

First task

The Codewars Council meets at a circular table with n seats. Depending on the day d of the month, d people will be chosen as leaders of the council. These d leaders are spaced equidistantly from each other on the table, like spokes on a wheel. The leaders are chosen based on which grouping of d equidistant people have the largest summed honor. The honor of the participants for that day in the Council is given in an array `arr` of length n . Return the combined honor of the d leaders of the Council.

```
function largestRadialSum(arr, d) {
  if (1 > d && d > 31) {
    return "d must be '1 <= d <= 31'"
  }
  if (arr.length <= 0) {
    return "n must be 'n > 0'"
  }
  let groups = [];
  let amountOfGroup = arr.length / d
  for (let i = 0; i < arr.length; i++) {
    if (groups[i % amountOfGroup] === undefined) {
      groups.push([arr[i]]);
    } else {
      groups[i % amountOfGroup].push(arr[i]);
    }
  }
  let max;
  for (const group of groups) {
    let honor = 0;
    for (let i = 0; i < group.length; i++) {
      honor = honor + group[i];
    }
    if (honor > max || max === undefined) {
      max = honor;
    }
  }
  return max;
}
```

6 kyu

Largest Radial Sum

☆ 10 1 88% of 53 106 of 256 zeroxonefour

Instructions Output Past Solutions

Time: 832ms Passed: 3 Failed: 0

Test Results:

Fixed Tests

regular

edge cases

Completed in 2ms

Random Tests

random

Completed in 8ms

You have passed all of the tests! :)

JavaScript Node v18.x VIM EMACS

Solution

```
13 } else {
14   groups[i % amountOfGroup].push(arr[i]);
15 }
16 }
17 let max;
18 for (const group of groups) {
19   let honor = 0;
20   for (let i = 0; i < group.length; i++) {
21     honor = honor + group[i];
22   }
23   if (honor > max || max === undefined) {
24     max = honor;
25   }
26 }
27 return max;
```

Great! 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([1,1,0], 1), 1);
9   });
10   it("edge cases" function() {
```

SKIP VIEW SOLUTIONS DISCUSS(10) RESET

TEST SUBMIT

Second task

You are going to be given an array of integers. Your job is to take that array and find an index N where the sum of the integers to the left of N is equal to the sum of the integers to the right of N. If there is no index that would make this happen, return -1.

```
function findEvenIndex(arr)
{
  let left_sum;
  let right_sum;

  for (let n = 0; n < arr.length; n++) {
    left_sum = Sum(arr.slice(0, n));
    right_sum = Sum(arr.slice(n + 1));
    if (left_sum === right_sum) {
      return n;
    }
  }
  return -1;

  function Sum(arr) {
    let sum = 0;
    for (let i = 0; i < arr.length; i++) {
      sum = sum + arr[i];
    }
    return sum;
  }
}
```

6 kyu

Equal Sides Of An Array

☆ 3205 590 91% of 11,988 32,678 of 110,230 Shivo

Instructions Output

Time: 853ms 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

```
1 function findEvenIndex(arr)
2 {
3   let left_sum;
4   let right_sum;
5
6   for (let n = 0; n < arr.length; n++) {
7     left_sum = Sum(arr.slice(0, n));
8     right_sum = Sum(arr.slice(n + 1));
9     if (left_sum === right_sum) {
10      return n;
11    }
12  }
13  return -1;
14
15  function Sum(arr) {
16    let sum = 0;
```

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

Sample Tests

```
1 const Test = require('@codewars/test-compat');
2
3 describe("FindEvenIndex", function() {
4   it("Tests", function() {
5     Test.assertEquals(findEvenIndex([1,2,3,4,3,2,1]),3, "The array was: [1,2,3,4,3,2,1] \n");
6     Test.assertEquals(findEvenIndex([1,100,50,-51,1,1]),1, "The array was: [1,100,50,-51,1,1] \n");
7     Test.assertEquals(findEvenIndex([1,2,3,4,5,6]),-1, "The array was: [1,2,3,4,5,6] \n");
8     Test.assertEquals(findEvenIndex([20,10,30,10,10,15,35]),3, "The array was: [20,10,30,10,10,15,35] \n");
9   });
10
11 });
```

SKIP UNLOCK SOLUTIONS DISCUSS (278) RESET TEST SUBMIT

Third task

Your job is to group the words in anagrams.

What is an anagram ?

star and tsar are anagram of each other because you can rearrange the letters for star to obtain tsar.

```
function groupAnagrams(words){
  let new_arr = [];
  let arrays = {};
  for (let i = 0; i < words.length; i++) {
    sortedString = words[i].split("").sort().join("");
    if (!(sortedString in arrays)) {
      arrays[sortedString] = [];
      arrays[sortedString].push(words[i]);
    } else {
      arrays[sortedString].push(words[i]);
    }
  }
  for (const key in arrays) {
    new_arr.push(arrays[key]);
  }
  return new_arr;
}
```

6kyu

Group Anagrams ✓

43 11 94% of 112 617 dulaccc 3 Issues Reported

Instructions Output Past Solutions

Time: 948ms Passed: 3 Failed: 0

Test Results:

Human cases

- Light lists
- Edge cases

Completed in 2ms

Superhero cases

- Heavy computation that is way too long to be output (so if it fails you need to optimize the algorithm!)

Completed in 118ms

You have passed all of the tests! :)

JavaScript Node v18.x VIM EMACS

Solution

```
1 function groupAnagrams(words){
2   let new_arr = [];
3   let arrays = {};
4   for (let i = 0; i < words.length; i++) {
5     sortedString = words[i].split("").sort().join("");
6     if (!(sortedString in arrays)) {
7       arrays[sortedString] = [];
8       arrays[sortedString].push(words[i]);
9     } else {
10      arrays[sortedString].push(words[i]);
11    }
12  }
13  for (const key in arrays) {
14    new_arr.push(arrays[key]);
15  }
16  return new_arr;
}
```

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

Sample Tests

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

SKIP VIEW SOLUTIONS DISCUSS (18) RESET

TEST SUBMIT

Fourth task

Unpack delicious sausages!

A food delivery truck carrying boxes of delicious sausages has arrived and it's your job to unpack them and put them in the store's display counter.

```
function unpackSausages(truck) {
  let sausages = "";
  let counter = 1;
  for (const box of truck) {
    for (let package of box) {
      if (package.indexOf('I') !== -1 && package.lastIndexOf('I') !== -1) {
        package = package.slice(0, 0) + package.slice(1, package.length - 1);
        let typeOfPackage = package[0];
        for (let i = 0; i < package.length; i++) {
          if (package[i] !== typeOfPackage || package.length !== 4) {
            typeOfPackage = 'damaged';
            break;
          }
        }
        if (typeOfPackage !== 'damaged') {
          if (counter % 5 !== 0) {
            sausages = sausages + `${package[0]}`.repeat(4);
          }
          counter++;
        }
      }
    }
  }
  return sausages.trim();
}
```

Unpack delicious sausages! ✓

☆ 19 🗨 7 🔄 92% of 107 📊 179 of 436 👤 WellWellWell

Instructions Output Past Solutions

Time: 809ms 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

Random test 5

JavaScript Node v18.x VIM EMACS

Solution

```
10 if (package[i] !== typeOfPackage || package.length !== 4) {
11   typeOfPackage = 'damaged';
12   break;
13 }
14 }
15 if (typeOfPackage !== 'damaged') {
16   if (counter % 5 !== 0) {
17     sausages = sausages + `${package[0]}`.repeat(4);
18   }
19   counter++;
20 }
21 }
22 }
23 }
24 return sausages.trim();
```

🟢 Outstanding! 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(["(----)", "[IIIII]", "_HHH_", ["IuI", "(I)"]]), "zz", ["[@@@@"
7   ]);
8   it("does not lay the 5th reward package", function() {
9     assert.strictEqual(unpackSausages(["[IIIII]", "[1111]", "[1111]", "[1111]", "[1111]"]),
10
```

⏮ SKIP 🔗 VIEW SOLUTIONS 💬 DISCUSS (9) RESET

TEST SUBMIT

Fifth task

Цьому завданні представлено 6 задачок різного рівня складності, проте матеріалів лекції має

Given an object of likely nested objects, where the final element is an array containing positive integers, write a function that returns the name of the root property that a particular integer lives in.

visualization:

```
function getRootProperty(object, val) {
  let root;
  root = IsObject(object);
  if (root === undefined) {
    root = null
  }

  function IsObject(object) {
    for (const key in object) {
      if (Array.isArray(object) && object.includes(val)) {
        return object;
      } else if (typeof object[key] === 'object') {
        root = IsObject(object[key])
        if (root !== undefined) {
          return key;
        }
      }
    }
  }
  return root;
}
```

6kyu

Get root property name ✓

☆ 48 🏆 8 🔄 89% of 121 🕒 241 of 371 👤 kodejuice 🚩 2 Issues Reported

Instructions Output Past Solutions

Time: 6048ms Passed: 600 Failed: 0

Test Results:

Random Test cases

- Test Passed: Value == null
- Test Passed: Value == '-q).g'
- Test Passed: Value == null
- Test Passed: Value == null
- Test Passed: Value == null
- Test Passed: Value == null
- Test Passed: Value == null
- Test Passed: Value == null
- Test Passed: Value == '4x4h'
- Test Passed: Value == null
- Test Passed: Value == null

JavaScript

Node v8.1.3

VIM EMACS

Solution

```
1 function getRootProperty(object, val) {
2   let root;
3   root = IsObject(object);
4   if (root === undefined) {
5     root = null
6   }
7
8   function IsObject(object) {
9     for (const key in object) {
10      if (Array.isArray(object) && object.includes(val)) {
11        return object;
12      } else if (typeof object[key] === 'object') {
13        root = IsObject(object[key])
14        if (root !== undefined) {
15          return key;
16        }
17      }
18    }
19  }
20  return root;
21 }
```

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

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 VIEW SOLUTIONS DISCUSS (40) RESET

TEST SUBMIT

Sixth task

Given an array `arr` of strings, complete the function by calculating the total perimeter of all the islands. Each piece of land will be marked with 'X' while the water fields are represented as 'O'. Consider each tile being a perfect 1 x 1 piece of land. Some examples for better visualization:

```
function landPerimeter(arr) {
  let totalPerimeter = 0;
  let visitedIslands = new Set();

  for (let i = 0; i < arr.length; i++) {
    for (let j = 0; j < arr[i].length; j++) {
      let PerimeterOfIsland = 0;
      let islandPerimeter = 0;
      FindIsland(i, j);
      totalPerimeter = totalPerimeter + PerimeterOfIsland;
    }
  }

  function FindIsland(row, column) {
    if (arr[row][column] === "X" && !visitedIslands.has(`${row},${column}`)) {
      PerimeterOfIsland = PerimeterOfIsland + 4;
      visitedIslands.add(`${row},${column}`);

      if (column + 1 < arr[i].length && arr[row][column + 1] === "X") {
        PerimeterOfIsland = PerimeterOfIsland - 2;
        FindIsland(row, column + 1);
      }
      if (row + 1 < arr.length && arr[row + 1][column] === "X") {
        PerimeterOfIsland = PerimeterOfIsland - 2;
        FindIsland(row + 1, column);
      }
    }
  }

  return `Total land perimeter: ${totalPerimeter}`;
}
```

5 kyu

Land perimeter ✓

☆ 392 🌟 76 📈 95% of 597 🕒 1,611 of 4,379 👤 St3f4n

Instructions

Output

Past Solutions

Time: 836ms Passed: 106 Failed: 0

Test Results:

Testing

Basic (6 of 6 Assertions)

Random (100 of 100 Assertions)

Completed in 52ms

You have passed all of the tests! :)

JavaScript

Node v8.1.3

VIM

EMACS

⛶

⛶

Solution

```
1 function landPerimeter(arr) {
2   let totalPerimeter = 0;
3   let visitedIslands = new Set();
4
5   for (let i = 0; i < arr.length; i++) {
6     for (let j = 0; j < arr[i].length; j++) {
7       let PerimeterOfIsland = 0;
8       let islandPerimeter = 0;
9       FindIsland(i, j);
10      totalPerimeter = totalPerimeter + PerimeterOfIsland;
11
12
13      function FindIsland(row, column) {
14        if (arr[row][column] === "X" && !visitedIslands.has(`${row},${column}`)) {
15          PerimeterOfIsland = PerimeterOfIsland + 1;
16          visitedIslands.add(`${row},${column}`);
17          FindIsland(row + 1, column);
18          FindIsland(row - 1, column);
19          FindIsland(row, column + 1);
20          FindIsland(row, column - 1);
21        }
22      }
23    }
24  }
25  return totalPerimeter;
26}
```

Correctamundo! 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", "0XX00", "00X00X", "0X0000", "00X
4     Test.assertEquals(landPerimeter(["0X000", "00XX", "0X0X0", "X0000", "X0000", "XX00", "X0X00", "X
5     Test.assertEquals(landPerimeter(["XXXXXX00", "00X00000", "000000X0", "XX0000X", "0X0XX00X"]), "1
6     Test.assertEquals(landPerimeter(["X000X00", "0X00000", "X0X0X0", "0X0X00", "00000X", "000X00X",
7     Test.assertEquals(landPerimeter(["0000X0", "X0X00X", "XX0X0X", "X0X00", "000000", "000X00", "00X
8   });
9 });
```

SKIP

VIEW SOLUTIONS

DISCUSS (36)

RESET

TEST

SUBMIT