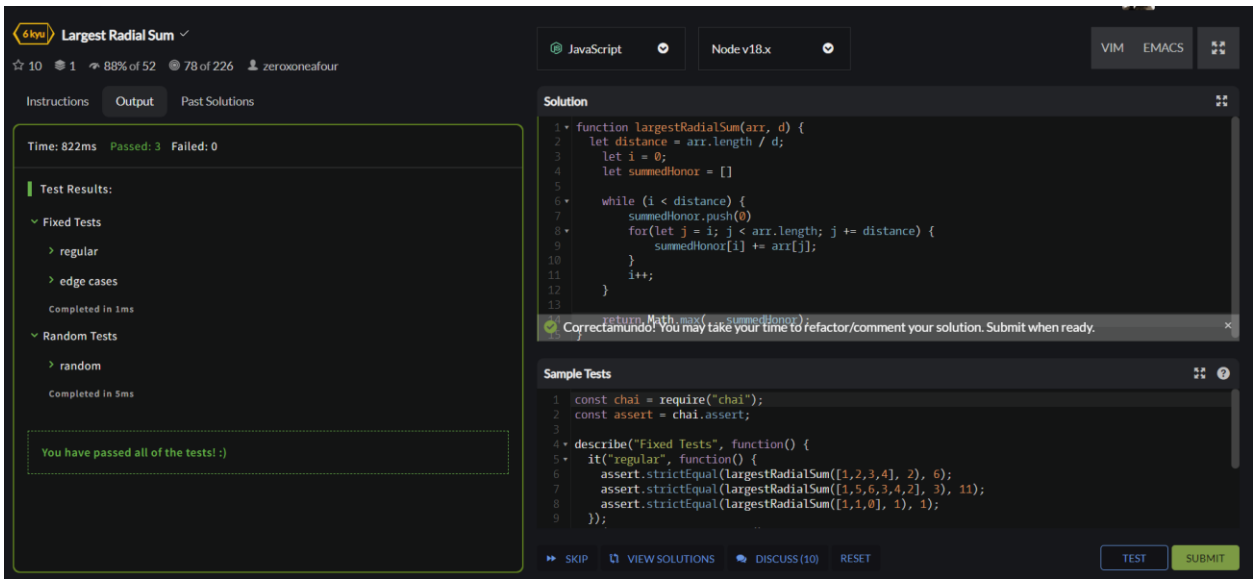
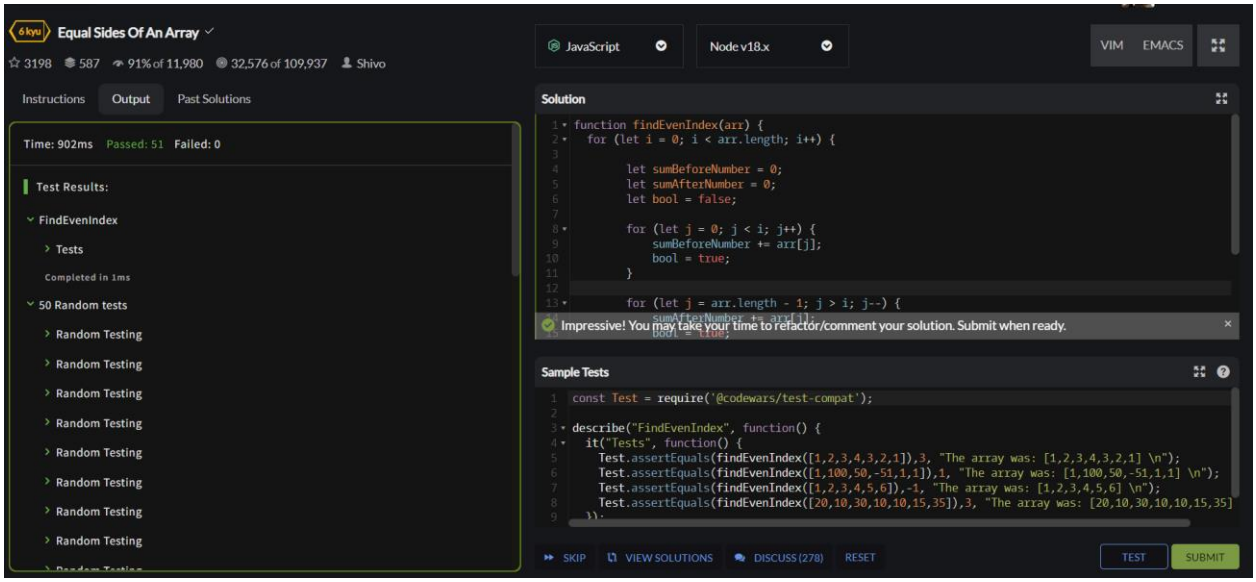
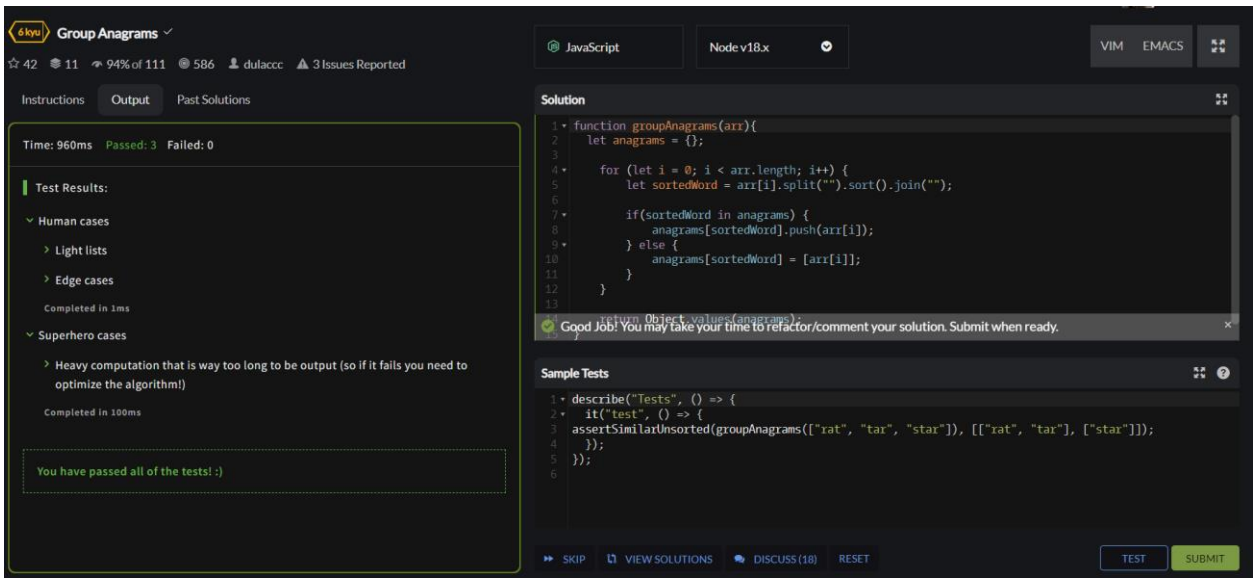
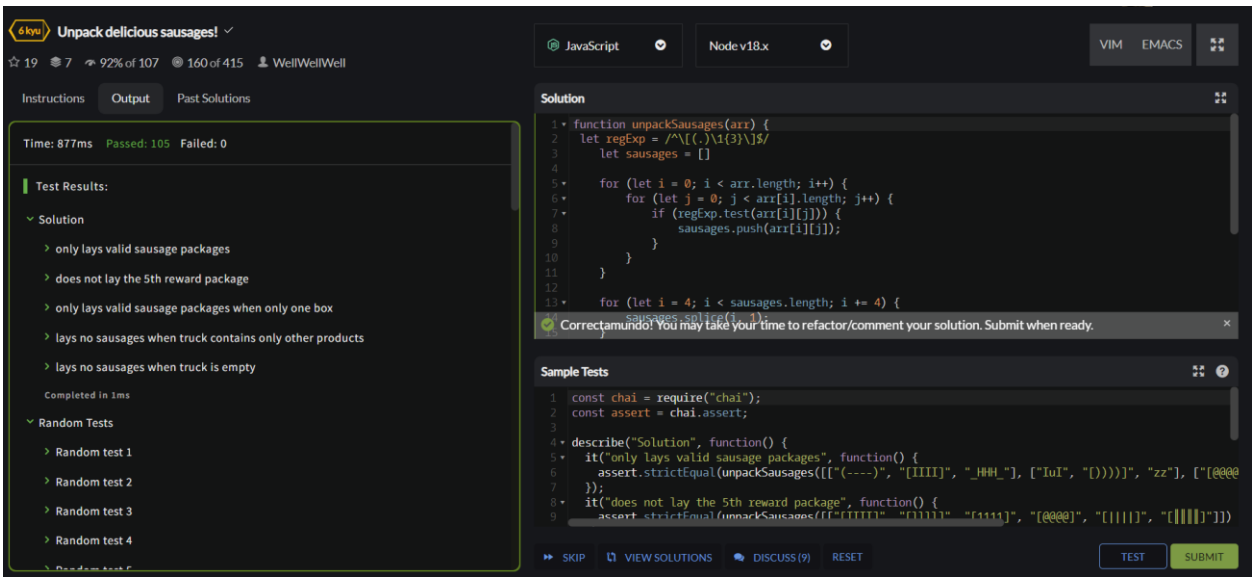
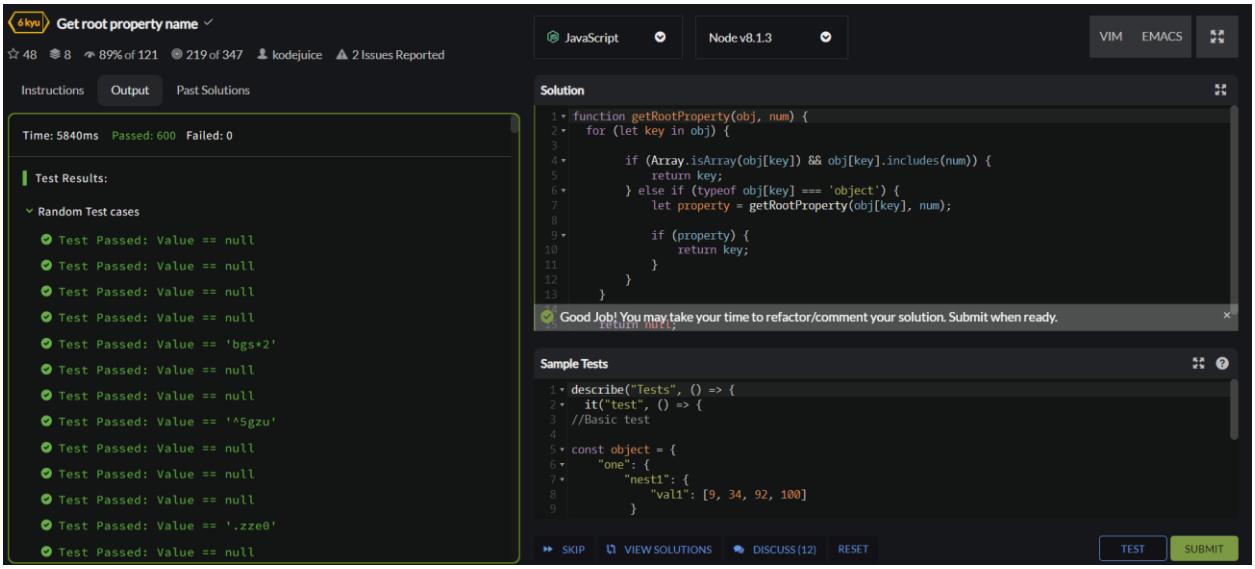
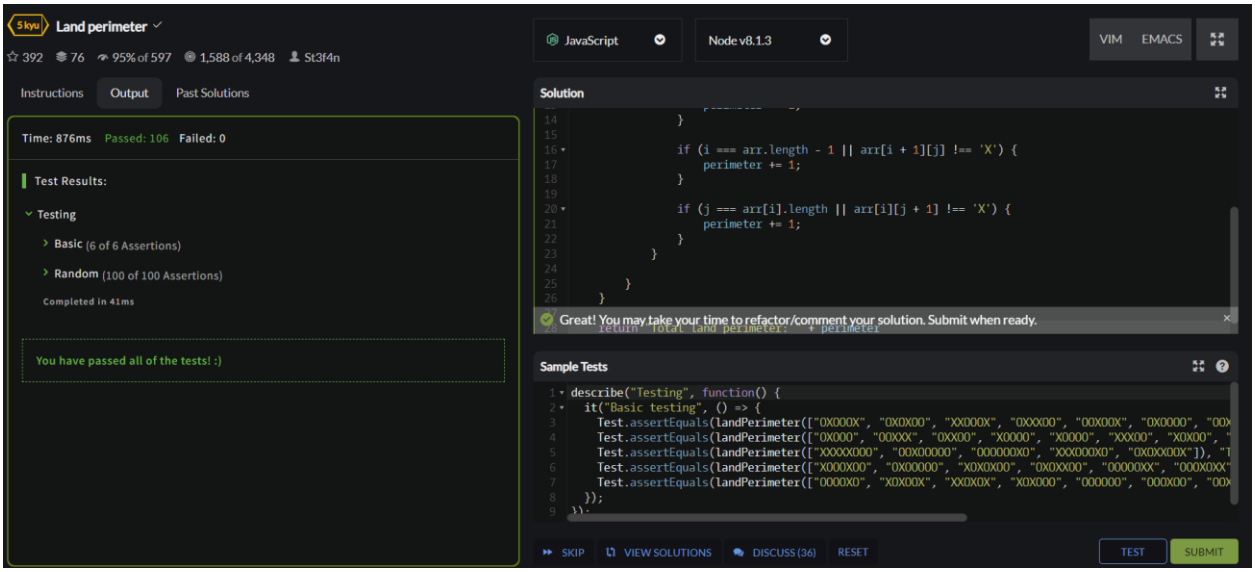

6 kyu
Largest Radial Sum
10 1 88% of 52 78 of 226 zeroxoneafour
Instructions Output Past Solutions
Time: 822ms Passed: 3 Failed: 0
Test Results:
Fixed Tests
regular
edge cases
Completed in 1ms
Random Tests
random
Completed in 5ms
You have passed all of the tests! :)
JavaScript
Node v18.x
VIM EMACS
Solution
function largestRadialSum(arr, d) {
let distance = arr.length / d;
let i = 0;
let summedHonor = []
while (i < distance) {
summedHonor.push(0)
for(let j = i; j < arr.length; j += distance) {
summedHonor[i] += arr[j];
}
i++;
}
return Math.max(...summedHonor);
}
Correctamundo! 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), 11);
assert.strictEqual(largestRadialSum([1,1,0], 1), 1);
});
SKIP VIEW SOLUTIONS DISCUSS (10) RESET
TEST SUBMIT


6 kyu
Equal Sides Of An Array
3198 587 91% of 11,980 32,576 of 109,937 Shivo
Instructions Output Past Solutions
Time: 902ms 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
Random Testing
JavaScript
Node v18.x
VIM EMACS
Solution
function findEvenIndex(arr) {
for (let i = 0; i < arr.length; i++) {
let sumBeforeNumber = 0;
let sumAfterNumber = 0;
let bool = false;
for (let j = 0; j < i; j++) {
sumBeforeNumber += arr[j];
bool = true;
}
for (let j = arr.length - 1; j > i; j--) {
sumAfterNumber += arr[j];
bool = 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


6 kyu
Group Anagrams
42 11 94% of 111 586 dulaccc 3 Issues Reported
Instructions Output Past Solutions
Time: 960ms 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 100ms
You have passed all of the tests! :)
JavaScript
Node v18.x
VIM EMACS
Solution
function groupAnagrams(arr){
let anagrams = {};
for (let i = 0; i < arr.length; i++) {
let sortedWord = arr[i].split("").sort().join("");
if(sortedWord in anagrams) {
anagrams[sortedWord].push(arr[i]);
} else {
anagrams[sortedWord] = [arr[i]];
}
}
return Object.values(anagrams);
}
Good Job! 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


6 kyu
Unpack delicious sausages!
19 7 92% of 107 160 of 415 WellWellWell
Instructions Output Past Solutions
Time: 877ms 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 1ms
Random Tests
Random test 1
Random test 2
Random test 3
Random test 4
JavaScript
Node v18.x
VIM EMACS
Solution
function unpackSausages(arr) {
let regExp = /^\[(.)\1{3}\]$/
let sausages = []
for (let i = 0; i < arr.length; i++) {
for (let j = 0; j < arr[i].length; j++) {
if (regExp.test(arr[i][j])) {
sausages.push(arr[i][j]);
}
}
}
for (let i = 4; i < sausages.length; i += 4) {
sausages.splice(i - 1, 1);
}
Correctamundo! 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() {
assert.strictEqual(unpackSausages([["[IIII]", "[1111]", "[1111]", "[@@@@]", "[|||||]", "[|||||]"]])
SKIP VIEW SOLUTIONS DISCUSS (9) RESET
TEST SUBMIT


6 kyu
Get root property name
48 8 89% of 121 219 of 347 kodejuice 2 Issues Reported
Instructions Output Past Solutions
Time: 5840ms Passed: 600 Failed: 0
Test Results:
Random Test cases
Test Passed: Value == null
Test Passed: Value == null
Test Passed: Value == null
Test Passed: Value == null
Test Passed: Value == 'bgs*2'
Test Passed: Value == null
Test Passed: Value == null
Test Passed: Value == '^5gzu'
Test Passed: Value == null
Test Passed: Value == null
Test Passed: Value == null
Test Passed: Value == '.zze0'
Test Passed: Value == null
JavaScript
Node v8.1.3
VIM EMACS
Solution
function getRootProperty(obj, num) {
for (let key in obj) {
if (Array.isArray(obj[key]) && obj[key].includes(num)) {
return key;
} else if (typeof obj[key] === 'object') {
let property = getRootProperty(obj[key], num);
if (property) {
return key;
}
}
}
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


5 kyu
Land perimeter
392 76 95% of 597 1,588 of 4,348 St3f4n
Instructions Output Past Solutions
Time: 876ms Passed: 106 Failed: 0
Test Results:
Testing
Basic (6 of 6 Assertions)
Random (100 of 100 Assertions)
Completed in 41ms
You have passed all of the tests! :)
JavaScript
Node v8.1.3
VIM EMACS
Solution
}
if (i === arr.length - 1 || arr[i + 1][j] !== 'X') {
perimeter += 1;
}
if (j === arr[i].length || arr[i][j + 1] !== 'X') {
perimeter += 1;
}
}
}
}
return "Total Land perimeter: " + perimeter
Great! 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(["XOOOXOOO", "OOXOOOOO", "OOOOOXO", "XXOOOXO", "OXOXOOX"]), "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