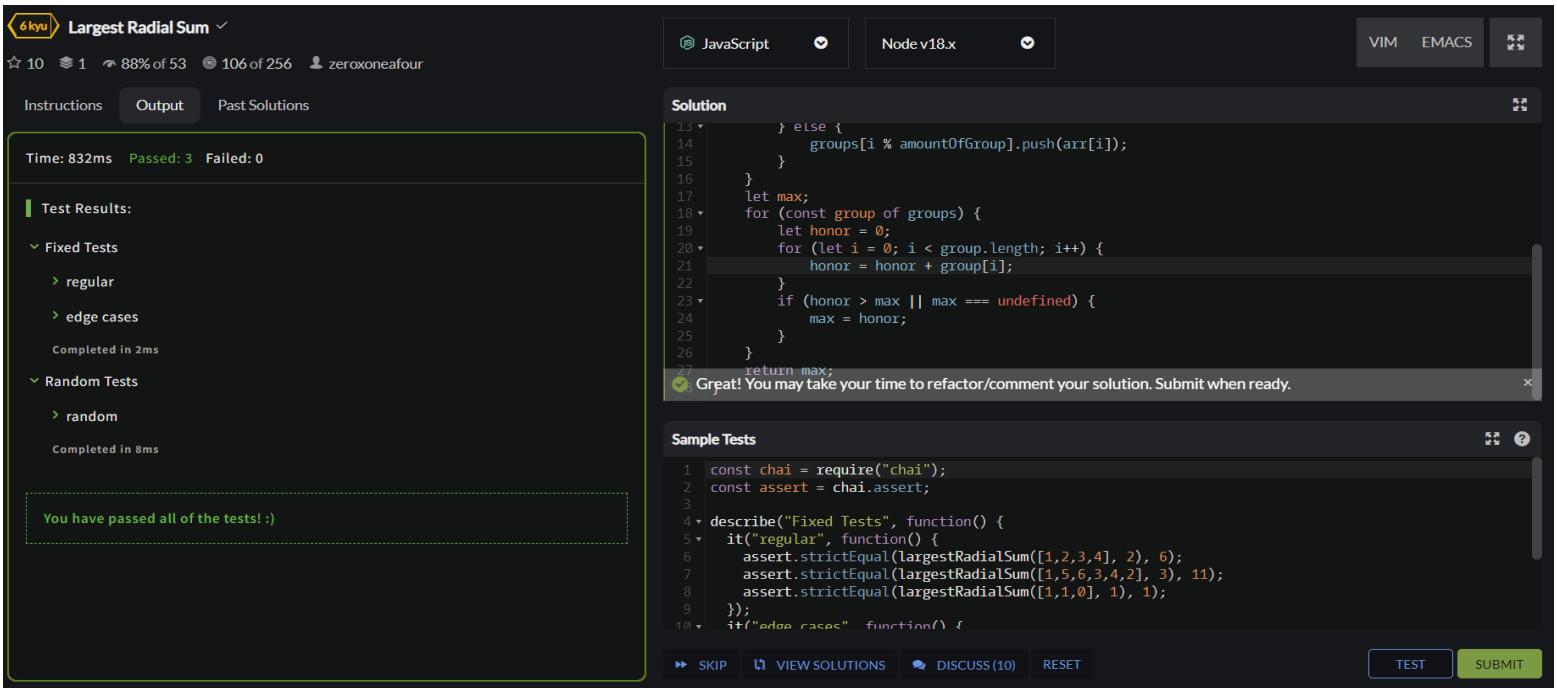
# First task

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;
}
```



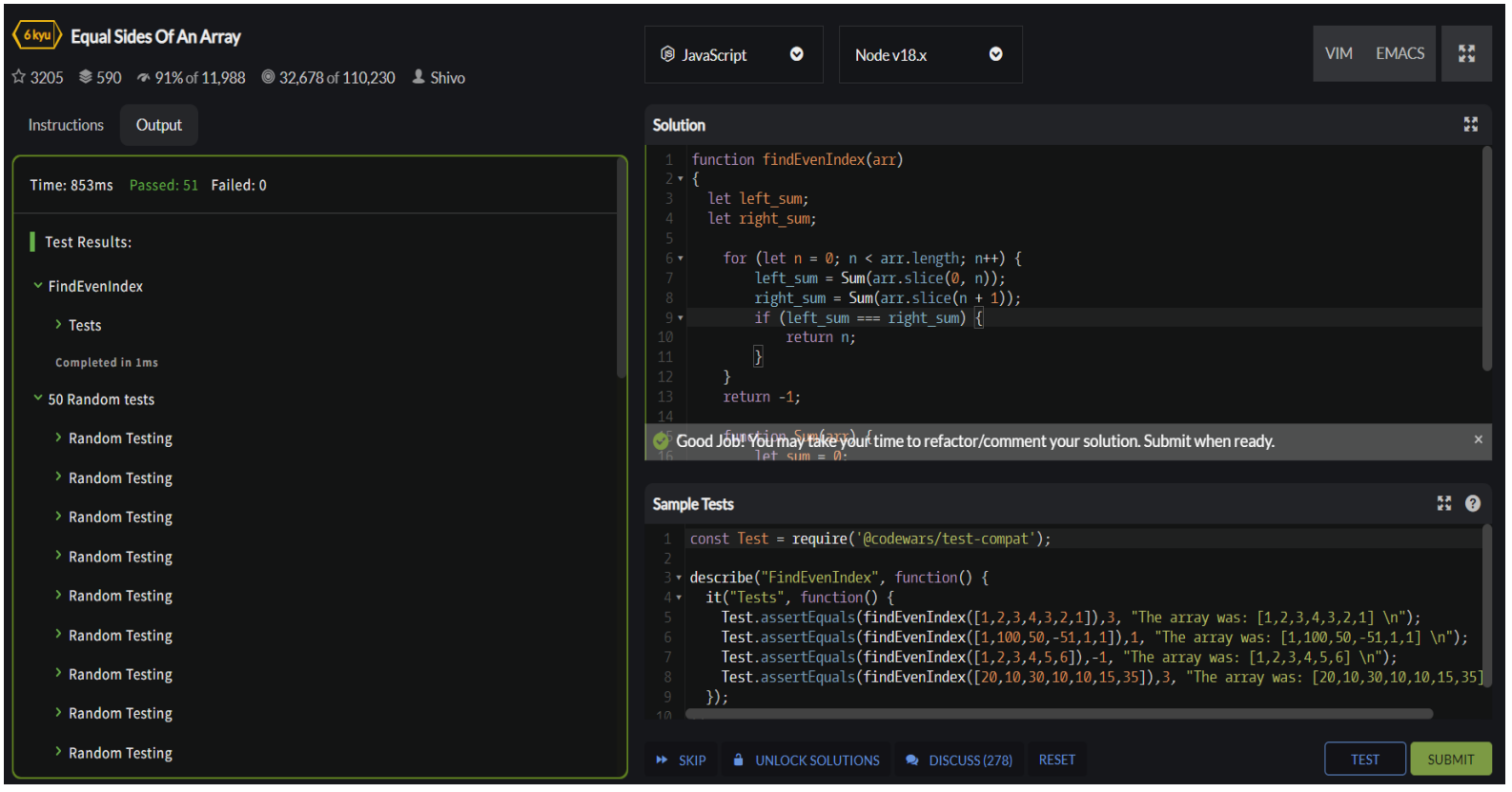
## Second task

**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;
    }
}
```

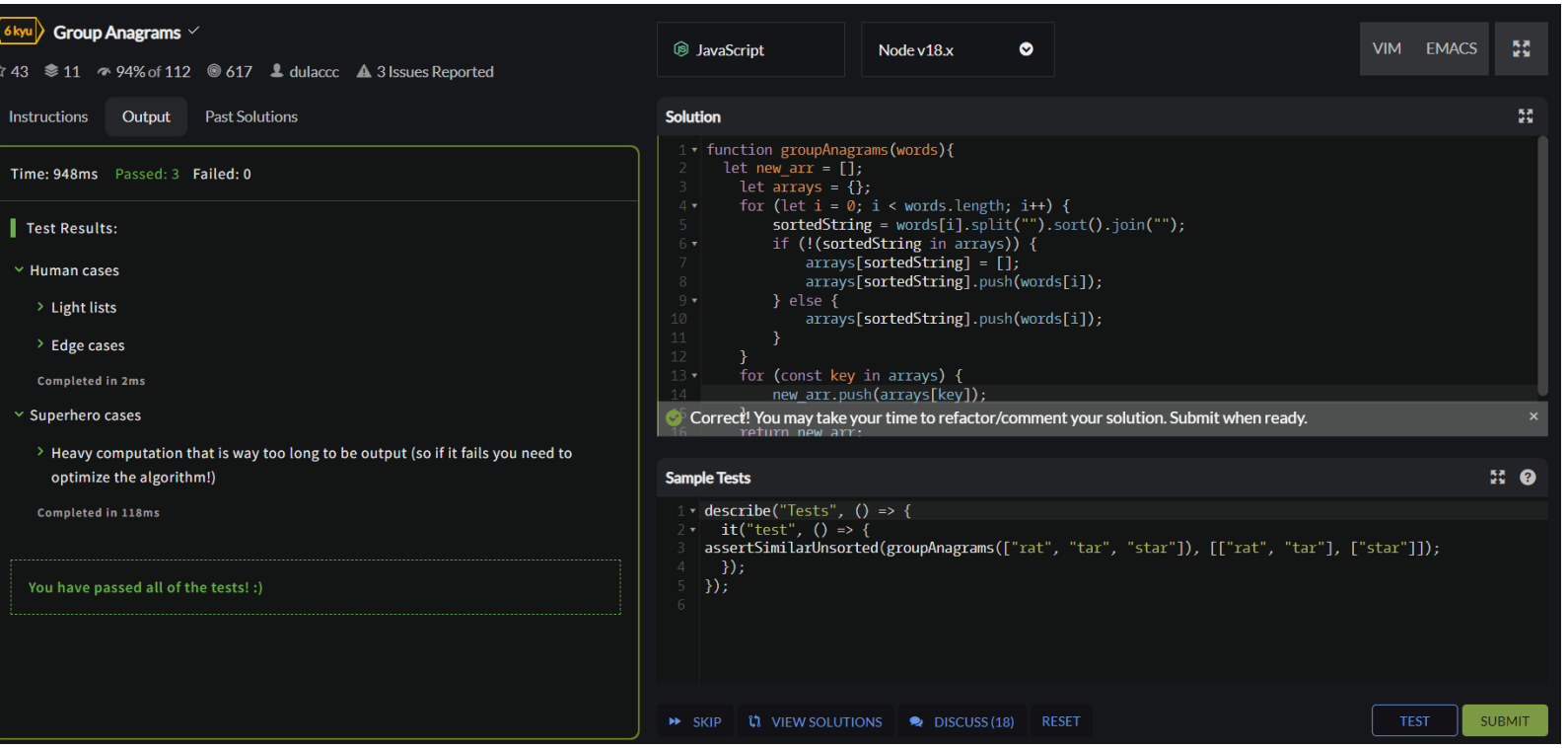


# Third task

**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;
}
```



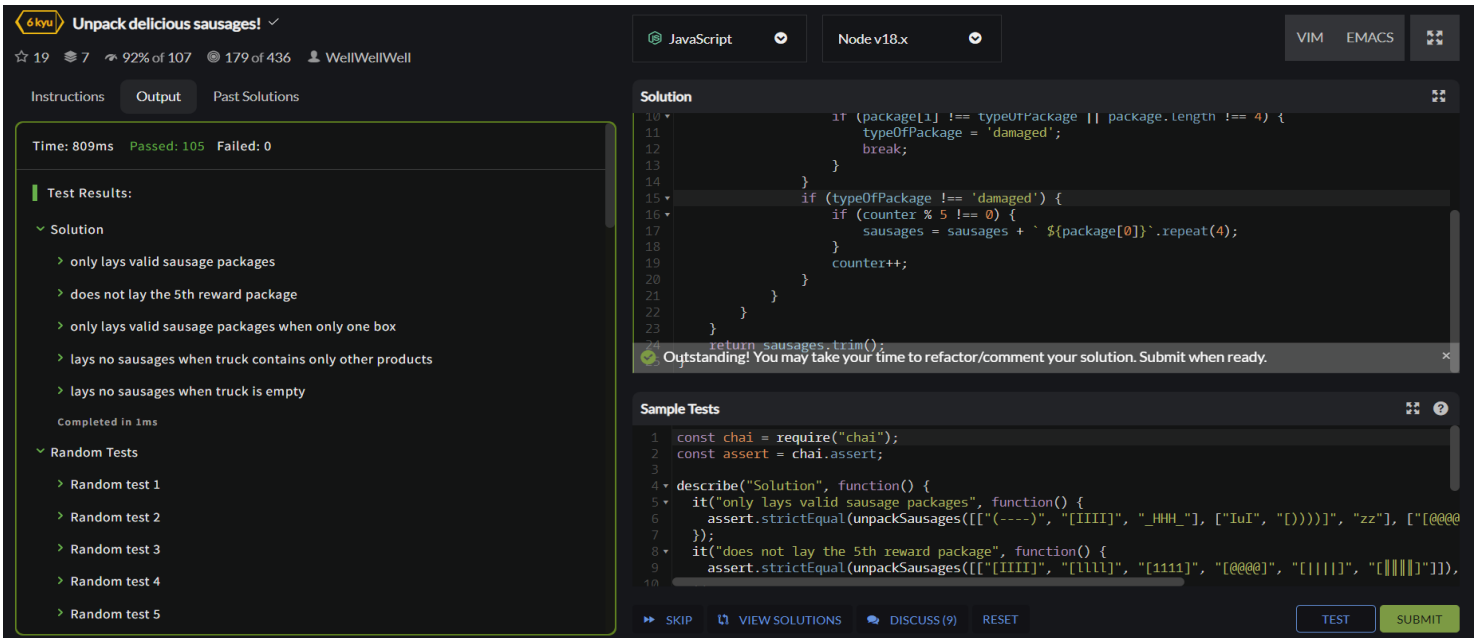
## Fourth task

**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('[') !== -1 && package.lastIndexOf(']') !== -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();
}
```



# Fifth task

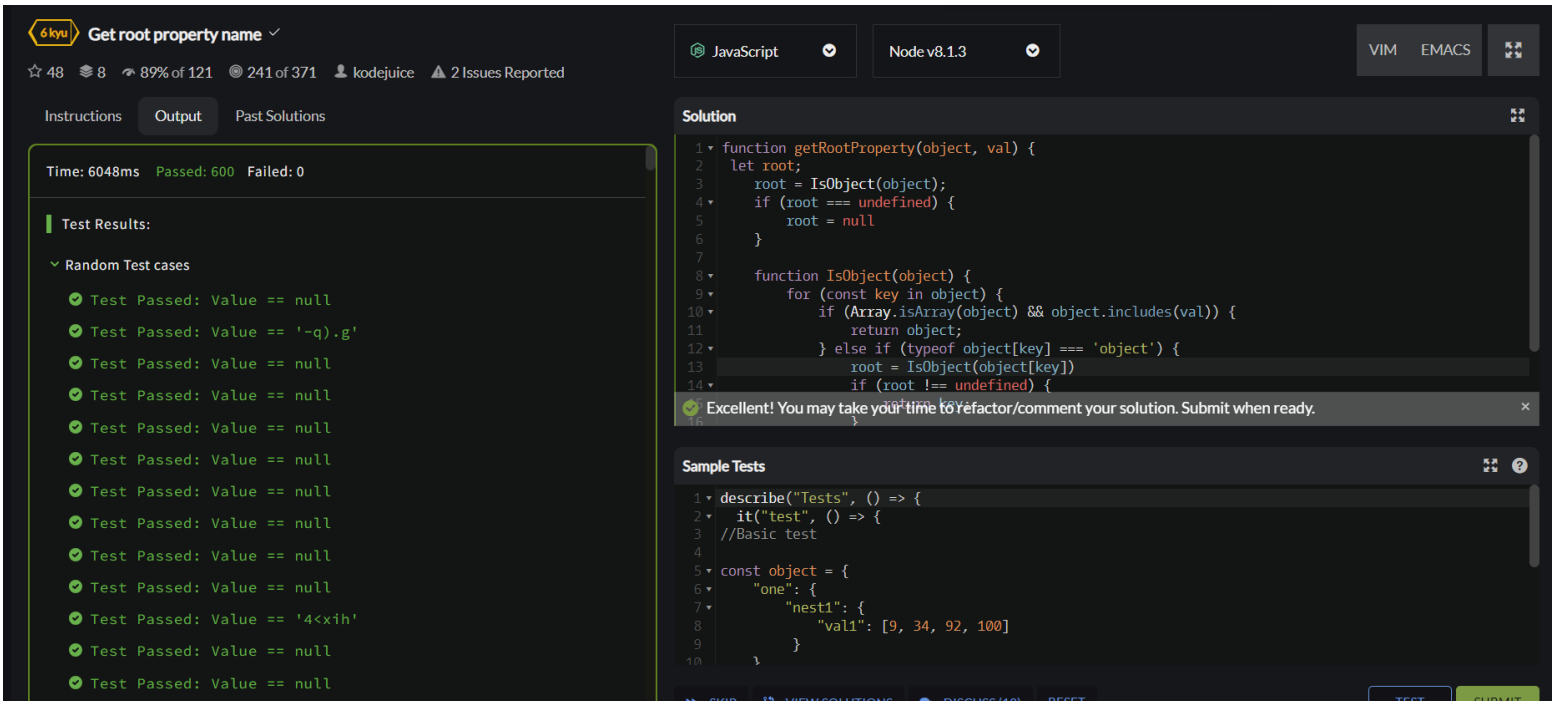
ньому завданні представлено 6 задачок різного рівня складності, проте матеріалів лекції має

**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;
}
```



## 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}`;
}
```