    });  
13});
```



¡Aviso! Libere todo el potencial de su cuenta: ¡confirme su correo electrónico para habilitar la experiencia dojo com.

8 kyu Bienvenido a la ciudad

☆ 209 ⚡ 59 🏃 91% de 2.913 ⌂ 10.956 de 27.457 🏆 troglodita

⚠ 3 problemas reportados

Instrucciones

Producción

Time: 801ms Passed: 41 Failed: 0

Test Results:

Log

Hello, John Smith! Welcome to Phoenix, Arizona!

Basic Tests

Should pass Basic Tests

Completed in 1ms

Random Tests

Testing for ["Frank", "Underwood"], "Philadelphia" and "Pennsylvania"

Testing for ["Peter", "Russo"], "Los Angeles" and "California"

Testing for ["Remy", "Danton"], "Detroit" and "Michigan"

Testing for ["Zoe", "Barnes"], "Detroit" and "Michigan"

Testing for ["Peter", "Russo"], "New York City" and "New York"

Testing for ["Claire", "Underwood"], "Detroit" and "Michigan"

Testing for ["Douglas", "Doug", "Stamper"], "Detroit" and "Michigan"

Testing for ["Remy", "Danton"], "Houston" and "Texas"

javascript

Nodo v18.x

EMPUJE EMACS

Solución

```
1 function sayHello(name, city, state) {  
2     const fullName = name.join(' ');  
3     return `Hello, ${fullName}! Welcome to ${city}, ${state}!`;  
4 }  
5  
6 // Ejemplo de uso  
7 console.log(sayHello(['John', 'Smith'], 'Phoenix', 'Arizona'));  
8 // Devuelve: 'Hello, John Smith! Welcome to Phoenix, Arizona!'  
9
```

✓ ¡Pendiente! Puede tomarse su tiempo para refactorizar/comentar su solución. Enviar cuando esté listo.

Pruebas de muestra

```
1 const chai = require("chai");  
2 const assert = chai.assert;  
3 chai.config.truncateThreshold=0;  
4  
5 describe("Sample Tests", () => {  
6     it("Should pass Sample Tests", () => {  
7         assert.strictEqual(sayHello(['John', 'Smith'], 'Phoenix', 'Arizona'), 'Hello, John Smith!');  
8         assert.strictEqual(sayHello(['Franklin', 'Delano', 'Roosevelt'], 'Chicago', 'Illinois'), 'Hello, Franklin Delano Roosevelt!');  
9         assert.strictEqual(sayHello(['Wallace', 'Russel', 'Osbourne'], 'Albany', 'New York'), 'Hello, Wallace Russel Osbourne!');  
10    });  
11});
```

AVISO: Libere todo el potencial de su cuenta. ¡Confirme su correo electrónico para habilitar la experiencia dojo.com



7 kyu

78

8 kyu Obtener número de cadena

☆ 133 🏆 59 🚶 93% de 1.853 ⌂ 4.940 de 11.480

wichú

⚠ 3 problemas reportados

Instrucciones

Producción

Time: 993ms Passed: 2 Failed: 0

Test Results:

Log

5

Basic tests

> should work as expected

Completed in 3ms

Random tests

> should work as expected

Completed in 3ms

> STDERR

You have passed all of the tests! :)

javascript

Nodo v18.x

Solución

```
1 function getNumberFromString(s) {  
2     const digitsOnly = s.replace(/\D/g, ''); // Eliminar todos los caracteres que no sean dígitos  
3     return parseInt(digitsOnly); // Convertir la cadena de dígitos en un número  
4 }  
5  
6 // Ejemplo de uso  
7 console.log(getNumberFromString("hell5o world")); // Devuelve: 56  
8
```

¡Impresionante! Puede tomarse su tiempo para refactorizar/comentar su solución. Enviar cuando esté listo.

Pruebas de muestra

```
1 const Test = require('@codewars/test-compat');  
2  
3 describe("Basic tests", function() {  
4     it("should work as expected", function() {  
5         Test.assertSimilar(getNumberFromString("1"), 1);  
6         Test.assertSimilar(getNumberFromString("123"), 123);  
7         Test.assertSimilar(getNumberFromString("this is number: 7"), 7);  
8     });  
9 });
```

▶ SALTAR

🔒 DESBLOQUEAR SOLUCIONES

💬 DISCUTIR (18)

REINICIAR

PRUEBA

ENTREGAR

¡Aviso! Libere todo el potencial de su cuenta: ¡confirme su correo electrónico para habilitar la experiencia dojo com



7 kyu 80

7 kyu Más alto y más bajo

☆ 3173 ⚡ 724 🏆 90% de 22,583 ⌂ 113.032 de 276.085

decanodos

Instrucciones

Producción

Time: 761ms Passed: 10 Failed: 0

Test Results:

Log

```
5 1  
5 -3  
9 -5
```

Fixed tests

- > Testing basic input
- > Testing common string sort mistake
- > Testing positive and negative
- > Testing positive and positive
- > Testing negative and negative
- > Testing positive, negative and zero
- > Testing positive, positive and zero
- > Testing negative, negative and zero
- > Testing single

Completed in 2ms

Random tests

javascript

Nodo v18.x

Solución

```
1 function highAndLow(numbers) {  
2     const numsArray = numbers.split(' ').map(Number);  
3     const maxNum = Math.max(...numsArray);  
4     const minNum = Math.min(...numsArray);  
5     return `${maxNum} ${minNum}`;  
6 }  
7  
8 console.log(highAndLow("1 2 3 4 5"));  
9 console.log(highAndLow("1 2 -3 4 5"));  
10 console.log(highAndLow("1 9 3 4 -5"));  
11
```

✓ ¡Buen trabajo! Puede tomarse su tiempo para refactorizar/comentar su solución. Enviar cuando esté listo.

Pruebas de muestra

```
1 const chai = require("chai");  
2 const assert = chai.assert;  
3 chai.config.truncateThreshold=0;  
4  
5 describe("Example tests", () => {  
6     it("Test 1", () => {  
7         assert.strictEqual(highAndLow("8 3 -5 42 -1 0 0 -9 4 7 4 -4"), "42 -9");  
8     });  
9     it("Test 2", () => {  
10        assert.strictEqual(highAndLow("1 2 3"), "3 1");  
11    });
`
```

► SALTAR

🔒 DESBLOQUEAR SOLUCIONES

💬 DISCUTIR (280)

REINICIAR

PRUEBA

ENTREGAR

AVISO: Libere todo el potencial de su cuenta. ¡Confirme su correo electrónico para habilitar la experiencia dojo.com.



82

7 kyu Suma del primer enésimo término de la Serie

☆ 1619 ⚡ 344 🏆 86% de 7.813 ⌂ 28.696 de 84.715

DivyanshBatham

⚠ 1 problema reportado

Instrucciones

Producción

Time: 1015ms Passed: 2 Failed: 0

Test Results:

Log

1.00
1.25
1.57

Solution tests

> Fixed tests

> Random tests

Completed in 5ms

You have passed all of the tests! :)

Solución

```
1 function SeriesSum(n) {  
2     let sum = 0;  
3     for (let i = 0; i < n; i++) {  
4         sum += 1 / (1 + i * 3);  
5     }  
6     return sum.toFixed(2);  
7 }  
8  
9 console.log(SeriesSum(1)); // Devuelve: "1.00"  
10 console.log(SeriesSum(2)); // Devuelve: "1.25"  
11 console.log(SeriesSum(5)); // Devuelve: "1.57"
```

✓ ¡Correcto! Puede tomarse su tiempo para refactorizar/comentar su solución. Enviar cuando esté listo.

Pruebas de muestra

```
1 const { assert } = require('chai');  
2  
3 describe("Sample tests", () => {  
4     it("n = 1", () => {  
5         let actual = SeriesSum(1);  
6         checkReturnValue(actual);  
7         assert.strictEqual(actual, "1.00", "n = 1")  
8     });  
9     it("n = 2", () => assert.strictEqual(SeriesSum(2), "1.25", "n = 2"));  
10    it("n = 3", () => assert.strictEqual(SeriesSum(3), "1.39", "n = 3"));  
11    it("n = 4" () => assert.strictEqual(SeriesSum(4), "1.49", "n = 4"));
```

▶ SALTAR

🔒 DESBLOQUEAR SOLUCIONES

💬 DISCUTIR (220)

REINICIAR

PRUEBA

ENTREGAR



¡Aviso!

Libere todo el potencial de su cuenta: ¡confirme su correo electrónico para habilitar la experiencia dojo cor



162

7 kyu encontrar las capitales ✓

☆ 385 ⚡ 127 🏆 93% de 3.476 ⌂ 22.415 de 38.602

⚠ 4 problemas reportados

Instrucciones

Producción

Soluciones pasadas

Time: 878ms Passed: 2 Failed: 0

Test Results:

Log

[0, 3, 4, 6]

Basic tests

> Testing for fixed tests

Completed in 1ms

Random tests

> Testing for 20 random tests

Completed in 2ms

You have passed all of the tests! :)

javascript

Nodo v18.x

EMPUJE

EMACS

Solución

```
1 var capitals = function (word) {  
2     const indexes = [];  
3     for (let i = 0; i < word.length; i++) {  
4         if (word[i] === word[i].toUpperCase()) {  
5             indexes.push(i);  
6         }  
7     }  
8     return indexes;  
9 };  
10  
11 console.log(capitals("CodEWaRs")); // Devuelve: [0, 3, 4, 6]  
12
```

✔ ¡Excelente! Puede tomarse su tiempo para refactorizar/comentar su solución. Enviar cuando esté listo.

Pruebas de muestra

```
1 const chai = require("chai");  
2 const assert = chai.assert;  
3 chai.config.truncateThreshold=0;  
4  
5 describe("Basic tests", () => {  
6     it("Testing for fixed tests", () => {  
7         assert.deepEqual(capitals('CodEWaRs'), [0,3,4,6] );  
8     });  
9 });
```



Entrenamiento de Kata



7 kyu 92

¡Aviso! Libere todo el potencial de su cuenta: ¡confirme su correo electrónico para habilitar la experiencia dojo completa!**6 kyu** Crear número de teléfono

☆ 3419 ⚡ 675 🏆 85% de 26,596 ⚡ 113.997 de 309.565 xDranik

Instrucciones

Producción

Time: 737ms Passed: 2 Failed: 0

Test Results:



(123) 456-7890

Create Phone Number

> Fixed tests

> Random tests

Completed in 4ms

You have passed all of the tests! :)

javascript

Nodo v18.x

EMPUJE EMACS

Solución

```
1 function createPhoneNumber(numbers) {  
2     const areaCode = numbers.slice(0, 3).join('');  
3     const firstPart = numbers.slice(3, 6).join('');  
4     const secondPart = numbers.slice(6, 10).join('');  
5     return `(${areaCode}) ${firstPart}-${secondPart}`;  
6 }  
7  
8 console.log(createPhoneNumber([1, 2, 3, 4, 5, 6, 7, 8, 9, 0])); // Devuelve: "(123) 456-7890"  
9
```

¡Pendiente! Puede tomarse su tiempo para refactorizar/comentar su solución. Enviar cuando esté listo.

Pruebas de muestra

```
1 const chai = require("chai");  
2 const assert = chai.assert;  
3 chai.config.truncateThreshold=0;  
4  
5 describe("Create Phone Number", () => {  
6     it("Fixed tests", () => {  
7         assert.strictEqual(createPhoneNumber([1, 2, 3, 4, 5, 6, 7, 8, 9, 0]), "(123) 456-7890");  
8         assert.strictEqual(createPhoneNumber([1, 1, 1, 1, 1, 1, 1, 1, 1]), "(111) 111-1111");  
9         assert.strictEqual(createPhoneNumber([1, 2, 3, 4, 5, 6, 7, 8, 9, 0]), "(123) 456-7890");  
10    });  
11});
```



Entrenamiento de Kata



7 kyu

130

¡Aviso! Libere todo el potencial de su cuenta: ¡confirme su correo electrónico para habilitar la experiencia dojo completa!**6 kyu** ¡Deja de gninnipS, mi sdroW!

☆ 3266 ⚡ 577 🏆 91% de 25.140 ⌂ 91.715 de 257.472

xDranik

⚠ 5 problemas reportados

Instrucciones

Producción

Time: 892ms Passed: 2 Failed: 0

Test Results:

Log

```
Hey wollef sroirraw
This is a test
This is rehtona test
```

Spinning words

> Fixed tests

> Random tests

Completed in 6ms

You have passed all of the tests! :)

javascript

Node v18.x

EMPUJE EMACS

**Solución**

```
1 function spinWords(string) {
2   return string.split(' ').map(word => word.length >= 5 ? word.split('').reverse().join('') :
3 }
4
5 console.log(spinWords("Hey fellow warriors"));
6 console.log(spinWords("This is a test"));
7 console.log(spinWords("This is another test"));
8
```

Correcto! Puede tomarse su tiempo para refactorizar/comentar su solución. Enviar cuando esté listo.**Pruebas de muestra**

```
1 const chai = require("chai");
2 const assert = chai.assert;
3
4 describe("Spinning words", ()=>{
5   it("Sample tests", ()=>{
6     assert.strictEqual(spinWords("Welcome"), "emocleW");
7     assert.strictEqual(spinWords("Hey fellow warriors"), "Hey wollef sroirraw");
8     assert.strictEqual(spinWords("This is a test"), "This is a test");
9     assert.strictEqual(spinWords("This is another test"), "This is rehtona test");
10    assert.strictEqual(spinWords("You are almost to the last test"), "You are tsomla to the la
```



Entrenamiento de Kata



6 kyu

138

¡Aviso! Libere todo el potencial de su cuenta: ¡confirme su correo electrónico para habilitar la experiencia dojo completa!

6 kyu Convertir cadena a caja de camellos

☆ 3071 🏆 551 🏆 88% de 15.722 ⚡ 61.256 de 176.223 🏆 jhoffner

⚠ 19 problemas reportados

Instrucciones

Producción

Time: 805ms Passed: 203 Failed: 0

Test Results:

Log

theStealthWarrior
TheStealthWarrior
TheStealthWarrior

toCamelCase

- > should handle empty values
- > should remove underscores and convert first letter of word to upper case
- > should remove hyphens and convert first letter of word to upper case

Completed in 1ms

Random Tests

- > Testing with The_Pippi_Is-Pippi
- > Testing with A_Pippi_is_pippi
- > Testing with the_Cat-is-pippi
- > Testing with the_pippi_Was_cute

javascript

Nodo v18.x

EMPUJE

EMACS

Solución

```
1 function toCamelCase(str) {  
2     const words = str.split(/[-_]/);  
3     const camelCaseWords = words.map((word, index) => {  
4         if (index === 0) {  
5             return word;  
6         } else {  
7             return word.charAt(0).toUpperCase() + word.slice(1);  
8         }  
9     });  
10    return camelCaseWords.join('');  
11}  
12  
13 console.log(toCamelCase("the-stealth-warrior")); // Devuelve: "theStealthWarrior"  
14 console.log(toCamelCase("The_S stealth_Warrior")); // Devuelve: "TheStealthWarrior"  
15 console.log(toCamelCase("The_Stealth-Warrior")); // Devuelve: "TheStealthWarrior"  
16
```

Correcto! Puede tomarse su tiempo para refactorizar/comentar su solución. Enviar cuando esté listo.

Pruebas de muestra

```
1 const _ = require('lodash');  
2  
3 describe("Tests", () => {  
4     it("test", () => {  
5         assert.strictEqual(toCamelCase(""), "", "An empty string was provided but not returned")  
6         assert.strictEqual(toCamelCase("the_stealth_warrior"), "theStealthWarrior", "toCamelCase(  
7             \"the_stealth_warrior\") did not return correct result")  
8         assert.strictEqual(toCamelCase("The-Stealth-Warrior"), "TheStealthWarrior", "toCamelCase(  
9             \"The-Stealth-Warrior\") did not return correct result")  
10        assert.strictEqual(toCamelCase("A-B-C"), "ABC", "toCamelCase('A-B-C') did not return corr
```

¡Aviso! Libere todo el potencial de su cuenta: ¡confirme su correo electrónico para habilitar la experiencia dojo cor



6 kyu

146

6 kyu Eliminar apariciones de un elemento si ocurre más de n veces

☆ 3008 ⚡ 625 🏆 91% de 7.655 ⌂ 27.369 de 88.800 🔍 justyfy

⚠ 4 problemas reportados

Instrucciones

Producción

Time: 819ms Passed: 2 Failed: 0

Test Results:

Log

```
[ 1, 2, 3, 1, 2, 3 ]  
[ 20, 37, 21 ]
```

Tests

Basic tests

Random tests

Completed in 10ms

You have passed all of the tests! :)

Solución

```
1 function deleteNth(arr, n) {  
2   const countMap = {};  
3   return arr.filter(num => {  
4     countMap[num] = (countMap[num] || 0) + 1;  
5     return countMap[num] <= n;  
6   });  
7 }  
8  
9 console.log(deleteNth([1, 2, 3, 1, 2, 1, 2, 3], 2)); // Devuelve: [1, 2, 3, 1, 2]  
10 console.log(deleteNth([20, 37, 20, 21], 1)); // Devuelve: [20, 37, 21]
```

✓ ¡Correcto! Puede tomarse su tiempo para refactorizar/comentar su solución. Enviar cuando esté listo.

Pruebas de muestra

```
1 const { assert } = require('chai');  
2  
3 describe("Tests", () => {  
4   it("test", () => {  
5     assert.sameOrderedMembers(deleteNth([20,37,20,21], 1), [20,37,21])  
6     assert.sameOrderedMembers(deleteNth([1,1,3,3,7,2,2,2,2], 3), [1, 1, 3, 3, 7, 2, 2, 2])  
7     assert.sameOrderedMembers(deleteNth([12,39,19,39,39,19,12], 1), [12, 39, 19])  
8   });  
9 });
```

Entrenamiento de Kata

**Aviso!** Libere todo el potencial de su cuenta: ¡confirme su correo electrónico para habilitar la experiencia dojo completa.**6 kyu** construir torre

☆ 2772 ⚡ 573 🏆 91% de 8.365 ⚙ 31.344 de 93.265 🎖 8fdafs2



Instrucciones

Producción

Time: 819ms Passed: 2 Failed: 0

Test Results:

Log

```
[ ' * ', ' *** ', '*****' ]  
[  
  ' * ',  
  ' *** ',  
  ' ***** ',  
  ' ***** ',  
  ' ***** ',  
  ' ***** ',  
  '*****' ]
```

towerBuilder

> Basic Test

> Blanket Test

Completed in 27ms

You have passed all of the tests! :)

javascript

Nodo v18.x

EMPUJE

EMACS

Solución

```
1 function towerBuilder(nFloors) {  
2   const tower = [];  
3   const maxWidth = 2 * nFloors - 1;  
4  
5   for (let i = 0; i < nFloors; i++) {  
6     const stars = 2 * i + 1;  
7     const spaces = (maxWidth - stars) / 2;  
8     const floor = ' '.repeat(spaces) + '*'.repeat(stars) + ' '.repeat(spaces);  
9     tower.push(floor);  
10  }  
11  
12  return tower;  
13}  
14  
15 console.log(towerBuilder(3)); // Devuelve una matriz con la torre de 3 pisos  
16 console.log(towerBuilder(6)); // Devuelve una matriz con la torre de 6 pisos
```

Correcto! Puede tomarse su tiempo para refactorizar/comentar su solución. Enviar cuando esté listo.

Pruebas de muestra

```
1 const Test = require('@codewars/test-compat');  
2  
3 describe("Tests", () => {  
4   it("test", () => {  
5     Test.assertDeepEquals(towerBuilder(1), ["*"]);  
6     Test.assertDeepEquals(towerBuilder(2), [" * ", "***"]);  
7     Test.assertDeepEquals(towerBuilder(3), [" * ", " *** ", "*****"]);  
8   });  
9});
```



Entrenamiento de Kata



6 kyu

162

¡Aviso! Libere todo el potencial de su cuenta: ¡confirme su correo electrónico para habilitar la experiencia dojo completa!



6 kyu Sigue a ese espía

☆ 147 ⚡ 38 🏆 95% de 339 ⚡ 819 de 1.910 adromil

⚠ 1 problema reportado

Instrucciones

Producción

Time: 730ms Passed: 108 Failed: 0

Test Results:

Log

USA, BRA, UAE, JPN, PHL

Follow That Spy

> Fixed (8 of 8 Assertions)

> Random: [["NHY","REW"],["REW","III"],["III","EFG"],["EFG","FGT"],["FGT","VVV"],["VVV","ASD"],["ASD","IYR"],["IYR","DDD"],["DDD","GFD"],["GFD","DPE"]]

> Random: [["FGT","VVV"],["VVV","REW"],["REW","GFD"],["GFD","NHY"],["NHY","EFG"],["EFG","DPE"],["DPE","JJJ"],["JJJ","IYR"],["IYR","YTR"],["YTR","III"]]

> Random: [["NHY","JJJ"],["JJJ","ASD"],["ASD","IYR"],["IYR","DPE"],["DPE","YTR"],["YTR","ABC"],["ABC","FGT"],["FGT","III"],["III","ZZZ"],["ZZZ","DDD"]]

> Random: [["ASD","EFG"],["EFG","ABC"],["ABC","GFD"],["GFD","IYR"],["IYR","JJJ"],["JJJ","OOO"],["OOO","YTR"],["YTR","VVV"],["VVV","ZZZ"]]

javascript

Nodo v8.1.3

EMPUJE

EMACS



Solución

```
1 function findRoutes(routes) {
2   const destinations = routes.reduce((acc, route) => {
3     acc.push(route[0], route[1]);
4     return acc;
5   }, []);
6
7   const uniqueDestinations = [...new Set(destinations)];
8
9   const startingPoints = routes.map(route => route[0]);
10  const endingPoints = routes.map(route => route[1]);
11
12  const start = startingPoints.find(point => !endingPoints.includes(point));
13
14  if (!start) return "No se puede encontrar un punto de inicio.";
15
16  let currentDestination = start;
17  const routeSequence = [start];
```

Pruebas de muestra

```
1 describe("Follow That Spy", function() {
2   it("Fixed", function() {
3     var routes1 = findRoutes([["MNL", "TAG"], ["CEB", "TAC"], ["TAG", "CEB"], ["TAC", "BOR"]]);
4     Test.assertEquals(routes1, "MNL, TAG, CEB, TAC, BOR");
5     var routes2 = findRoutes([["UK", "GER"], ["GER", "BEL"], ["BEL", "CAN"]]);
6     Test.assertEquals(routes2, "UK, GER, BEL, CAN");
7     var routes3 = findRoutes([["Chicago", "Winnipeg"], ["Halifax", "Montreal"], ["Montreal", "Toronto"], ["Toronto", "Chicago"], ["Chicago", "Winnipeg"], ["Winnipeg", "Seattle"]]);
8     Test.assertEquals(routes3, "Halifax, Montreal, Toronto, Chicago, Winnipeg, Seattle");
9     var routes4 = findRoutes([["Agra", "Tokyo"], ["Tokyo", "Seoul"], ["Seoul", "Ljubljana"], ["Ljubljana", "Wroclaw"], ["Wroclaw", "Nashville"], ["Nashville", "Amsterdam"], ["Amsterdam", "Hull"], ["Hull", "Vancouver"]]);
10    Test.assertEquals(routes4, "Agra, Tokyo, Seoul, Ljubljana, Wroclaw, Nashville, Amsterdam, Hull, Vancouver")});
```

