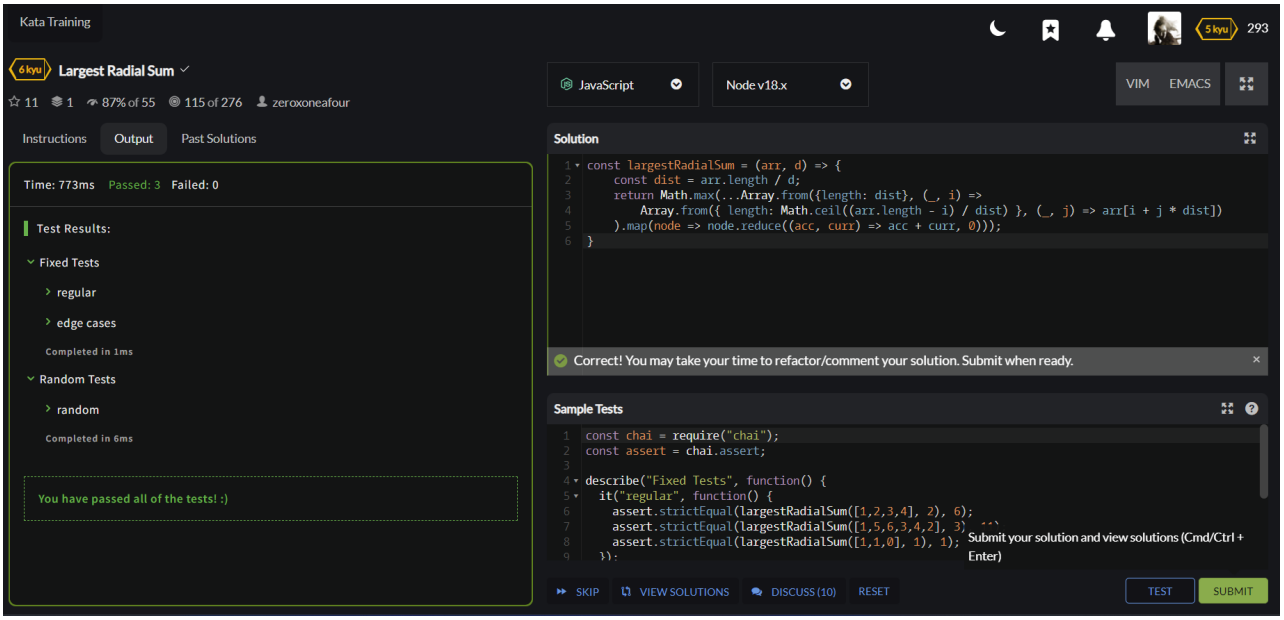
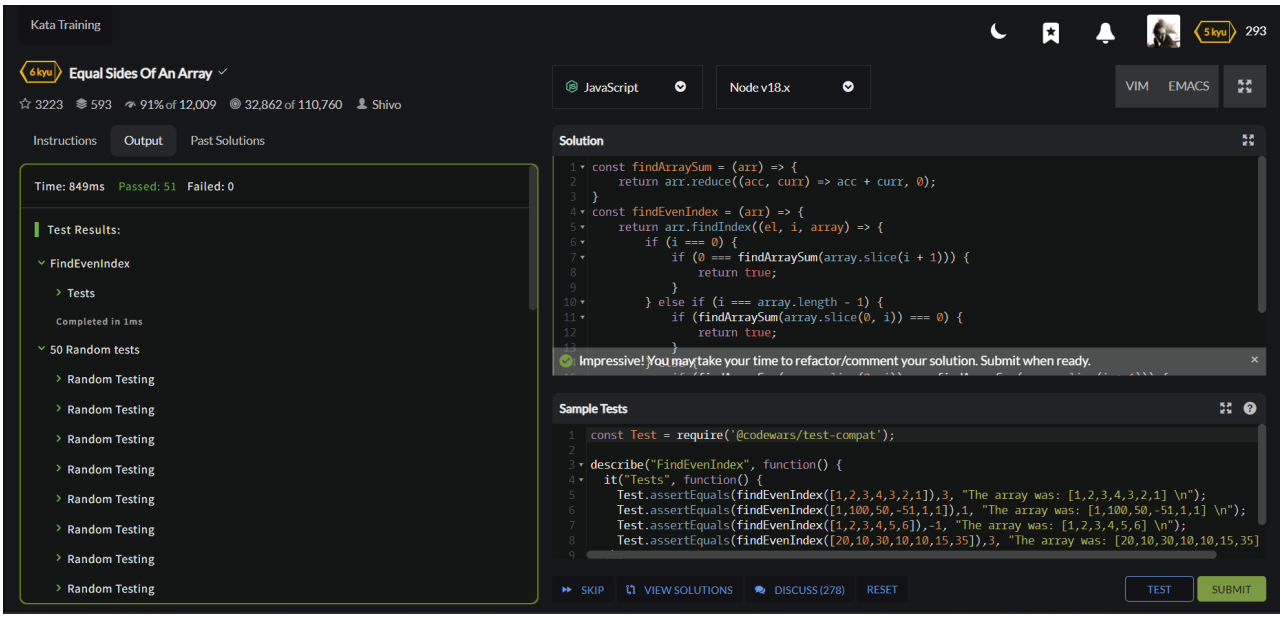
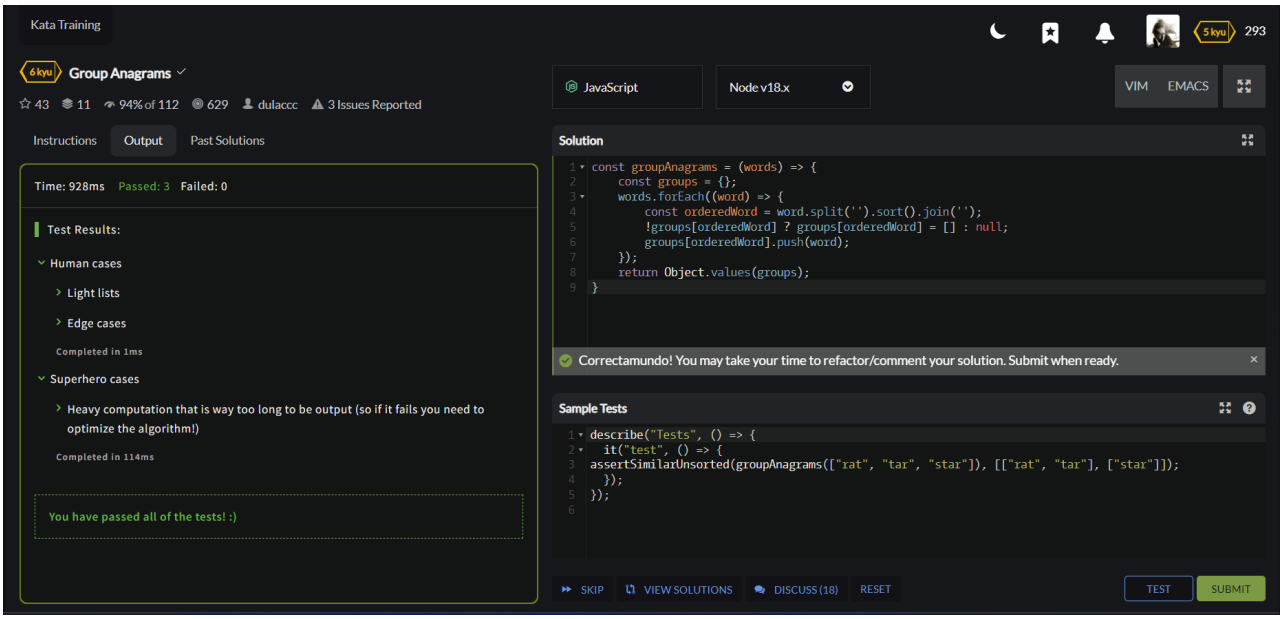
1.


Kata Training
6 kyu
Largest Radial Sum
11
1
87% of 55
115 of 276
zeroxoneafour
Instructions
Output
Past Solutions
Time: 773ms
Passed: 3
Failed: 0
Test Results:
Fixed Tests
regular
edge cases
Completed in 1ms
Random Tests
random
Completed in 6ms
You have passed all of the tests! :)
JavaScript
Node v18.x
VIM
EMACS
Solution
const largestRadialSum = (arr, d) => {
const dist = arr.length / d;
return Math.max(...Array.from({length: dist}, (_, i) =>
Array.from({ length: Math.ceil((arr.length - i) / dist) }, (_, j) => arr[i + j * dist])
).map(node => node.reduce((acc, curr) => acc + curr, 0)));
}
Correct! You may take your time to refactor/comment your solution. Submit when ready.
Sample Tests
const chai = require("chai");
const assert = chai.assert;
describe("Fixed Tests", function() {
it("regular", function() {
assert.strictEqual(largestRadialSum([1,2,3,4], 2), 6);
assert.strictEqual(largestRadialSum([1,5,6,3,4,2], 3),
assert.strictEqual(largestRadialSum([1,1,0], 1), 1);
Submit your solution and view solutions (Cmd/Ctrl + Enter)
SKIP
VIEW SOLUTIONS
DISCUSS (10)
RESET
TEST
SUBMIT

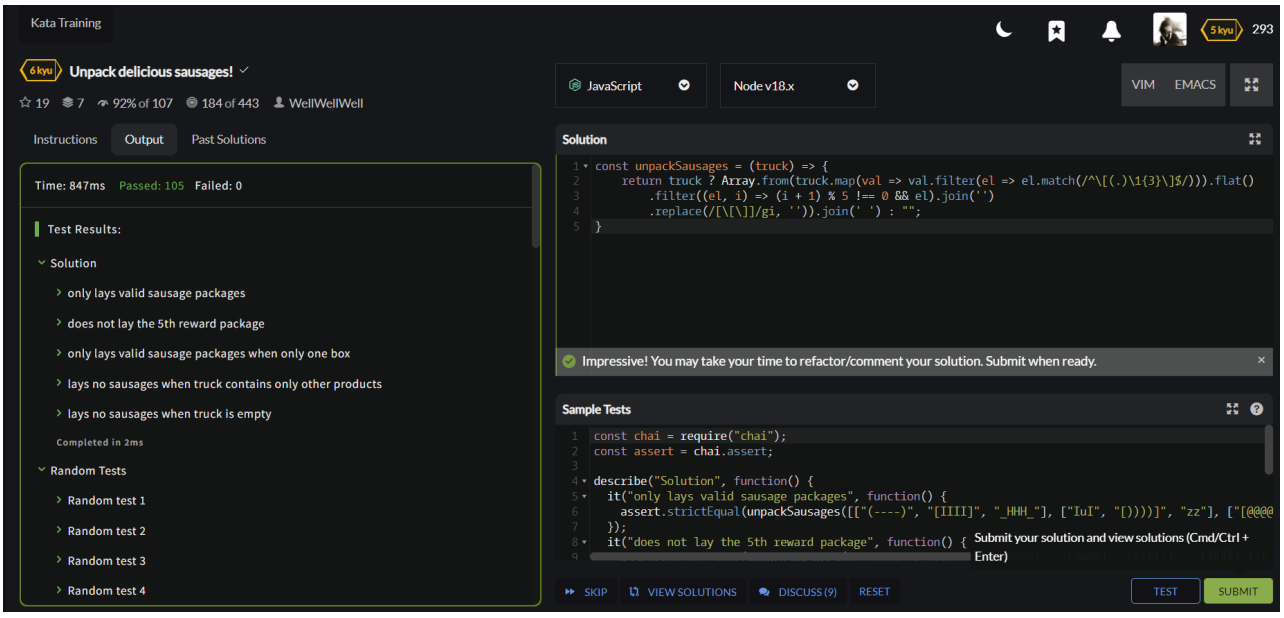
2.


Kata Training
6 kyu
Equal Sides Of An Array
3223
593
91% of 12,009
32,862 of 110,760
Shivo
Instructions
Output
Past Solutions
Time: 849ms
Passed: 51
Failed: 0
Test Results:
FindEvenIndex
Tests
Completed in 1ms
50 Random tests
Random Testing
Random Testing
Random Testing
Random Testing
Random Testing
Random Testing
Random Testing
Random Testing
JavaScript
Node v18.x
VIM
EMACS
Solution
const findArraySum = (arr) => {
return arr.reduce((acc, curr) => acc + curr, 0);
}
const findEvenIndex = (arr) => {
return arr.findIndex((el, i, array) => {
if (i === 0) {
if (0 === findArraySum(array.slice(i + 1))) {
return true;
}
} else if (i === array.length - 1) {
if (findArraySum(array.slice(0, i)) === 0) {
return true;
}
Impressive! You may take your time to refactor/comment your solution. Submit when ready.
Sample Tests
const Test = require('@codewars/test-compat');
describe("FindEvenIndex", function() {
it("Tests", function() {
Test.assertEquals(findEvenIndex([1,2,3,4,3,2,1]),3, "The array was: [1,2,3,4,3,2,1] \n");
Test.assertEquals(findEvenIndex([1,100,50,-51,1,1]),1, "The array was: [1,100,50,-51,1,1] \n");
Test.assertEquals(findEvenIndex([1,2,3,4,5,6]),-1, "The array was: [1,2,3,4,5,6] \n");
Test.assertEquals(findEvenIndex([20,10,30,10,10,15,35]),3, "The array was: [20,10,30,10,10,15,35]
SKIP
VIEW SOLUTIONS
DISCUSS (278)
RESET
TEST
SUBMIT

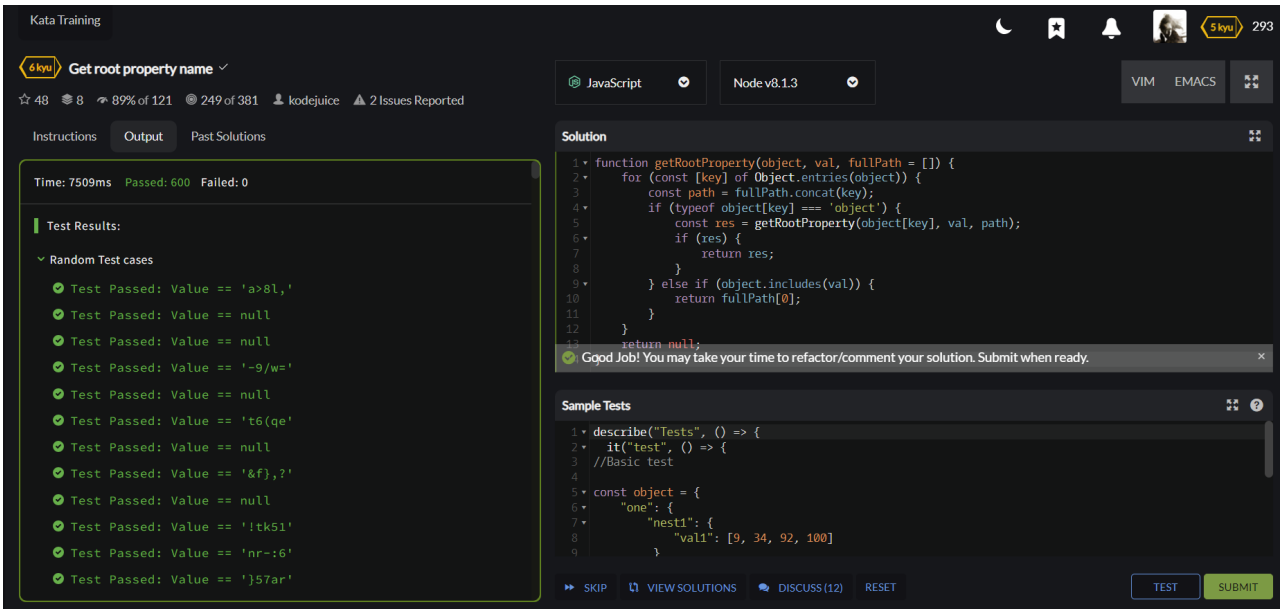
3.


Kata Training
5 kyu
293
6 kyu
Group Anagrams
43
11
94% of 112
629
dulaccc
3 Issues Reported
Instructions
Output
Past Solutions
Time: 928ms
Passed: 3
Failed: 0
Test Results:
Human cases
Light lists
Edge cases
Completed in 1ms
Superhero cases
Heavy computation that is way too long to be output (so if it fails you need to optimize the algorithm!)
Completed in 114ms
You have passed all of the tests! :)
JavaScript
Node v18.x
VIM
EMACS
Solution
const groupAnagrams = (words) => {
const groups = {};
words.forEach((word) => {
const orderedWord = word.split('').sort().join('');
!groups[orderedWord] ? groups[orderedWord] = [] : null;
groups[orderedWord].push(word);
});
return Object.values(groups);
}
Correctamundo! You may take your time to refactor/comment your solution. Submit when ready.
Sample Tests
describe("Tests", () => {
it("test", () => {
assertSimilarUnsorted(groupAnagrams(["rat", "tar", "star"]), [["rat", "tar"], ["star"]]);
});
});
SKIP
VIEW SOLUTIONS
DISCUSS (18)
RESET
TEST
SUBMIT

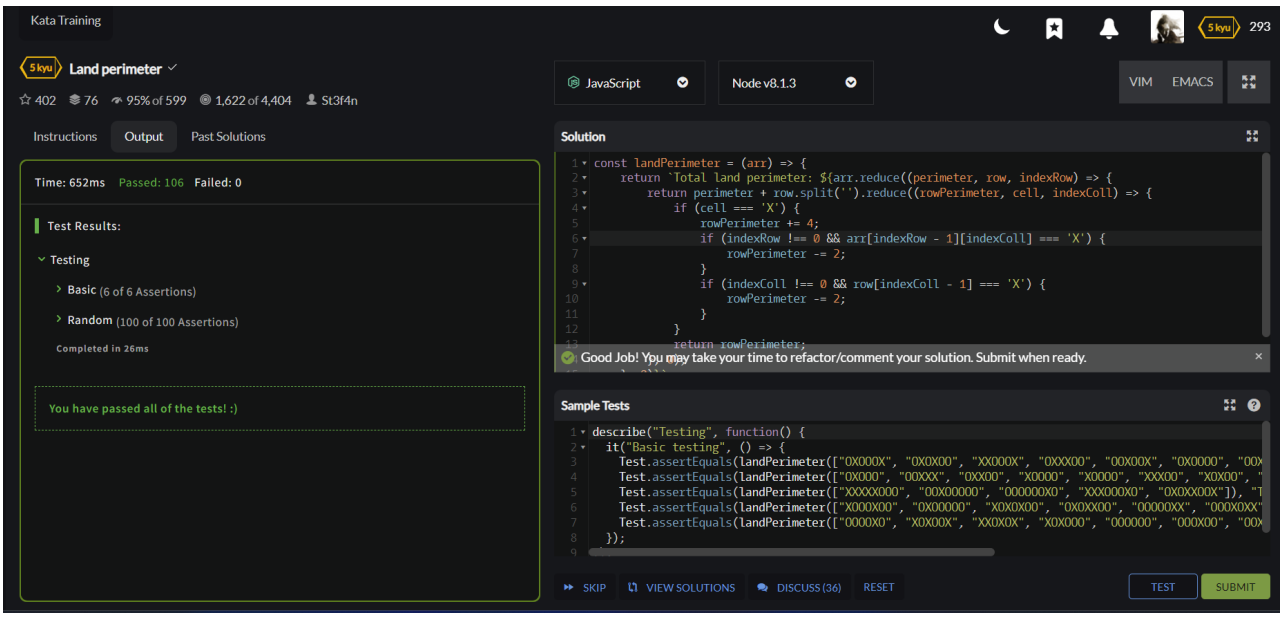
4.


Kata Training
5 kyu
293
6 kyu
Unpack delicious sausages!
19
7
92% of 107
184 of 443
WellWellWell
Instructions
Output
Past Solutions
Time: 847ms
Passed: 105
Failed: 0
Test Results:
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 2ms
Random Tests
Random test 1
Random test 2
Random test 3
Random test 4
JavaScript
Node v18.x
VIM
EMACS
Solution
const unpackSausages = (truck) => {
return truck ? Array.from(truck.map(val => val.filter(el => el.match(/^\[(.)\1{3}\]$/))).flat()
.filter((el, i) => (i + 1) % 5 !== 0 && el).join('')
.replace(/[\[\]]/gi, '')).join(' ') : "";
}
Impressive! You may take your time to refactor/comment your solution. Submit when ready.
Sample Tests
const chai = require("chai");
const assert = chai.assert;
describe("Solution", function() {
it("only lays valid sausage packages", function() {
assert.strictEqual(unpackSausages([["(----)", "[IIII]", "_HHH_"], ["IuI", "[)))))]", "zz"], ["[@@@
});
it("does not lay the 5th reward package", function() {
Submit your solution and view solutions (Cmd/Ctrl + Enter)
SKIP
VIEW SOLUTIONS
DISCUSS (9)
RESET
TEST
SUBMIT

5.


Kata Training
5 kyu 293
6 kyu Get root property name
48 8 89% of 121 249 of 381 kodejuice 2 Issues Reported
Instructions Output Past Solutions
Time: 7509ms Passed: 600 Failed: 0
Test Results:
Random Test cases
Test Passed: Value == 'a>8l,'
Test Passed: Value == null
Test Passed: Value == null
Test Passed: Value == '-9/w='
Test Passed: Value == null
Test Passed: Value == 't6(qe'
Test Passed: Value == null
Test Passed: Value == '&f},?'
Test Passed: Value == null
Test Passed: Value == '!tk51'
Test Passed: Value == 'nr-:6'
Test Passed: Value == '}57ar'
JavaScript
Node v8.1.3
VIM EMACS
Solution
function getRootProperty(object, val, fullPath = []) {
for (const [key] of Object.entries(object)) {
const path = fullPath.concat(key);
if (typeof object[key] === 'object') {
const res = getRootProperty(object[key], val, path);
if (res) {
return res;
}
} else if (object.includes(val)) {
return fullPath[0];
}
}
return null;
Good Job! You may take your time to refactor/comment your solution. Submit when ready.
Sample Tests
describe("Tests", () => {
it("test", () => {
//Basic test
const object = {
"one": {
"nest1": {
"val1": [9, 34, 92, 100]
}
SKIP VIEW SOLUTIONS DISCUSS (12) RESET
TEST SUBMIT

6.


Kata Training
5 kyu 293
5 kyu Land perimeter
402 76 95% of 599 1,622 of 4,404 St3f4n
Instructions Output Past Solutions
Time: 652ms Passed: 106 Failed: 0
Test Results:
Testing
Basic (6 of 6 Assertions)
Random (100 of 100 Assertions)
Completed in 26ms
You have passed all of the tests! :)
JavaScript
Node v8.1.3
VIM EMACS
Solution
const landPerimeter = (arr) => {
return `Total land perimeter: ${arr.reduce((perimeter, row, indexRow) => {
return perimeter + row.split('').reduce((rowPerimeter, cell, indexColl) => {
if (cell === 'X') {
rowPerimeter += 4;
if (indexRow !== 0 && arr[indexRow - 1][indexColl] === 'X') {
rowPerimeter -= 2;
}
if (indexColl !== 0 && row[indexColl - 1] === 'X') {
rowPerimeter -= 2;
}
}
return rowPerimeter;
Good Job! You may take your time to refactor/comment your solution. Submit when ready.
Sample Tests
describe("Testing", function() {
it("Basic testing", () => {
Test.assertEquals(landPerimeter(["OXOOOX", "OXOXOO", "XXOOOX", "OXXXOO", "OOXOOX", "OXOOOO", "OOX
Test.assertEquals(landPerimeter(["OXOOO", "OOXXX", "OXXOO", "XOOOO", "XOOOO", "XXXOO", "XOXOO", "
Test.assertEquals(landPerimeter(["XXXXXOOO", "OOXOOOOO", "OOOOOXOO", "XXXOOOXO", "OXOXXOOX"]), "T
Test.assertEquals(landPerimeter(["XOOOXOO", "OXOOOOO", "XOXOXOO", "OXOXXOO", "OOOOOXX", "OOOXOXX"
Test.assertEquals(landPerimeter(["OOOOXO", "XOXOOX", "XXOXOX", "XOXOOO", "OOOOOO", "OOOXOO", "OOX
});
SKIP VIEW SOLUTIONS DISCUSS (36) RESET
TEST SUBMIT