

JAVASCRIPT-CODE CHALLENGES

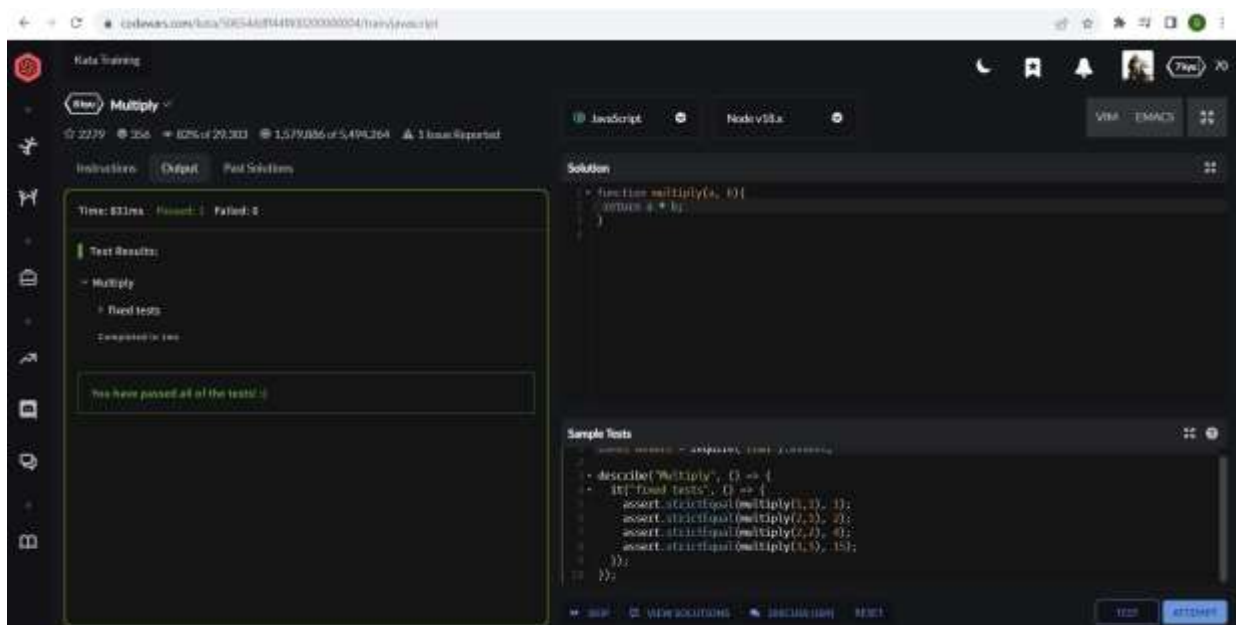
1) Multiply

This code does not execute properly. Try to figure out why.

```
function multiply(a, b){  
  a * b  
}
```

Corrected code:

```
function multiply(a, b){  
  return a * b  
}
```



2) Get Planet Name By ID

Example (Input --> Output):

```
function getPlanetName(id){  
  var name;  
  switch(id){  
    case 1:  
      name = 'Mercury';  
    case 2:  
      name = 'Venus'
```

```
case 3:
    name = 'Earth'
case 4:
    name = 'Mars'
case 5:
    name = 'Jupiter'
case 6:
    name = 'Saturn'
case 7:
    name = 'Uranus'
case 8:
    name = 'Neptune'
}

return name;
}
```

Corrected code:

```
function getPlanetName(id){
    var name;
    switch(id){
        case 1:
            name = 'Mercury';
            break;
        case 2:
            name = 'Venus'
            break;
        case 3:
            name = 'Earth'
            break;
        case 4:
            name = 'Mars'
```

```

    break;

case 5:

    name = 'Jupiter'

    break;

case 6:

    name = 'Saturn'

    break;

case 7:

    name = 'Uranus'

    break;

case 8:

    name = 'Neptune'

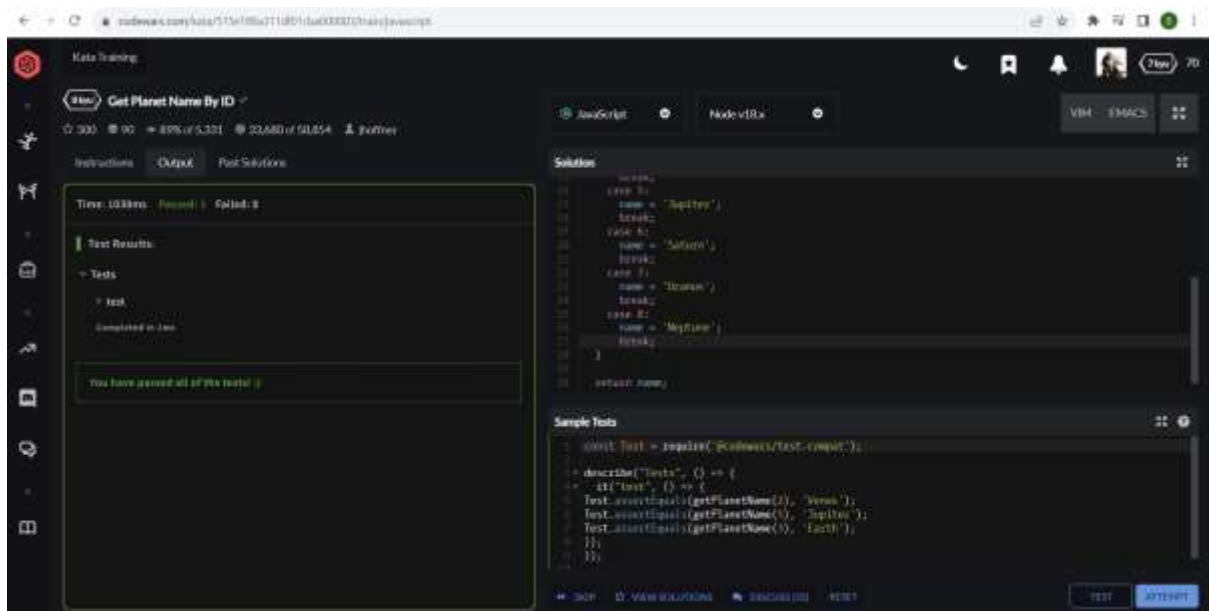
    break;

}

return name;

}

```



3) Reversed Strings

```

function solution(str){

  const arr=str.split("");

```

```

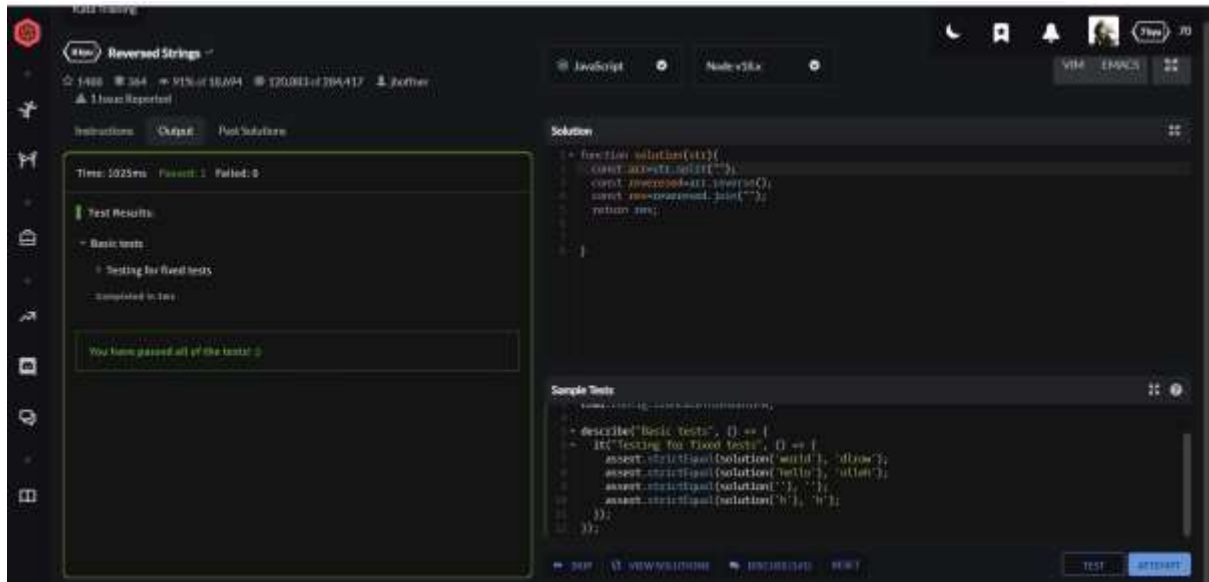
const reversed=arr.reverse();

const res=reversed.join("");

return res;

}

```



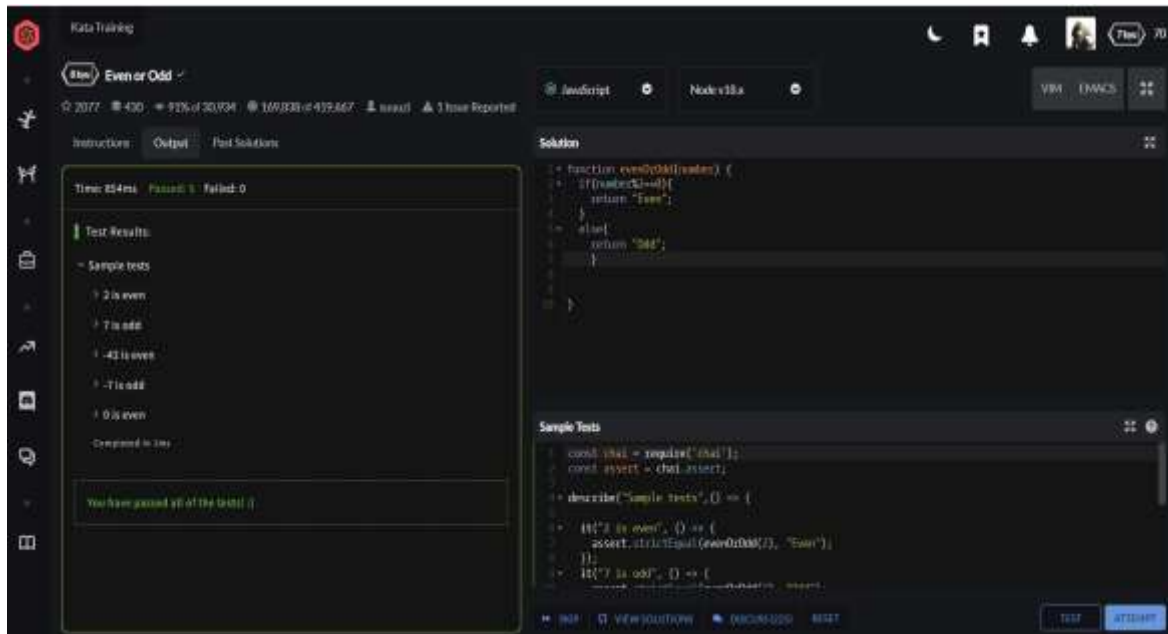
4) Even or Odd

Create a function that takes an integer as an argument and returns "Even" for even numbers or "Odd" for odd numbers.

```

function evenOrOdd(number) {
  if(number%2==0){
    return "Even";
  }
  else{
    return "Odd";
  }
}

```



5) Vowel Count

Return the number (count) of vowels in the given string. We will consider a, e, i, o, u as vowels for this Kata (but not y). The input string will only consist of lower case letters and/or spaces.

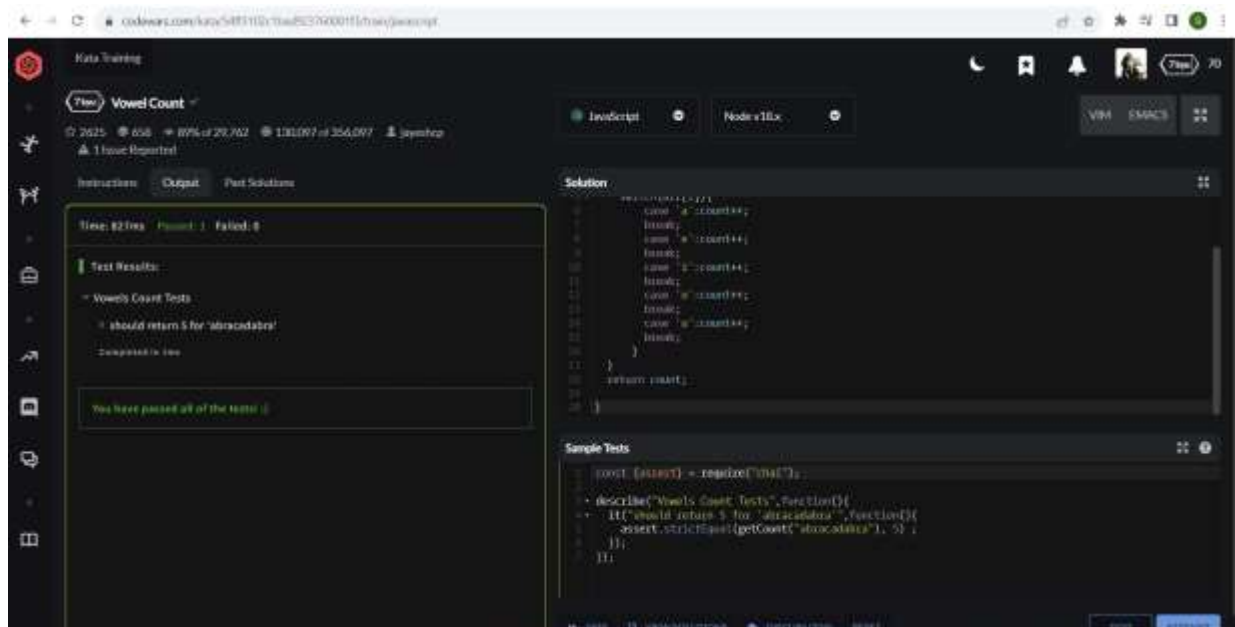
```
function getCount(str) {
  let count=0;
  const arr=str.split("");
  for(let i=0;i<arr.length;i++){
    switch(arr[i]){
      case 'a':count++;
      break;
      case 'e':count++;
      break;
      case 'i':count++;
      break;
      case 'o':count++;
      break;
      case 'u':count++;
      break;
```

```

    }
}

return count;
}

```



6) Jenny's secret message

Jenny has written a function that returns a greeting for a user. However, she's in love with Johnny, and would like to greet him slightly different. She added a special case to her function, but she made a mistake. Can you help her?

```

function greet(name){
    return "Hello, " + name + "!";

    if(name === "Johnny")
        return "Hello, my love!";
}

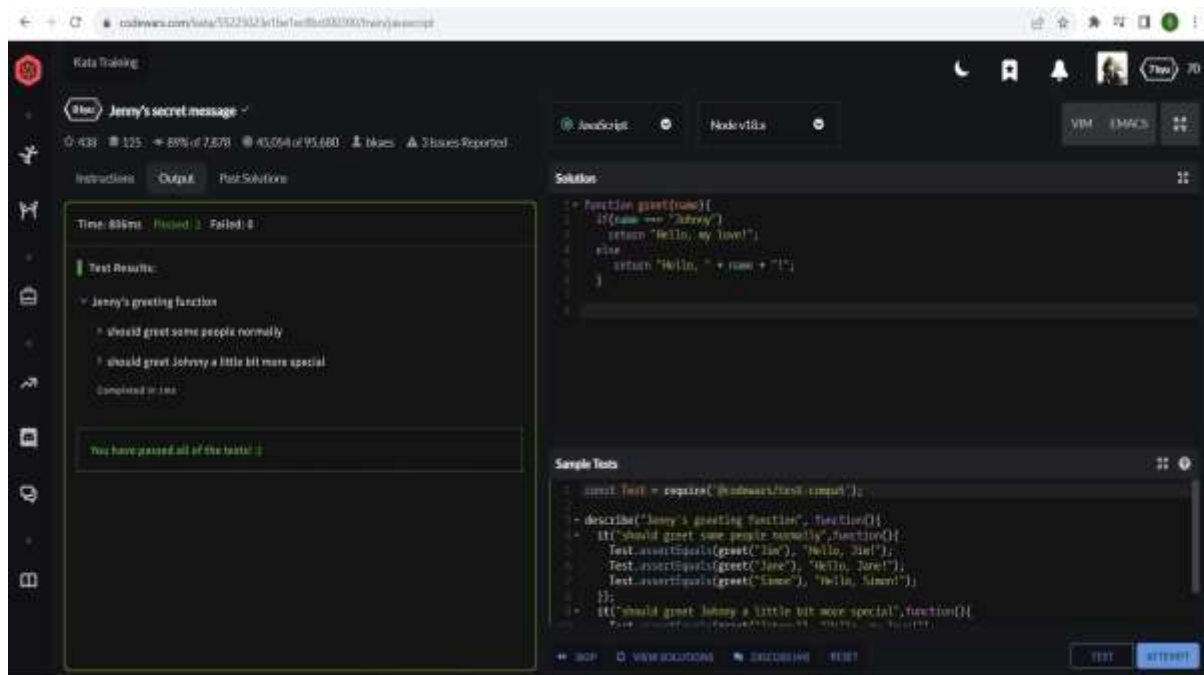
```

Corrected code:

```

function greet(name){
    if(name === "Johnny")
        return "Hello, my love!";
    else
        return "Hello, " + name + "!";
}

```



7) Is n divisible by x and y?

Create a function that checks if a number n is divisible by two numbers x AND y . All inputs are positive, non-zero numbers.

function isDivisible(n, x, y) {

if (n%x==0 && n%y==0){

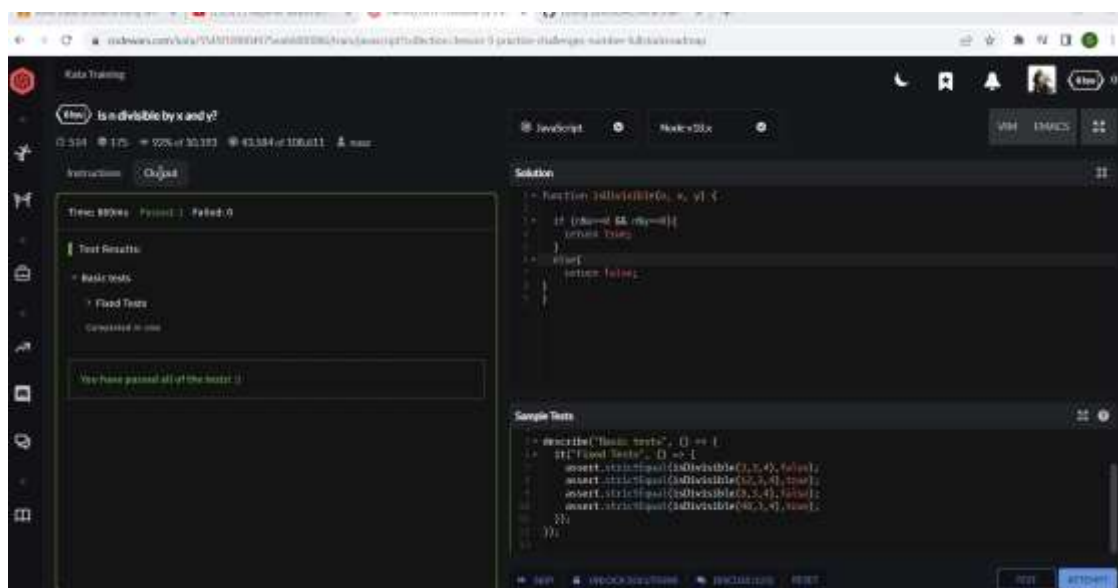
return true;

}

else{

return false;}

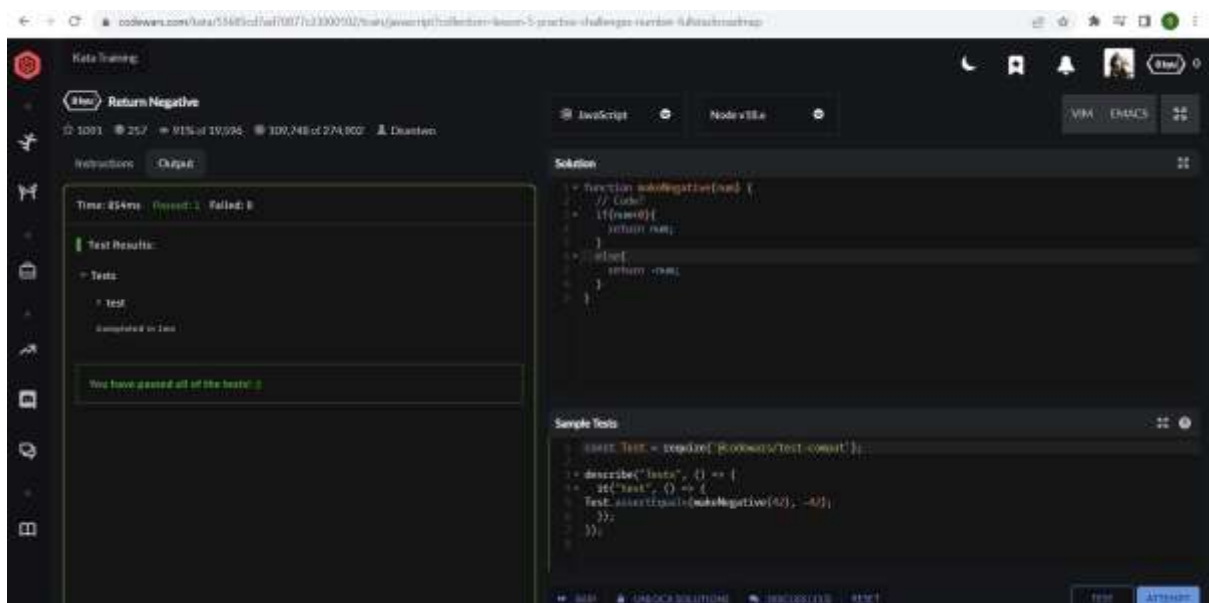
}



8) Return Negative

In this simple assignment you are given a number and have to make it negative. But maybe the number is already negative?

```
function makeNegative(num) {  
  // Code?  
  if(num<0){  
    return num;  
  }  
  else{  
    return -num;  
  }  
}
```



9) Find the smallest integer in the array

Given an array of integers your solution should find the smallest integer.

For example:

Given [34, 15, 88, 2] your solution will return 2

Given [34, -345, -1, 100] your solution will return -345

You can assume, for the purpose of this kata, that the supplied array will not be empty.

```
class SmallestIntegerFinder {  
  findSmallestInt(args) {
```



```

let small=args[0];

for (let i=0;i<args.length;i++){

    if (args[i]<small){

        small=args[i];

    }

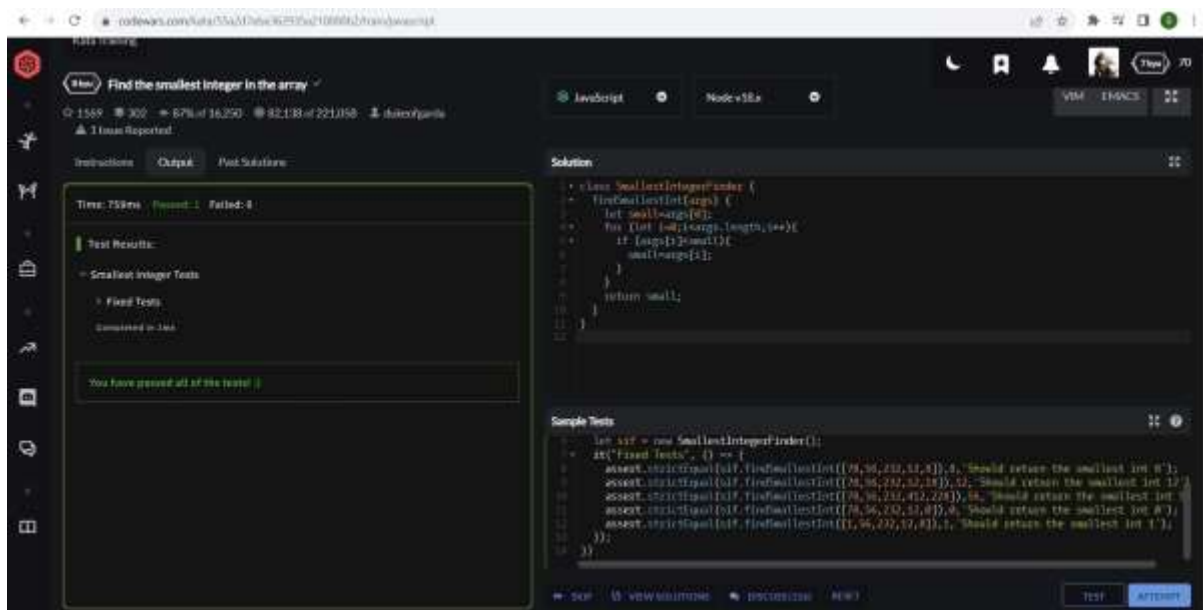
}

return small;

}

}

```



10) Grasshopper – Summation

Write a program that finds the summation of every number from 1 to num. The number will always be a positive integer greater than 0.

```

var summation = function (num) {

    // Code here

    let sum=0;

    for(let i=1;i<=num;i++){

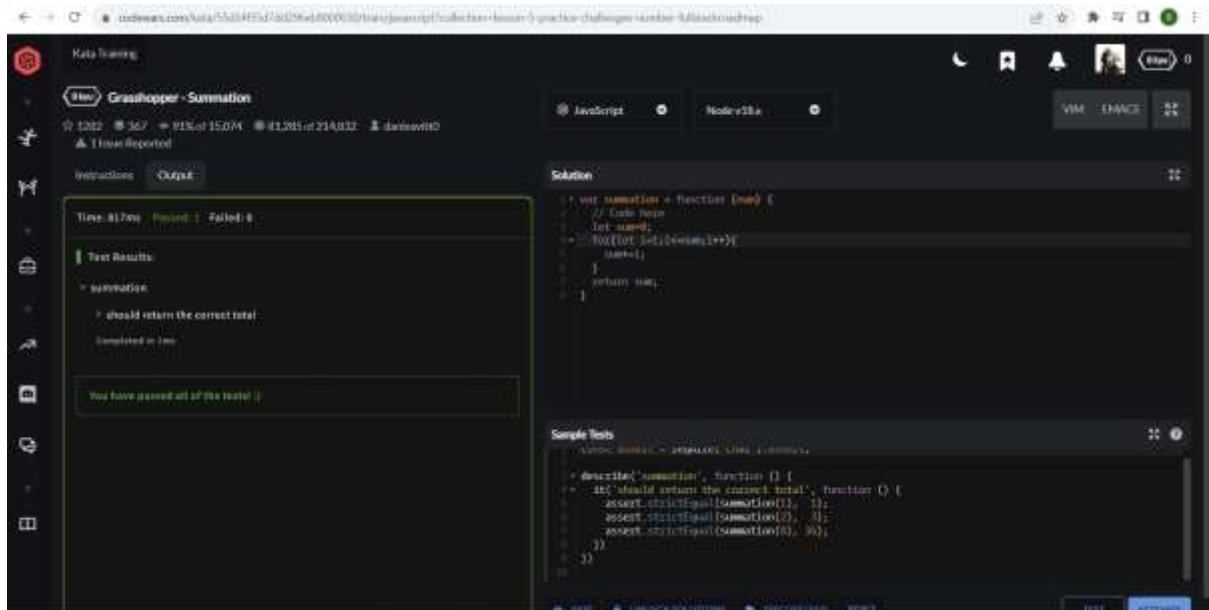
        sum+=i;

    }

    return sum;

}

```



11) Get the mean of an array.

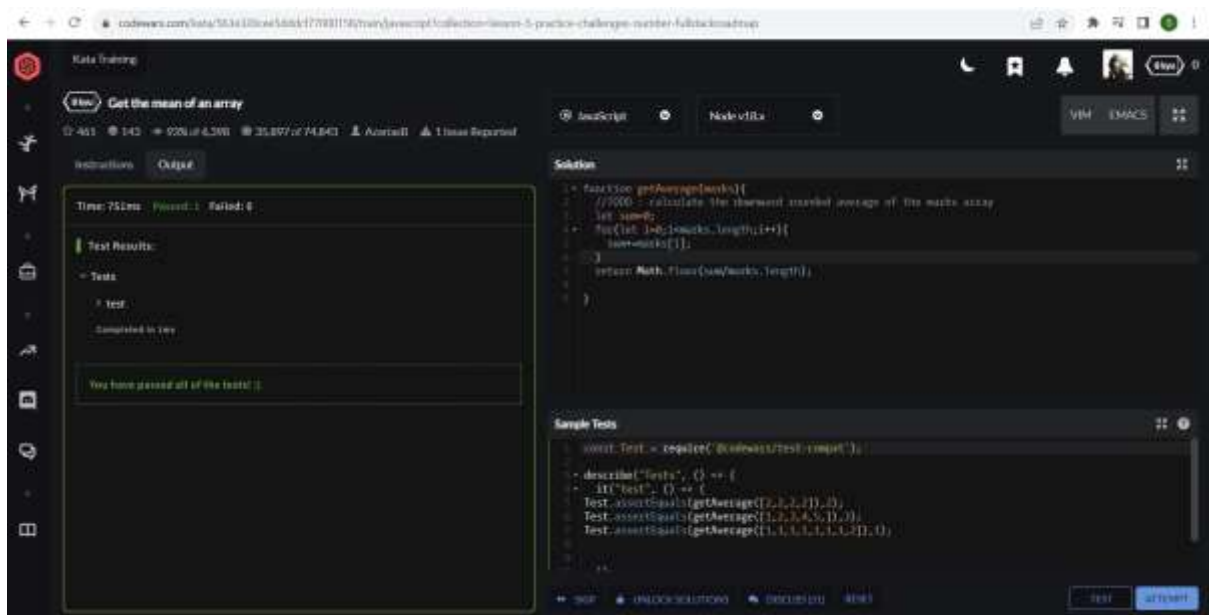
It's the academic year's end, fateful moment of your school report. The averages must be calculated. All the students come to you and entreat you to calculate their average for them. Easy ! You just need to write a script.

```
function getAverage(marks){
  //TODO : calculate the downward rounded average of the marks array

  let sum=0;

  for(let i=0;i<marks.length;i++){
    sum+=marks[i];
  }

  return Math.floor(sum/marks.length);
}
```



12) Remove First and Last Character

It's pretty straightforward. Your goal is to create a function that removes the first and last characters of a string. You're given one parameter, the original string. You don't have to worry with strings with less than two characters.

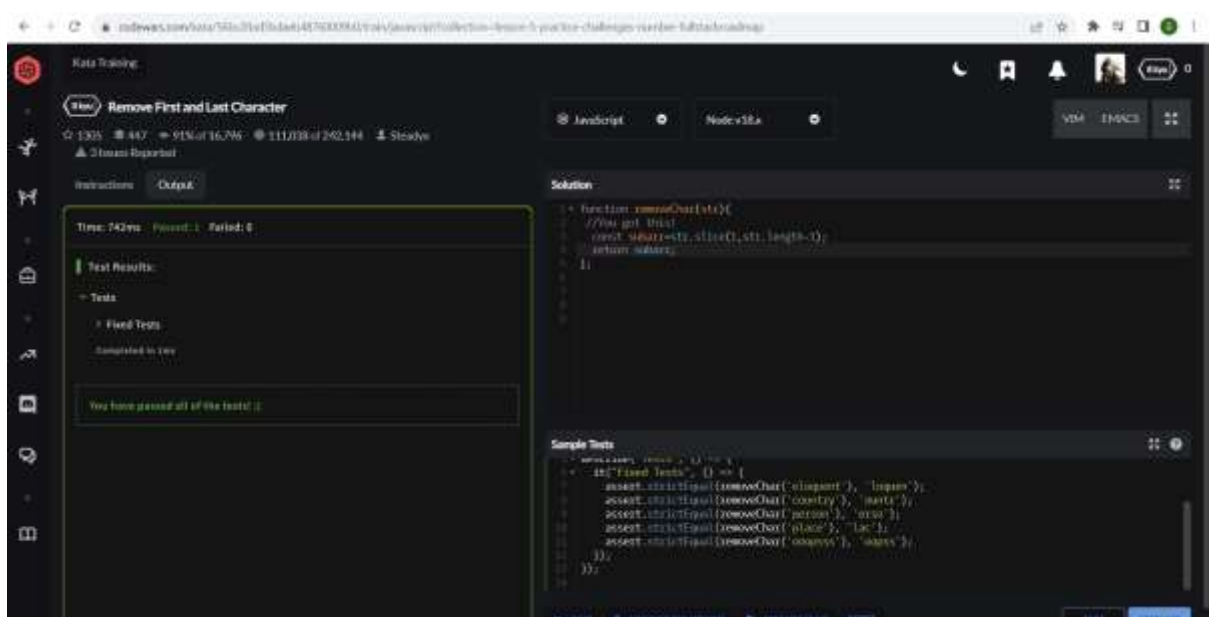
```
function removeChar(str){
```

```
//You got this!
```

```
const subarr=str.slice(1,str.length-1);
```

```
return subarr;
```

```
};
```



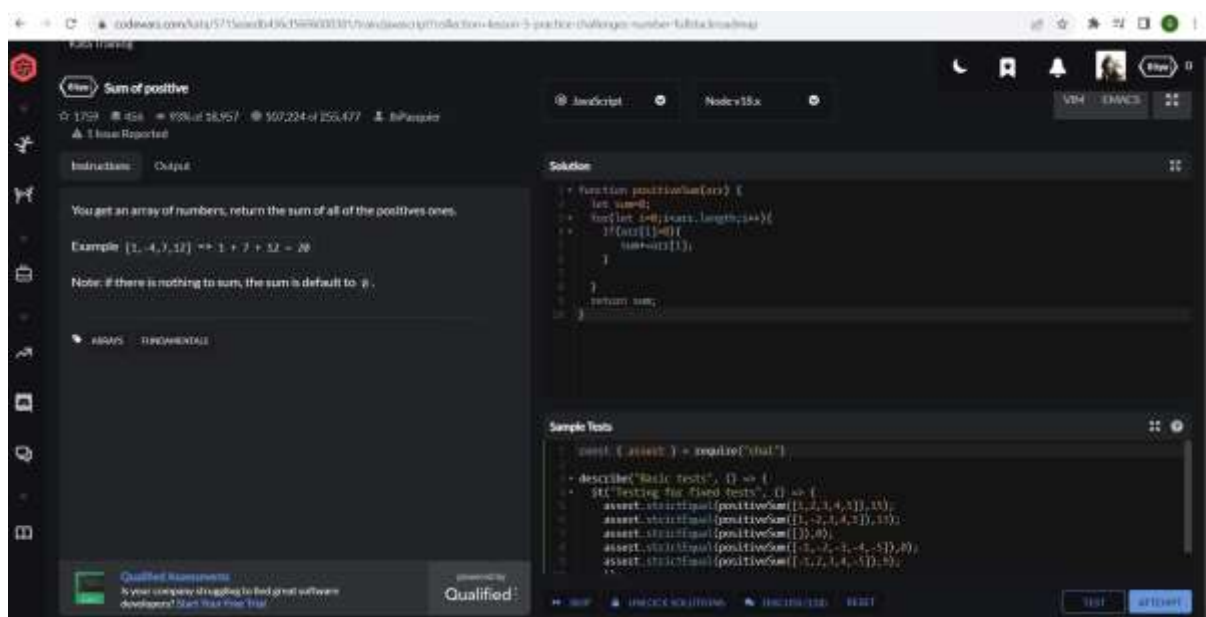
13) Sum of positive.

You get an array of numbers, return the sum of all of the positives ones.

Example [1,-4,7,12] => 1 + 7 + 12 = 20

Note: if there is nothing to sum, the sum is default to 0.

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



14) Basic Mathematical Operations.

Your task is to create a function that does four basic mathematical operations.

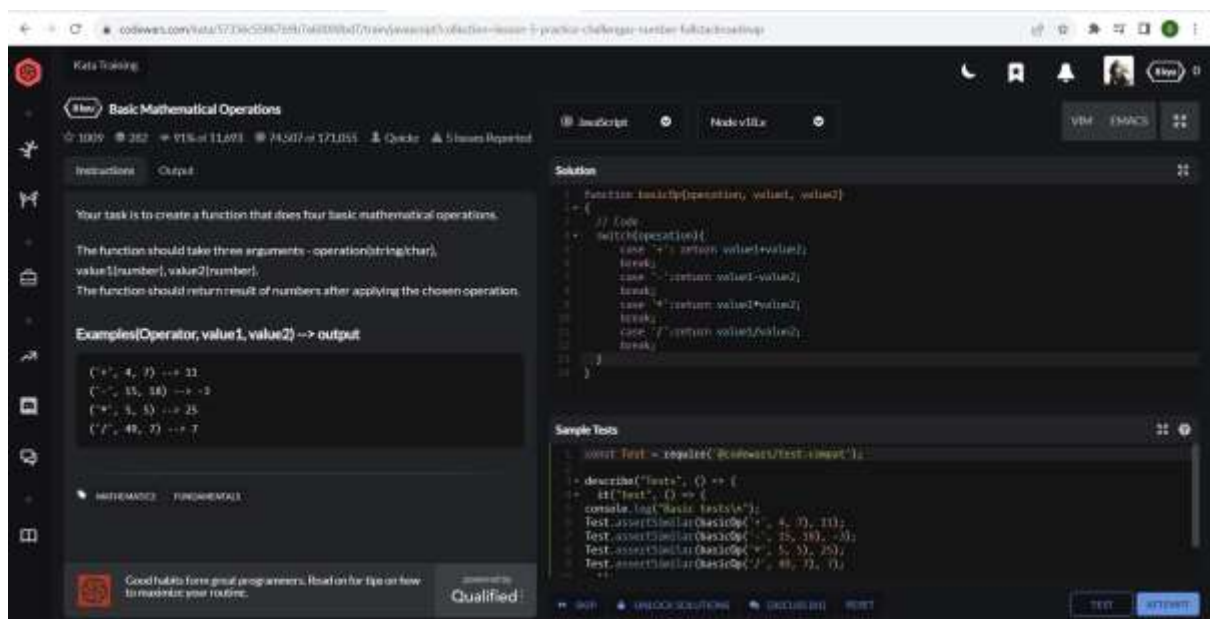
The function should take three arguments - operation(string/char), value1(number), value2(number).The function should return result of numbers after applying the chosen operation.

```
function basicOp(operation, value1, value2)
```

```

{
  // Code
  switch(operation){
    case '+': return value1+value2;
    break;
    case '-':return value1-value2;
    break;
    case '*':return value1*value2;
    break;
    case '/':return value1/value2;
    break;
  }
}

```



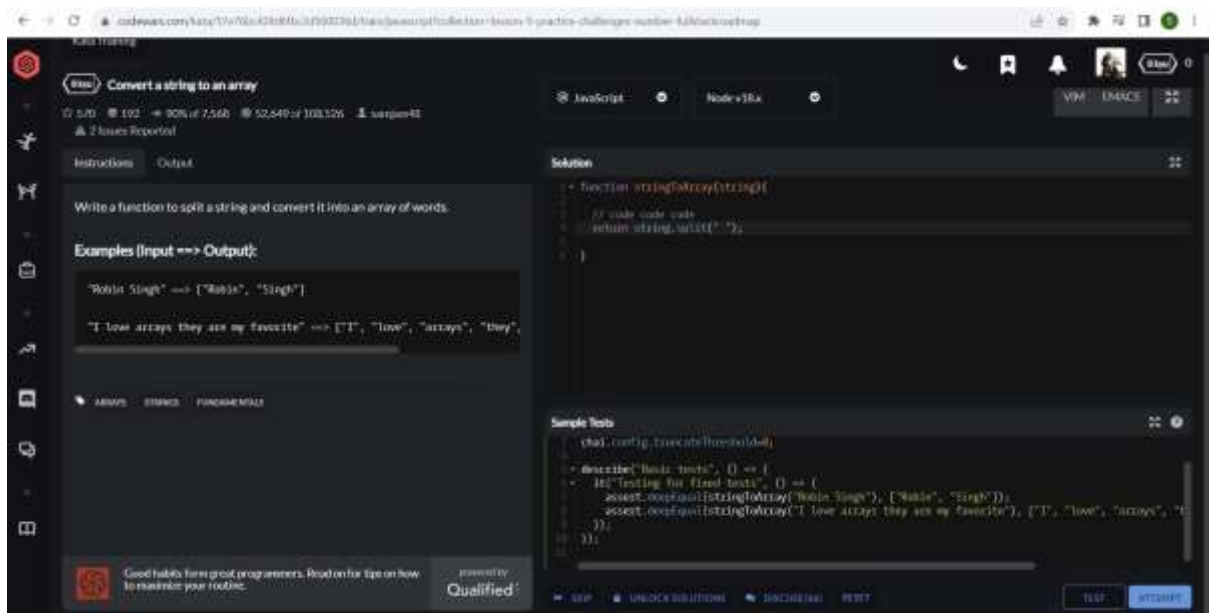
15) Convert a string to an array.

Write a function to split a string and convert it into an array of words.

```

function stringToArray(string){
  return string.split(" ");
}

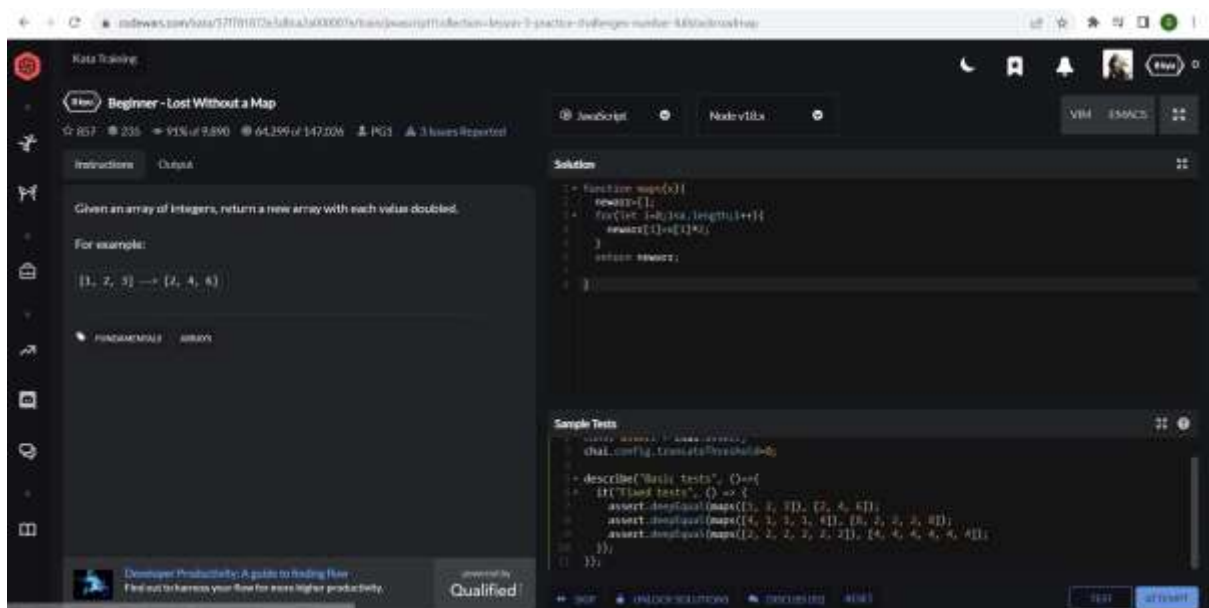
```



16) Beginner - Lost Without a Map.

Given an array of integers, return a new array with each value doubled.

```
function maps(x){
  newarr=[];
  for(let i=0;i<x.length;i++){
    newarr[i]=x[i]*2;
  }
  return newarr;
}
```



17) Is he gonna survive?

A hero is on his way to the castle to complete his mission. However, he's been told that the castle is surrounded with a couple of powerful dragons! each dragon takes 2 bullets to be defeated, our hero has no idea how many bullets he should carry.. Assuming he's gonna grab a specific given number of bullets and move forward to fight another specific given number of dragons, will he survive?Return true if yes, false otherwise :)

```
function hero(bullets, dragons){
```

```
//Get Coding!
```

```
  if(bullets>=2*dragons){
```

```
    return true;
```

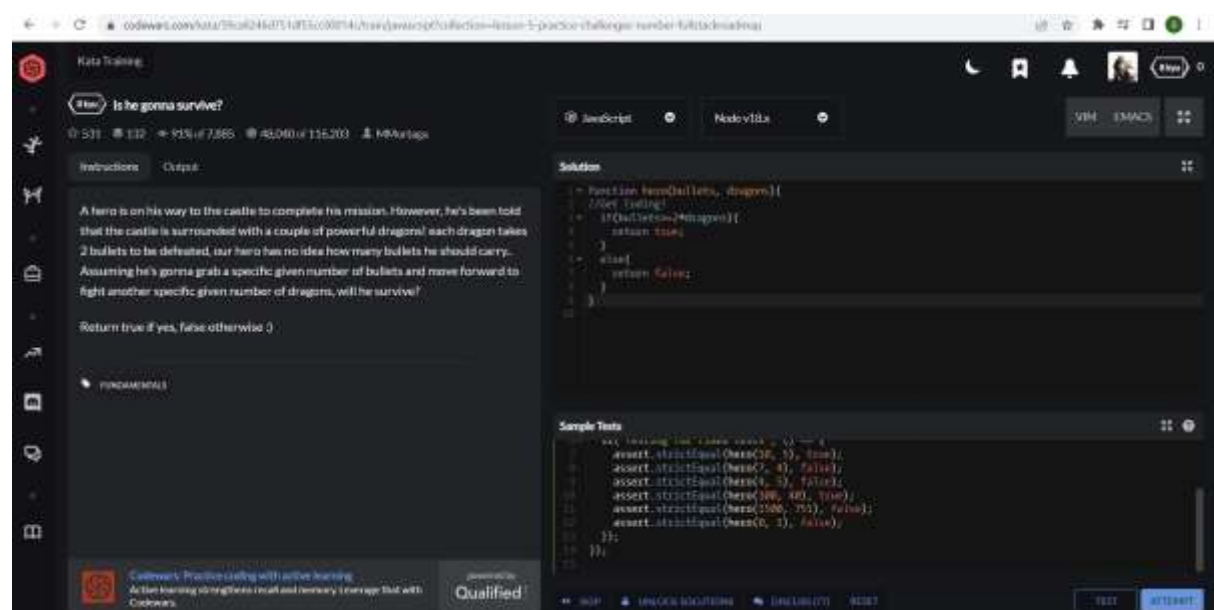
```
  }
```

```
  else{
```

```
    return false;
```

```
  }
```

```
}
```



18) Century From Year.

The first century spans from the year 1 up to and including the year 100, the second century - from the year 101 up to and including the year 200, etc.Task:Given a year, return the century it is in.

```
function century(year) {
```

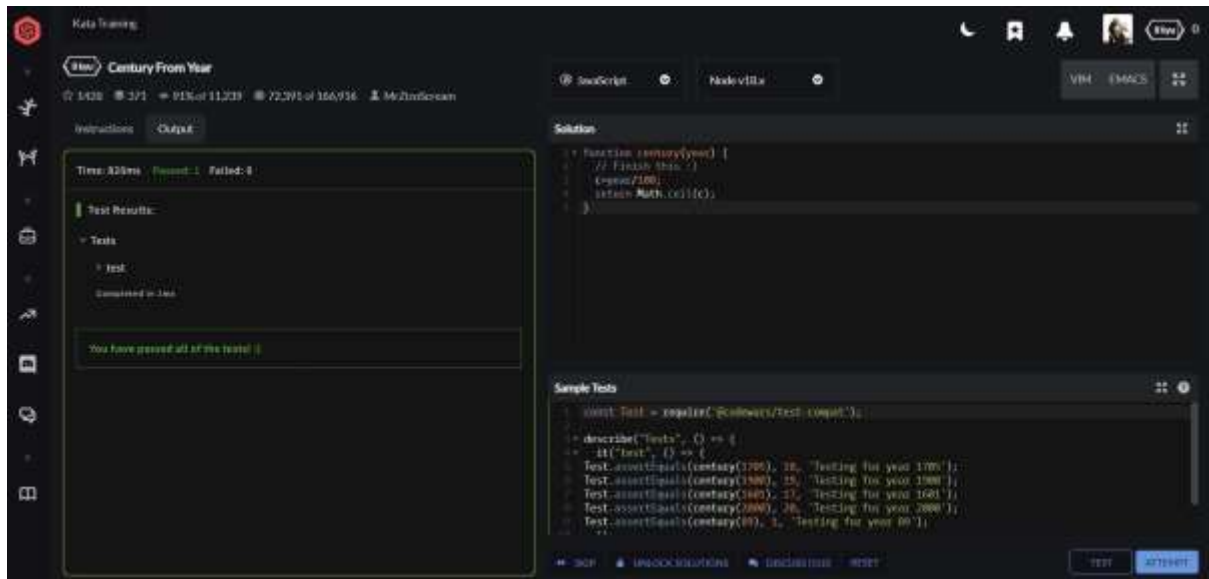
```
  // Finish this :)
```

```
  c=year/100;
```

```

return Math.ceil(c);
}

```



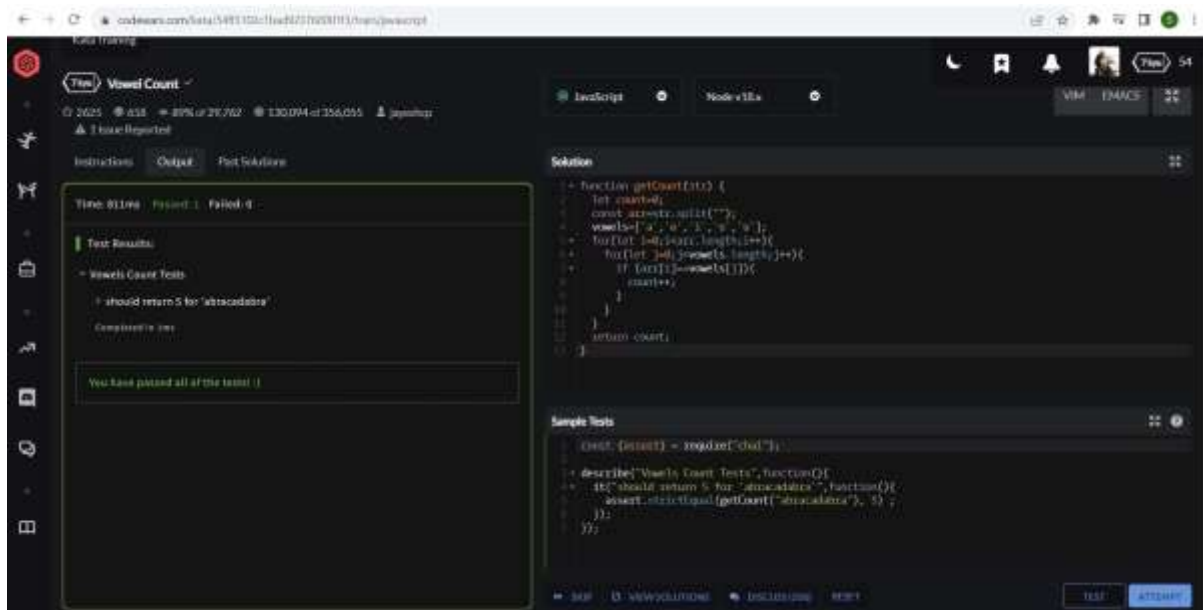
19) Vowel Count.

Return the number (count) of vowels in the given string. We will consider a, e, i, o, u as vowels for this Kata (but not y). The input string will only consist of lower case letters and/or spaces.

```

function getCount(str) {
  let count=0;
  const arr=str.split("");
  vowels=['a','e','i','o','u'];
  for(let i=0;i<arr.length;i++){
    for(let j=0;j<vowels.length;j++){
      if (arr[i]==vowels[j]){
        count++;
      }
    }
  }
  return count;
}

```

20) Cat years, Dog years.

I have a cat and a dog. I got them at the same time as kitten/puppy. That was humanYears years ago. Return their respective ages now as [humanYears, catYears, dogYears]

```
var humanYearsCatYearsDogYears = function(humanYears) {
```

```
  // Your code here!
```

```
  let cat=0;
```

```
  let dog=0;
```

```
  if (humanYears==1){
```

```
    return [humanYears,15,15];
```

```
  }
```

```
  else if(humanYears==2){
```

```
    return [humanYears,24,24];
```

```
  }
```

```
  else{
```

```
    hy=humanYears-2;
```

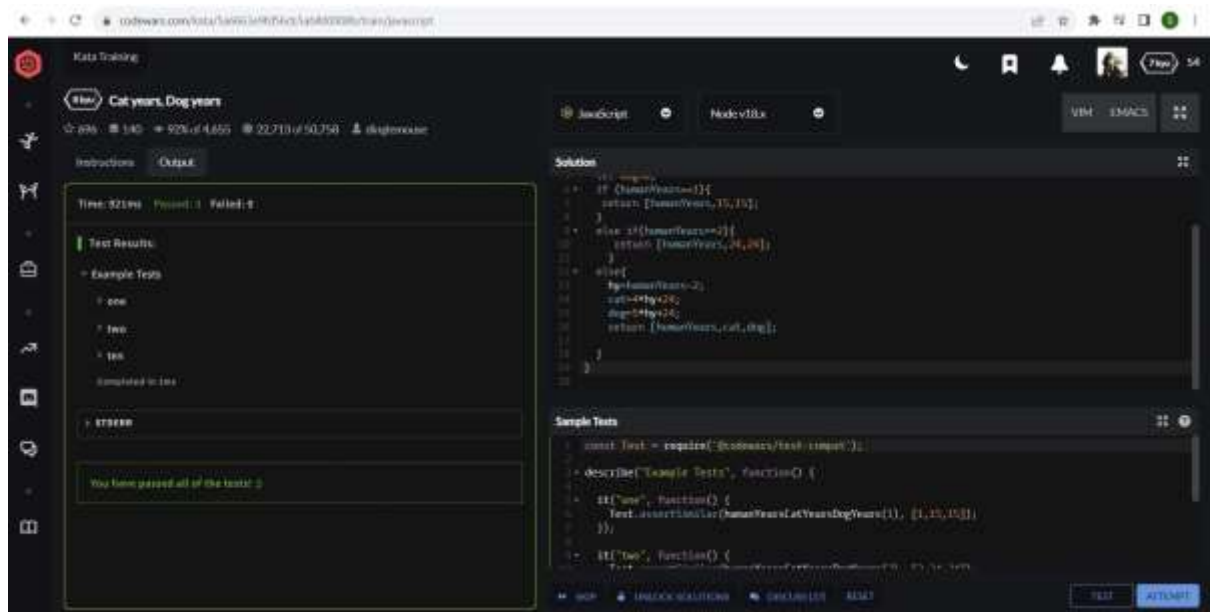
```
    cat=4*hy+24;
```

```
    dog=5*hy+24;
```

```
    return [humanYears,cat,dog]
```

```
  }
```

```
}
```



21) Array plus array.

I'm new to coding and now I want to get the sum of two arrays... Actually the sum of all their elements. I'll appreciate for your help.

P.S. Each array includes only integer numbers. Output is a number too. function
arrayPlusArray(arr1, arr2) {

let sum1=0;

let sum2=0;

for(let i=0;i<arr1.length;i++){

sum1+=arr1[i];

}

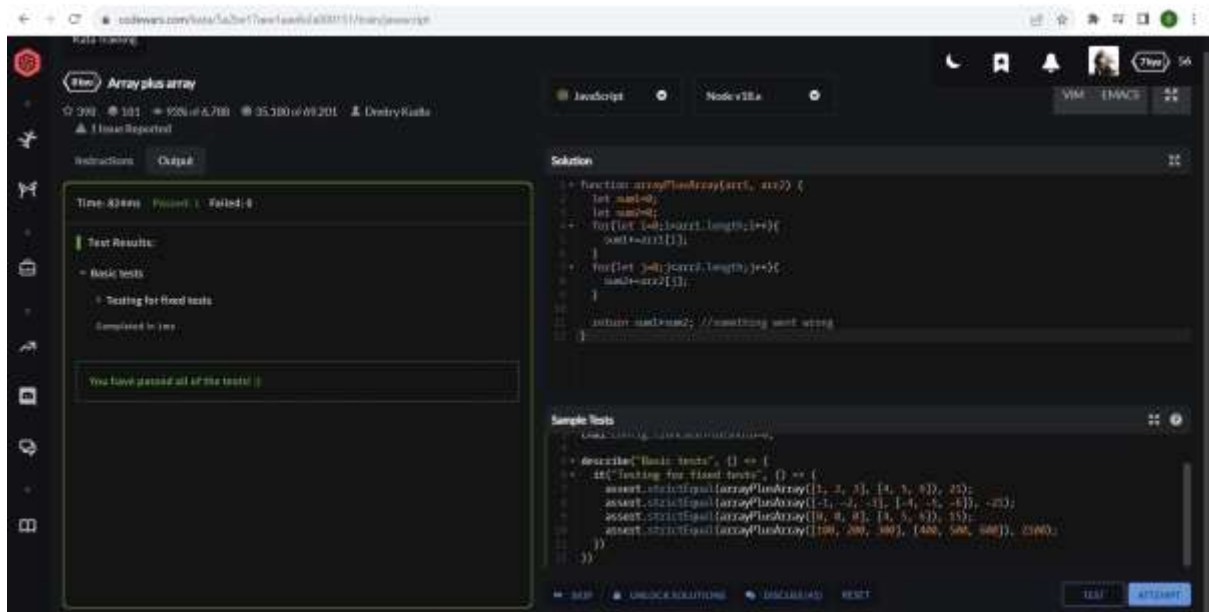
for(let j=0;j<arr2.length;j++){

sum2+=arr2[j];

}

return sum1+sum2; //something went wrong

}



22) String repeat.

Write a function that accepts an integer *n* and a string *s* as parameters, and returns a string of *s* repeated exactly *n* times.

```
function repeatStr (n, s) {

  let str="";

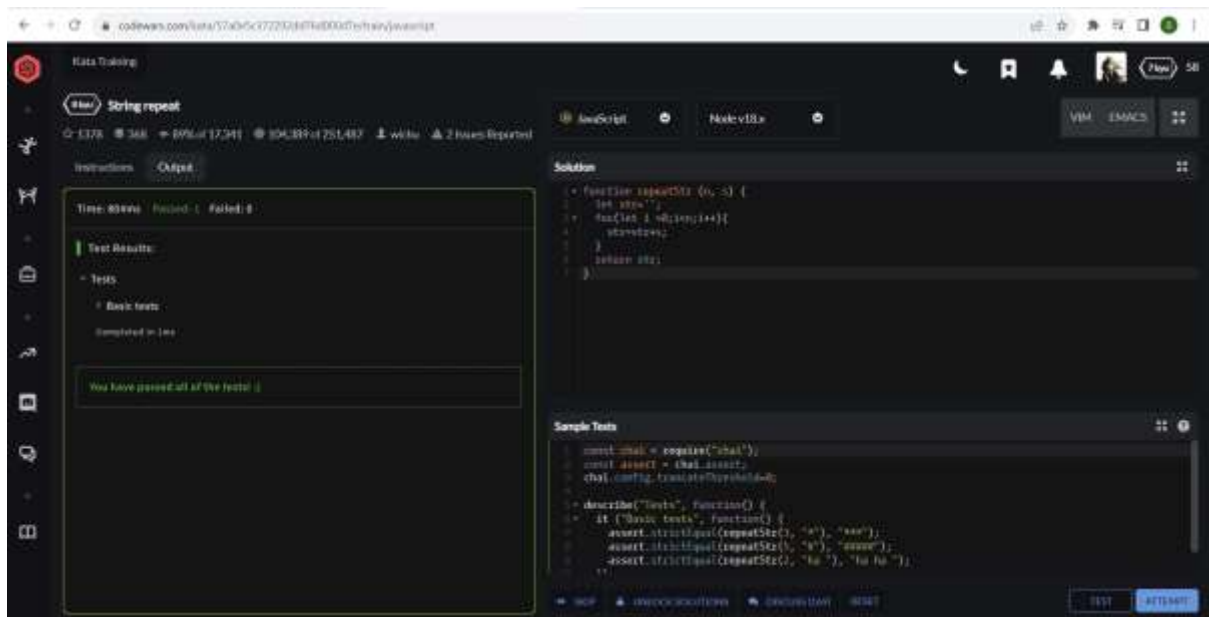
  for(let i =0;i<n;i++){

    str=str+s;

  }

  return str;

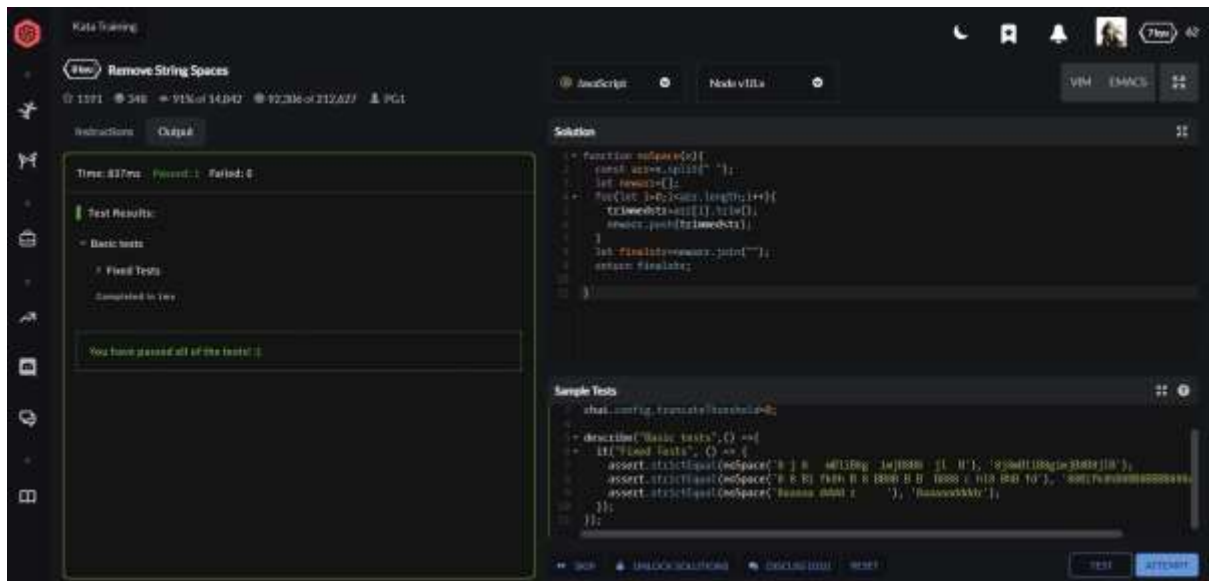
}
```



23) Remove String Spaces.

Write a function that removes the spaces from the string, then return the resultant string.

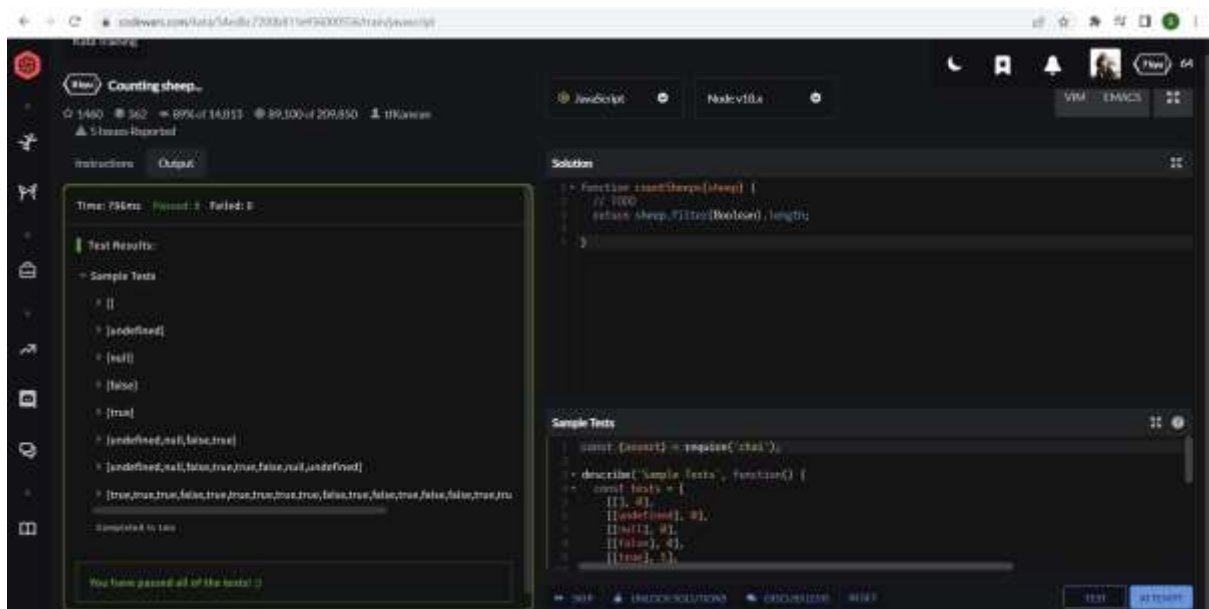
```
function noSpace(x){
  const arr=x.split(" ");
  let newarr=[];
  for(let i=0;i<arr.length;i++){
    trimmedstr=arr[i].trim();
    newarr.push(trimmedstr);
  }
  let finalstr=newarr.join("");
  return finalstr;
}
```



24) Counting sheep...

Consider an array/list of sheep where some sheep may be missing from their place. We need a function that counts the number of sheep present in the array (true means present).

```
function countSheeps(sheep) {  
  // TODO  
  return sheep.filter(Boolean).length;  
}
```



25) Rock Paper Scissors!

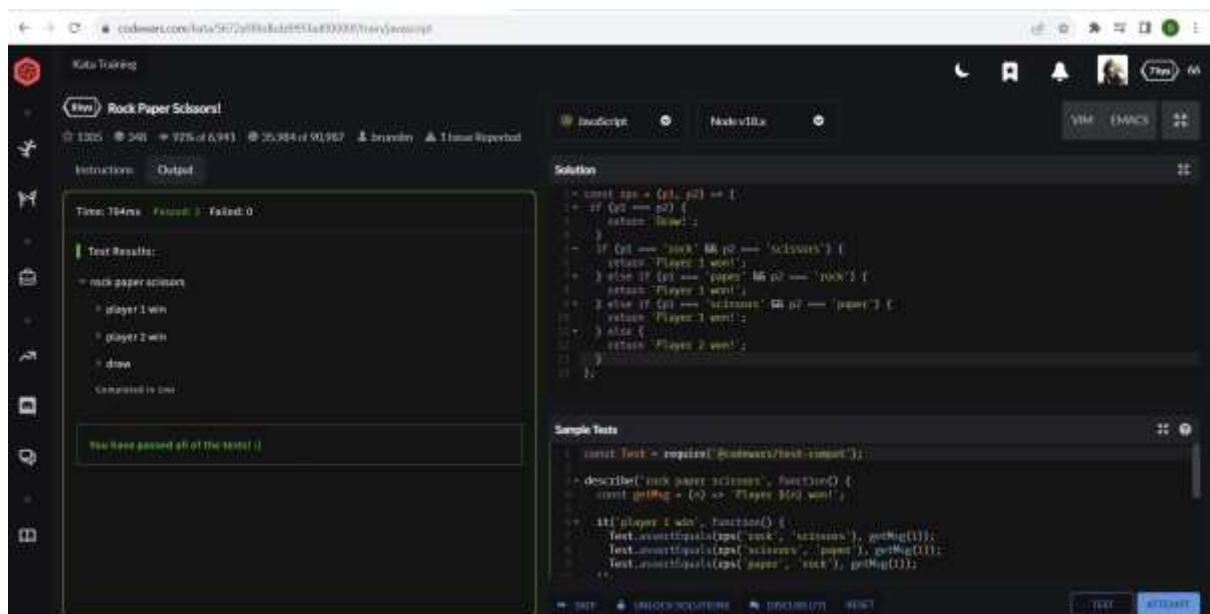
Let's play! You have to return which player won! In case of a draw return Draw!.

```

const rps = (p1, p2) => {
  if (p1 === p2) {
    return 'Draw!';
  }

  if (p1 === 'rock' && p2 === 'scissors') {
    return 'Player 1 won!';
  } else if (p1 === 'paper' && p2 === 'rock') {
    return 'Player 1 won!';
  } else if (p1 === 'scissors' && p2 === 'paper') {
    return 'Player 1 won!';
  } else {
    return 'Player 2 won!';
  }
};

```



26) Total amount of points.

Our football team has finished the championship. Our team's match results are recorded in a collection of strings. Each match is represented by a string in the format "x:y", where x is our team's score and y is our opponents score.

```

function points(games) {

```

```

  let sum=0;

```

```

for (let i=0; i<games.length; ++i)
{
    if (games[i][0]>games[i][2])
        sum+=3;

    if (games[i][0]==games[i][2])
        sum+=1;
}

return sum;
}

```

The screenshot shows a CodeWars challenge titled "Total amount of points". The interface includes a left sidebar with navigation icons, a central editor area, and a right sidebar with test results and sample tests.

Challenge Details:

- Title:** Total amount of points
- Progress:** 1225 / 317 (92% of 6.6M)
- Language:** JavaScript
- Mode:** Node v18.12.0

Test Results:

- Time:** 0.000s
- Passed:** 1
- Failed:** 0
- Sample Tests:** Completed in 1ms
- Status:** You have passed all of the tests!

Solution:

```

function points(games) {
    let sum=0;
    for (let i=0; i<games.length; ++i)
    {
        if (games[i][0]>games[i][2])
            sum+=3;
        if (games[i][0]==games[i][2])
            sum+=1;
    }
    return sum;
}

```

Sample Tests:

```

describe("Total amount of points", function() {
    it("Sample Tests", function() {
        assert.strictEqual(points(["1-0","2-0","3-0","4-0","2-1","3-1","4-1","3-2","4-2","4-3"]), 30);
        assert.strictEqual(points(["1-1","2-2","3-3","4-4","2-2","3-3","4-4","3-3","4-4","4-4"]), 10);
        assert.strictEqual(points(["0-1","0-2","0-3","0-4","1-2","1-3","1-4","2-3","2-4","3-4"]), 0);
        assert.strictEqual(points(["1-0","2-0","3-0","4-0","2-1","3-1","1-4","2-3","2-4","3-4"]), 15);
        assert.strictEqual(points(["1-0","2-0","3-0","4-0","2-2","3-3","1-4","2-3","2-4","3-4"]), 12);
    });
});

```