

Task1

☆ 10 1 88% of 52 34 of 251 zeroxonefour

Instructions Output

Time: 792ms Passed: 3 Failed: 0

Test Results:

Log

6

Fixed Tests

regular

edge cases

Completed in 1ms

Random Tests

random

Solution

```
1 function largestRadialSum(arr, d) {
2   let maxHonor = -Infinity;
3   const n = arr.length;
4
5   for (let i = 0; i < n; i++) {
6     let currentSum = 0;
7
8     for (let j = 0; j < d; j++) {
9       currentSum += arr[(i + j * (n / d)) % n];
10    }
11  }
12}
```

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

Sample Tests

```
1 const chai = require("chai");
2 const assert = chai.assert;
3
4 describe("Fixed Tests", function() {
5   it("regular", function() {
6     assert.strictEqual(largestRadialSum([1,2,3,4], 2), 6);
7     assert.strictEqual(largestRadialSum([1,5,6,3,4,2], 3), 11);
8     assert.strictEqual(largestRadialSum([1,1,1,1,1,1], 3), 3);
9   });
10});
```

SKIP DISCUSS(10) RESET TEST SUBMIT

Task2

Instructions Output

Time: 883ms Passed: 51 Failed: 0

Test Results:

Log

1

FindEvenIndex

Tests

Completed in 1ms

50 Random tests

Random Testing

Random Testing

Solution

```
1 function findEvenIndex(arr)
2 {
3   let foundIndex = -1;
4
5   for (let i = 0; i < arr.length; i++)
6   {
7     let sum1 = getArraySum(arr,0,i);
8     let sum2 = getArraySum(arr,i+1,arr.length);
9     if(sum1 === sum2)
10    {
11      foundIndex = i;
12    }
13  }
14}
```

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

Sample Tests

```
4 it("Tests", function() {
5   Test.assertEquals(findEvenIndex([1,2,3,4,3,2,1]),3, "The array was: [1,2,3
6   Test.assertEquals(findEvenIndex([1,100,50,-51,1]),1, "The array was: [1
7   Test.assertEquals(findEvenIndex([1,2,3,4,5,6]),-1, "The array was: [1,2,3
8   Test.assertEquals(findEvenIndex([20,10,30,10,10,15,35]),3, "The array was
9   });
10});
```

Task3

☆ 43 11 94% of 112 610 dulaccc 3 Issues Reported

Instructions Output

Time: 930ms Passed: 3 Failed: 0

Test Results:

Log

[['tsar', 'star', 'tars'], ['rat', 'tar'],

Human cases

Light lists

Edge cases

Completed in 2ms

Superhero cases

Heavy computation that is way too long to be output (so if it

Solution

```
1 function groupAnagrams(words){
2   const groupedAnagrams = {};
3
4   for (const word of words) {
5     const sortedWord = word.split('').sort().join('');
6
7     if (groupedAnagrams[sortedWord]) {
8       groupedAnagrams[sortedWord].push(word);
9     } else {
10      groupedAnagrams[sortedWord] = [word];
11    }
12  }
13}
```

Sample Tests

```
1 describe("Tests", () => {
2   it("test", () => {
3     assertSimilarUnsorted(groupAnagrams(["rat", "tar", "star"]), [
4     ]);
5   });
6 });
```

Task 4

6kyu

Unpack delicious sausages! ✓

☆ 19 8 92% of 107 181 of 440 WellWellWell

Instructions Output Past Solutions

Time: 901ms Passed: 105 Failed: 0

Test Results:

Log

I I I I))) l l l l @ @ @ @

Solution

- > only lays valid sausage packages
- > does not lay the 5th reward package
- > only lays valid sausage packages when only one box
- > lays no sausages when truck contains only other products
- > lays no sausages when truck is empty

Completed in 1ms

Random Tests

- > Random test 1
- > Random test 2

JavaScript Node v18.x VIM EMACS

Solution

```
1 function unpackSausages(truck) {
2   let sausagesCounter = 0;
3   let output = "";
4
5   for (const box of truck) {
6     for (const package of box) {
7       if (package.startsWith("(") && package.endsWith(")") && (package.length - 2) % 4 === 0) {
8         let validPackage = true;
9         for (let i = 1; i < package.length - 1; i++) {
10          if (package[i] !== package[i+1]) {
11            validPackage = false;
12            break;
13          }
14        }
15      }
16    }
17  }
18 }
19
20 export default unpackSausages;
```

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

Sample Tests

```
1 const chai = require("chai");
2 const assert = chai.assert;
3
4 describe("Solution", function() {
5   it("only lays valid sausage packages", function() {
6     assert.strictEqual(unpackSausages(["(----)", "[IIII]", "_HHH_"], ["IuI", "[]"])), "zz", ["[####
7   ]);
8   it("does not lay the 5th reward package", function() {
9     assert.strictEqual(unpackSausages(["[IIII]", "[IIII]", "[IIII]", "[IIII]", "[IIII]"], ["[####", "[IIII]", "[IIII]"])),
9   });
10 });
```

SKIP VIEW SOLUTIONS DISCUSS (9) RESET TEST SUBMIT

Task5

6kyu

Get root property name

☆ 48 8 89% of 121 244 of 375 kodejuice 2 Issues Reported

Instructions Output

Time: 5490ms Passed: 600 Failed: 0

Test Results:

Log

2f7

Random Test cases

- Test Passed: Value == null
- Test Passed: Value == null
- Test Passed: Value == 'jn&r.'
- Test Passed: Value == null
- Test Passed: Value == 'o)c=#'
- Test Passed: Value == null
- Test Passed: Value == null
- Test Passed: Value == null
- Test Passed: Value == '?.(r7'

JavaScript Node v8.1.3 VIM EMACS

Solution

```
1 function getRootProperty(obj, target) {
2   for (const key in obj) {
3     if (Array.isArray(obj[key]) && obj[key].includes(target)) {
4       return key;
5     } else if (typeof obj[key] === 'object') {
6       const result = getRootProperty(obj[key], target);
7       if (result !== null) return key;
8     }
9   }
10  return null;
11 }
12
13 const object1 = {
14   "one": {
15     "nest1": {
16       "val1": [9, 34, 92, 100]
17     }
18   }
19 }
```

Sample Tests

```
1 describe("Tests", () => {
2   it("test", () => {
3     //Basic test
4
5     const object = {
6       "one": {
7         "nest1": {
8           "val1": [9, 34, 92, 100]
9         }
10       }
11     }
12   });
13 });
```

SKIP UNLOCK SOLUTIONS DISCUSS (12) RESET TEST SUBMIT

Task6

5kyu

Land perimeter

☆ 401 🌟 76 ↗ 95% of 597 🌐 1,617 of 4,389 👤 St3f4n

Instructions Output

Time: 652ms Passed: 106 Failed: 0

Test Results:

Testing

Basic (6 of 6 Assertions)

Random (100 of 100 Assertions)

Completed in 27ms

You have passed all of the tests! :)

JavaScript Node v8.1.3 VIM EMACS

Solution

```
2 let totalPerimeter = 0;
3
4 for (let i = 0; i < arr.length; i++) {
5   for (let j = 0; j < arr[i].length; j++) {
6     if (arr[i][j] === 'X') {
7       if (i === 0 || arr[i-1][j] === '0') totalPerimeter++;
8       if (i === arr.length - 1 || arr[i+1][j] === '0') totalPerimeter++;
9       if (j === 0 || arr[i][j-1] === '0') totalPerimeter++;
10      if (j === arr[i].length - 1 || arr[i][j+1] === '0') totalPerimeter++;
11    }
12  }
13 }
14
15 return `Total land perimeter: ${totalPerimeter}`;
```

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

Sample Tests

```
1 describe("Testing", function() {
2   it("Basic testing", () => {
3     Test.assertEquals(landPerimeter(["0X000X", "0X0X00", "XX000X", "0XX0X0", "00X00X", "0X0000", "00
4     Test.assertEquals(landPerimeter(["0X000", "000XX", "0XX00", "X0000", "X0000", "XX000", "X0X00", "00
5     Test.assertEquals(landPerimeter(["XXXXX000", "00X00000", "000000X0", "XX0000X0", "0X0X000X"]), "
6     Test.assertEquals(landPerimeter(["X000X00", "0X00000", "X0X0X00", "0X0XX00", "000000X", "000X00X
7     Test.assertEquals(landPerimeter(["0000X0", "X0X00X", "XX0X0X", "X0X000", "000000", "000X00", "00
8   });
9 }
```