1. Do the below programs in anonymous function & IIFE
   1. Print odd numbers in an array
   2. Convert all the strings to title caps in a string array
   3. Sum of all numbers in an array
   4. Return all the prime numbers in an array
   5. Return all the palindromes in an array
   6. Return median of two sorted arrays of same size
   7. Remove duplicates from an array
   8. Rotate an array by k times

a.code

//finding odd numbers present in an array using anonymous function

let arrodd=function(arrr){

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

if(arrr[i]%2===1){

console.log(arrr[i]);

}

}

}

let array=[1,2,3,4,5,6,7,8,9,10,11];

let n=array.length;

//arrodd(n)

console.log(arrodd(array));

---------------------------

(function(arrr){

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

if(arrr[i]%2===1){

console.log(arrr[i]);

}

}

}) ([1,2,3,4,6,7,8,77,88,99]);

-------------------------------------------------------

b.code:

//Convert all the strings to title caps in a string array using anonymous function

let lilcaps = function(str){

str = str.toLowerCase().split(' ');

for (var i = 0; i < str.length; i++) {

str[i] = str[i].charAt(0).toUpperCase() + str[i].slice(1);

}

return str.join(' ');

}

console.log(lilcaps(("main boot camp")));

console.log(lilcaps(("guvi geeeks")));

//Convert all the strings to title caps in a string array using IIFE Function

(function(str){

str = str.toLowerCase().split(' ');

for (var i = 0; i < str.length; i++) {

str[i] = str[i].charAt(0).toUpperCase() + str[i].slice(1);

}

console.log(str.join(' '));

})("guvi geeks chennai");

--------------------------------------------------------------------------

c.code:

//Sum of all numbers in an array usinh anonymous fun

let sumarr=function(arrr){

let sum=0;

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

console.log(arrr.reduce((a, b) => a + b, 0));

//}

}

let array=[1,2,3,4,5,6,7,8,9,10];

console.log(sumarr(array));

//Sum of all numbers in an array usinh iife fun

(function(arrr){

console.log(arrr.reduce((a, b) => a + b, 0));

})([1,2,3,4,5,6,7,8,9,10]);

-----------------------------------------------------

d.code:

//Return all the prime numbers in an array

let primes = (arr) => {

    return arr.filter((e) => {

      for (let i = 2; i < e; i++) {

        if (e % i === 0) {

          return false;

        }

      }

      return e > 1;

    });

  };

  console.log(primes([1, 2, 5, 16, 25, 99, 101]));

  //prime using iife function

  (function (arr){

    return arr.filter((e) => {

        for (let i = 2; i < e; i++) {

          if (e % i === 0) {

            return false;

          }

        }

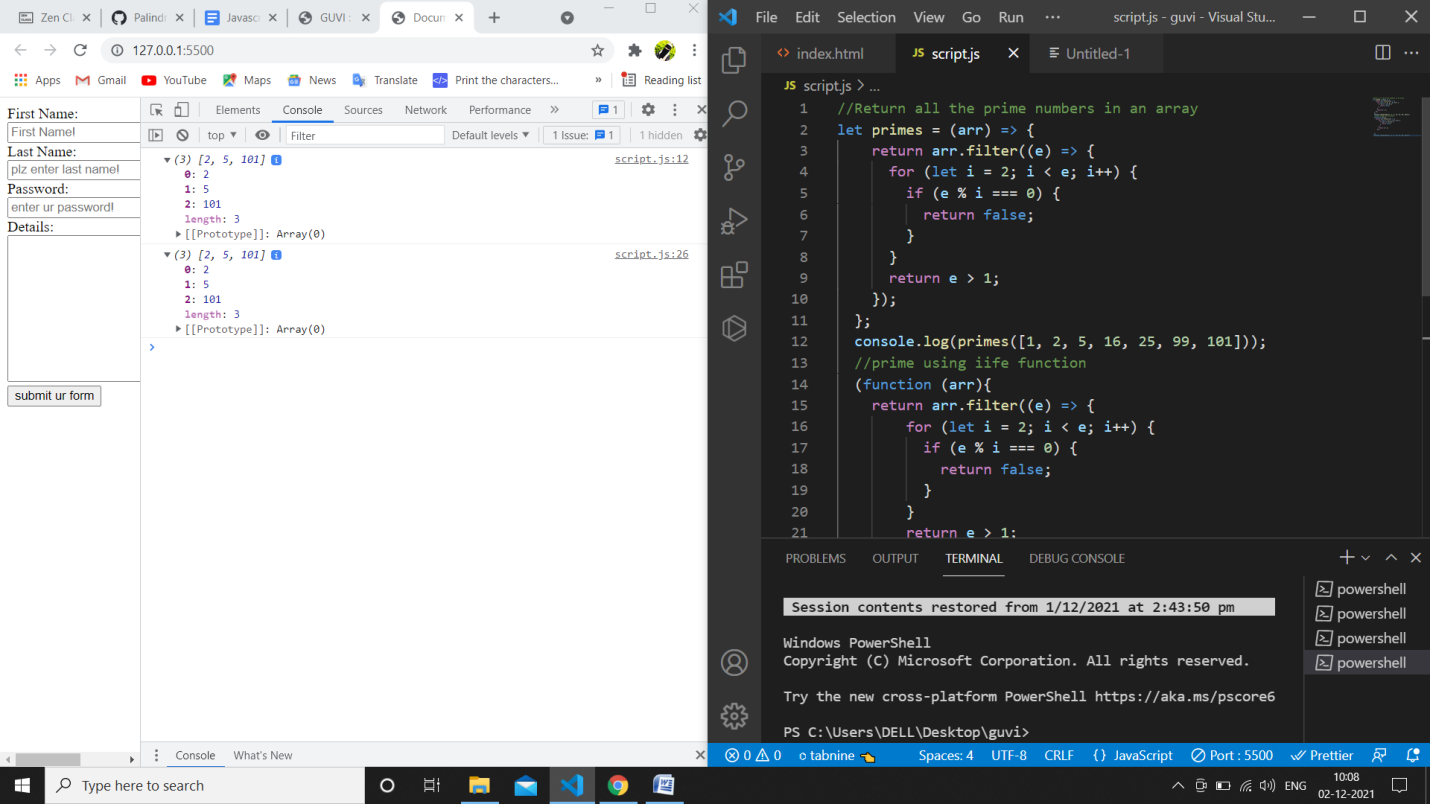
        return e > 1;

      });

  })

  console.log(primes([1, 2, 5, 16, 25, 99, 101]));

o/p:



e.code

let palindromes = function (arr) {

    let arr1 = [];

    for (let i in arr) {

      // console.log(arr[i].split("").reverse());

      // console.log(arr[i].split(""));

      if (arr[i].split("").reverse().join("") === arr[i].split("").join("")) {

        arr1.push(arr[i]);

      }

    }

    return arr1;

  };

  console.log(palindromes(["eye", "aaa", "arpit", "arcade", "aya"]));

  //iife function:

  (function(arr){

    let arr1 = [];

    for (let i in arr) {

      // console.log(arr[i].split("").reverse());

      // console.log(arr[i].split(""));

      if (arr[i].split("").reverse().join("") === arr[i].split("").join("")) {

        arr1.push(arr[i]);

      }

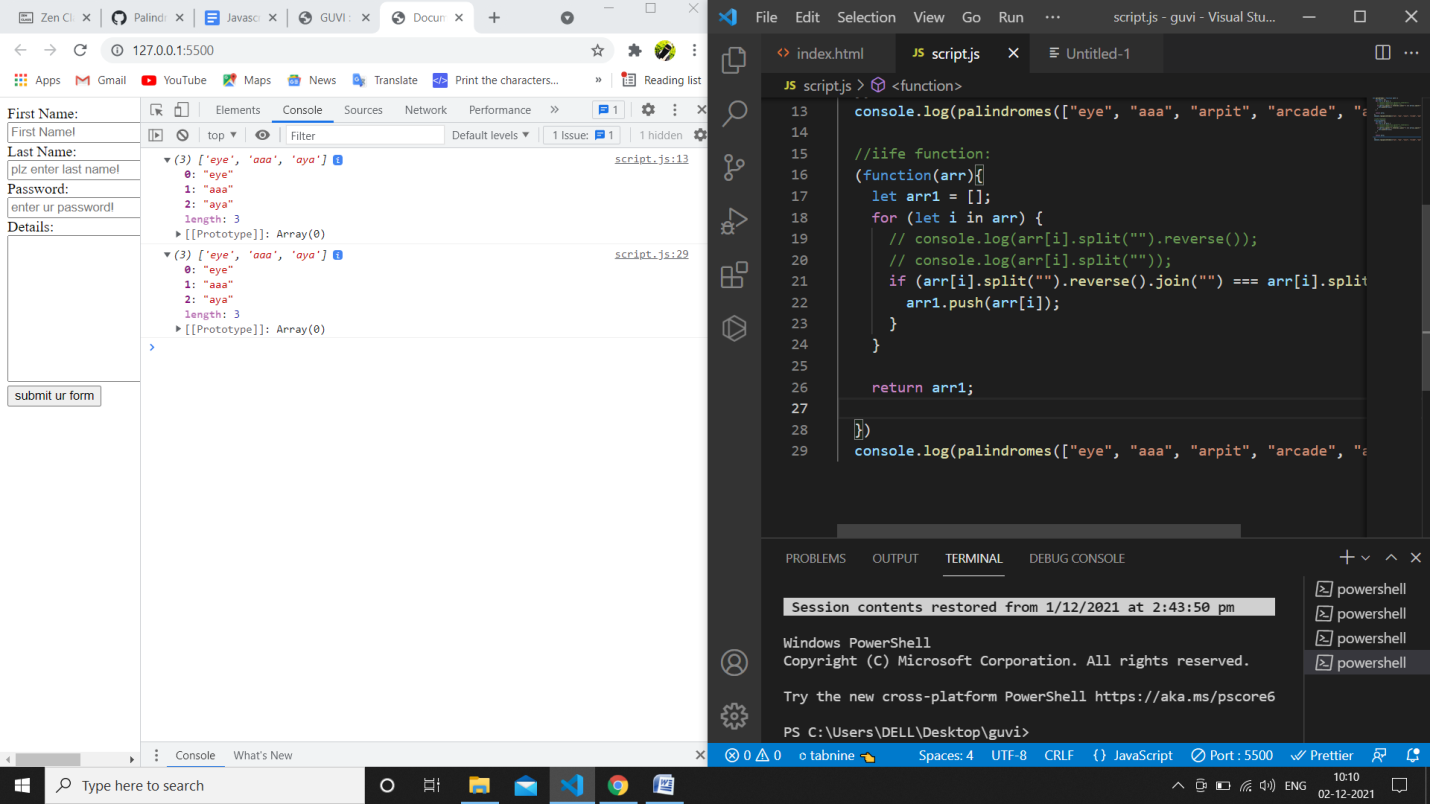
    }

    return arr1;

  })

  console.log(palindromes(["eye", "aaa", "arpit", "arcade", "aya"]));

o/p:



f.code:

let median=function(ar1, ar2, n){

{

    var i = 0; /\* Current index of i/p array ar1[] \*/

    var j = 0; /\* Current index of i/p array ar2[] \*/

    var count;

    var m1 = -1, m2 = -1;

    /\* Since there are 2n elements, median will be average

    of elements at index n-1 and n in the array obtained after

    merging ar1 and ar2 \*/

    for (count = 0; count <= n; count++)

    {

        /\*Below is to handle case where all elements of ar1[] are

        smaller than smallest(or first) element of ar2[]\*/

        if (i == n)

        {

            m1 = m2;

            m2 = ar2[0];

            break;

        }

        /\*Below is to handle case where all elements of ar2[] are

        smaller than smallest(or first) element of ar1[]\*/

        else if (j == n)

        {

            m1 = m2;

            m2 = ar1[0];

            break;

        }

        /\* equals sign because if two

            arrays have some common elements \*/

        if (ar1[i] <= ar2[j])

        {

            m1 = m2; /\* Store the prev median \*/

            m2 = ar1[i];

            i++;

        }

        else

        {

            m1 = m2; /\* Store the prev median \*/

            m2 = ar2[j];

            j++;

        }

    }

    return (m1 + m2)/2;

}}

median()

/\* Driver program to test above function \*/

var ar1 = [1, 12, 15, 26, 38];

var ar2 = [2, 13, 17, 30, 45];

var n1 = ar1.length;

var n2 = ar2.length;

if (n1 == n2)

    console.log("Median is "+ median(ar1, ar2, n1));

else

    console.log("Doesn't work for arrays of unequal size");

o/p: 16

g.Remove duplicates from an array

// program to remove duplicate value from an array using anonymous function

let getUnique=function(arr){

    let uniqueArr = [];

    // loop through array

    for(let i of arr) {

        if(uniqueArr.indexOf(i) === -1) {

            uniqueArr.push(i);

        }

    }

    console.log(uniqueArr);

}

const array = [1, 2, 3, 2, 3];

// calling the function

// passing array argument

console.log(getUnique(array));

//using iife function:

(function (arr) {

    let uniqueArr = [];

    // loop through array

    for(let i of arr) {

        if(uniqueArr.indexOf(i) === -1) {

            uniqueArr.push(i);

        }

    }

    console.log(uniqueArr);

})

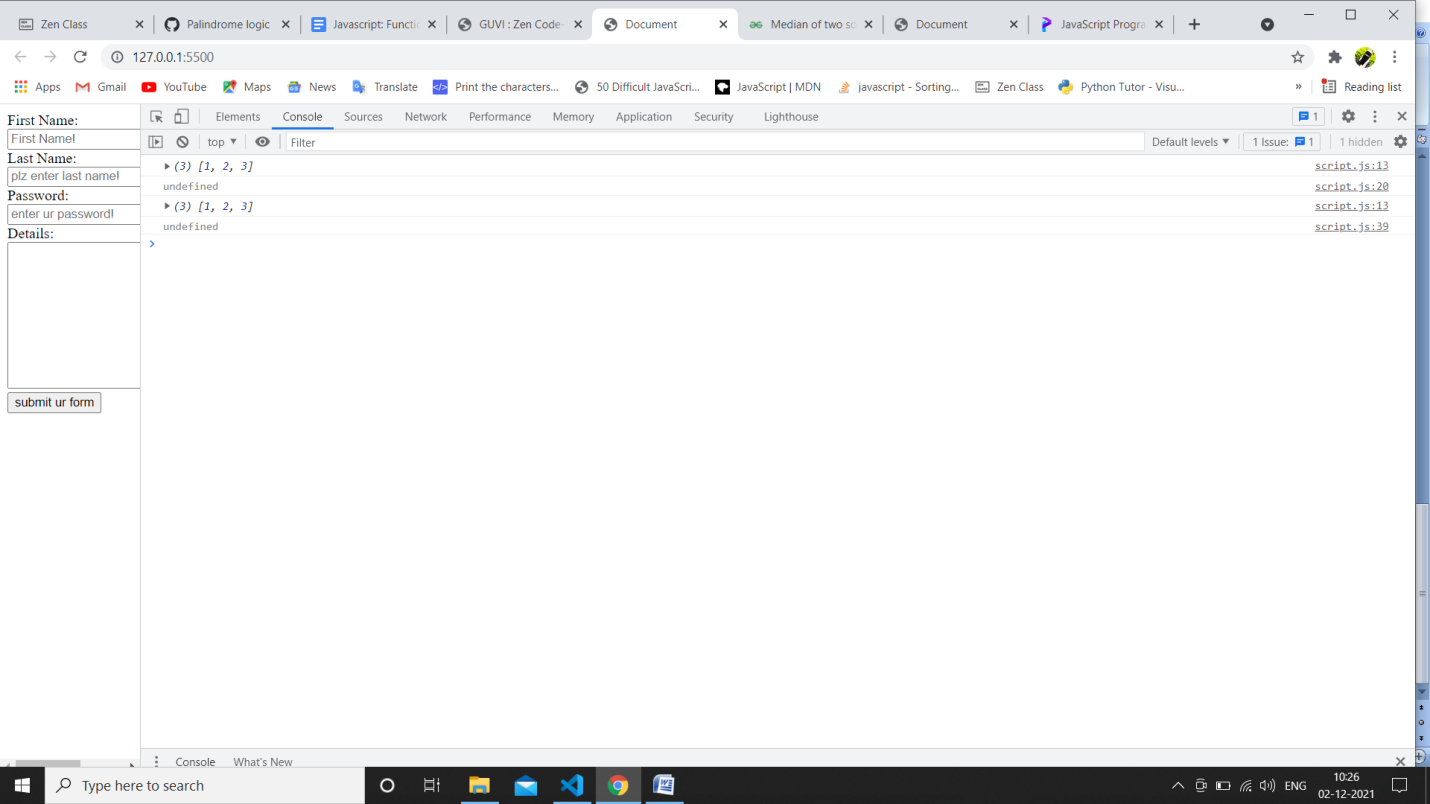
const array1 = [1, 2, 3, 2, 3];

// calling the function

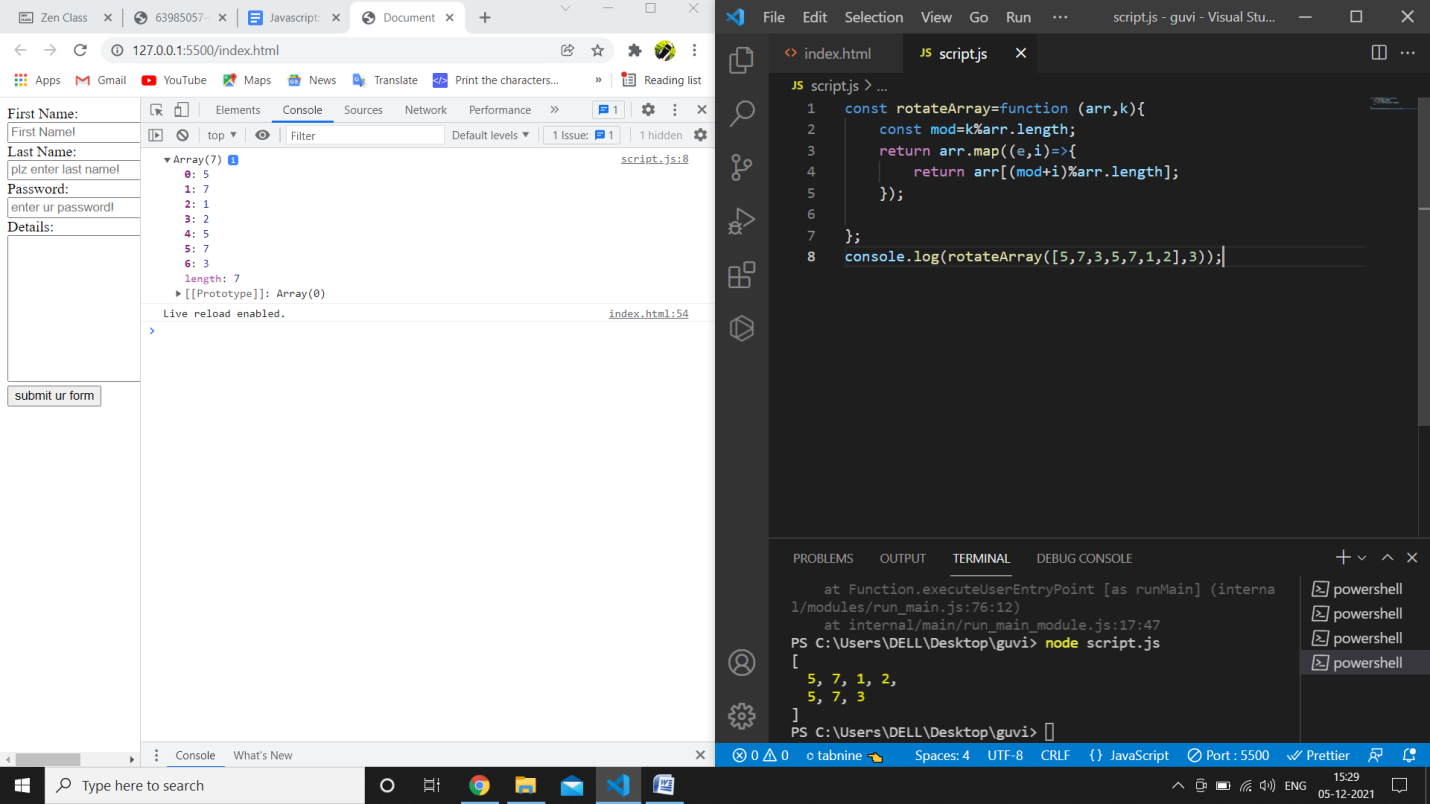
// passing array argument

console.log(getUnique(array1));

o/p:



h.Rotate an array by k times



---------------------------------------------------------------------------------------------------------------------

**Problem**:

Write a function called “addFive”.  
Given a number, “addFive” returns 5 added to that number.

var num = 5;

function addFive(num1) {

    return num+num1;

}

let num1;

var result = addFive(5)

console.log(result);

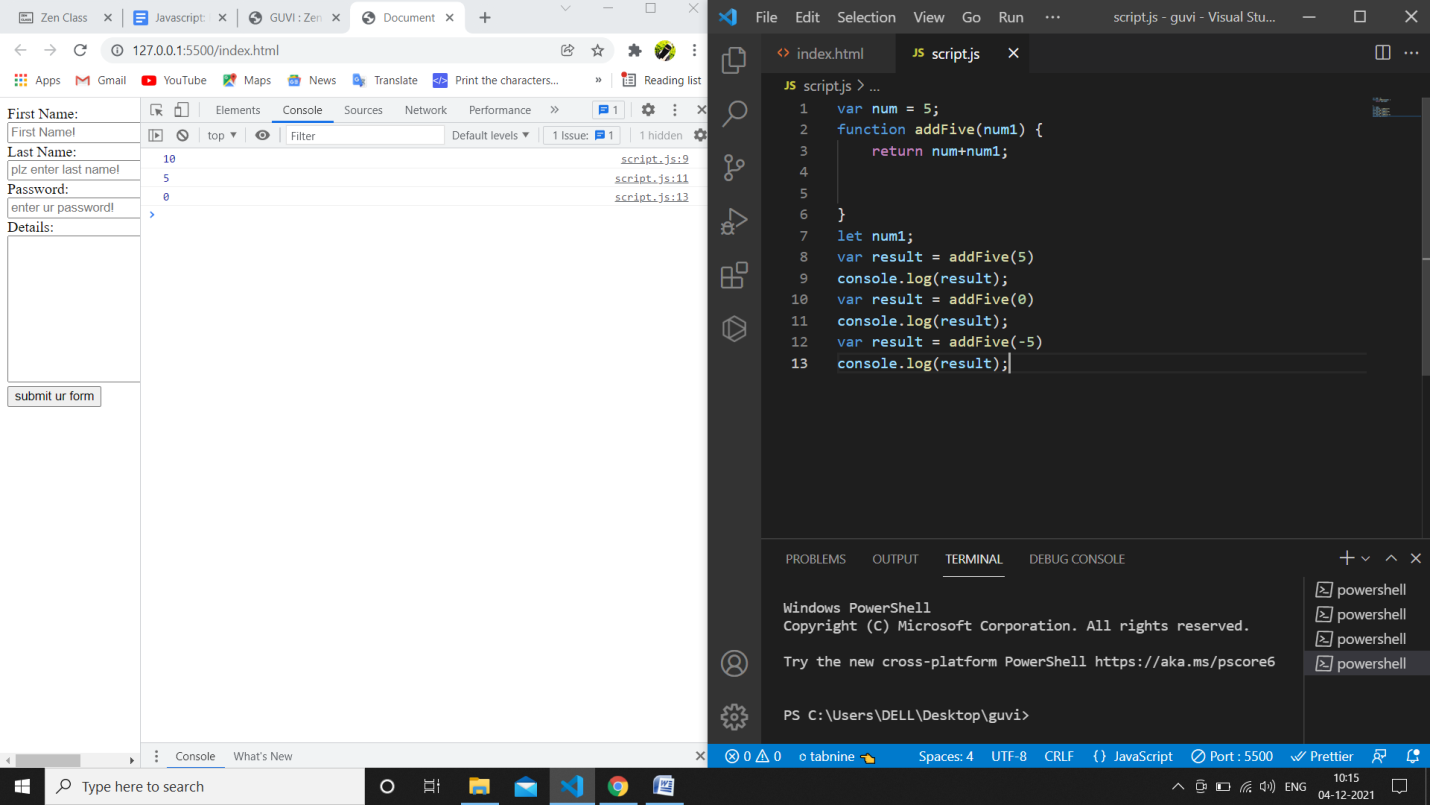
var result = addFive(0)

console.log(result);

var result = addFive(-5)

console.log(result);

o/p:



Write a function called “getOpposite”.  
Given a number, return its opposite

var num = 5.5;

//  console.log((parseFloat(num)));

function getOpposite(num) {

    if((num>0 || num<0) && num%1===0){

        return (num)\*(-1);

    }

    else if(num===0){

        return 0;

    }

    else  if(num===num || num%1!==0){

        return -1;

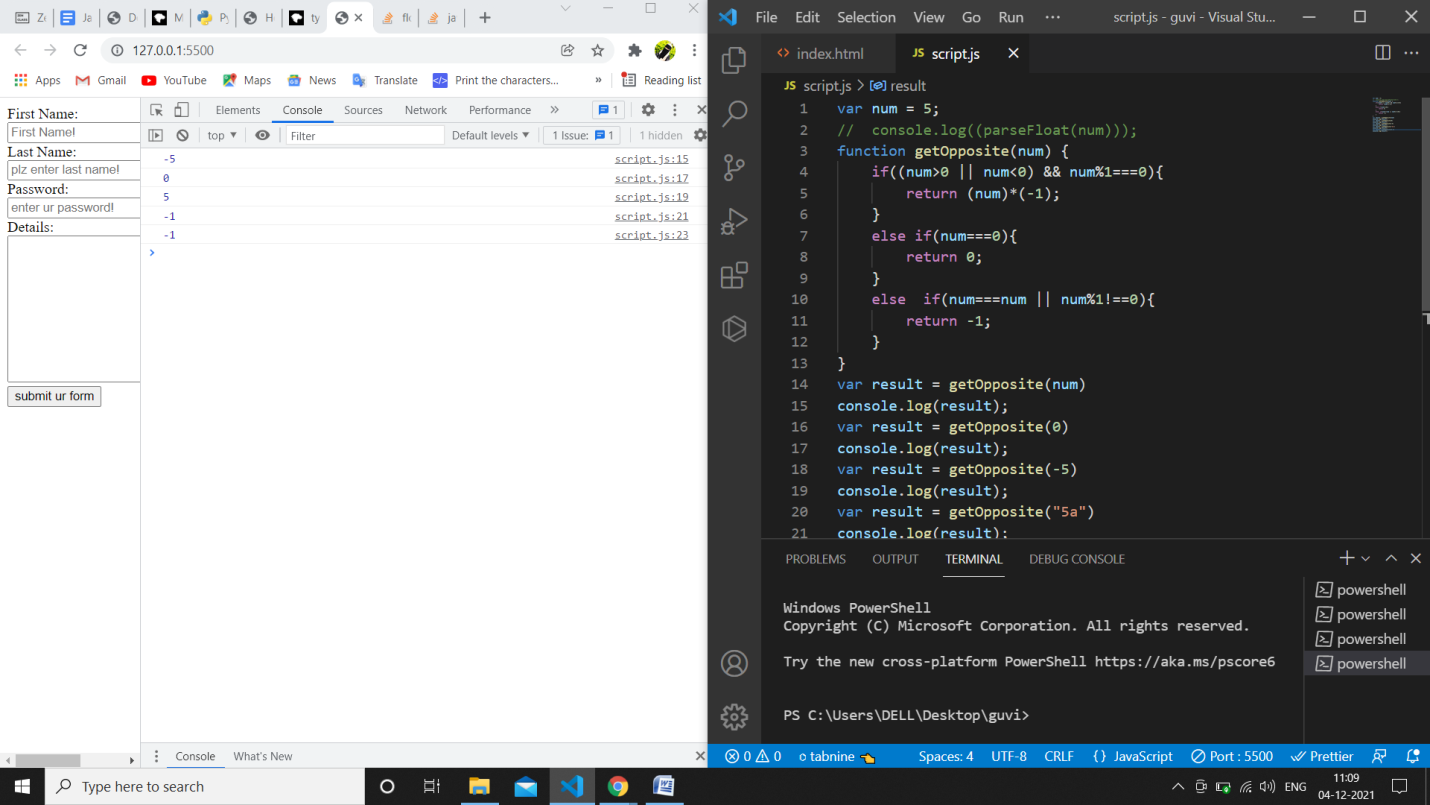
    }

}

var result = getOpposite(num)

console.log(result);

o/p:



Fill in your code that takes an number minutes and converts it to seconds.

var min = 3;

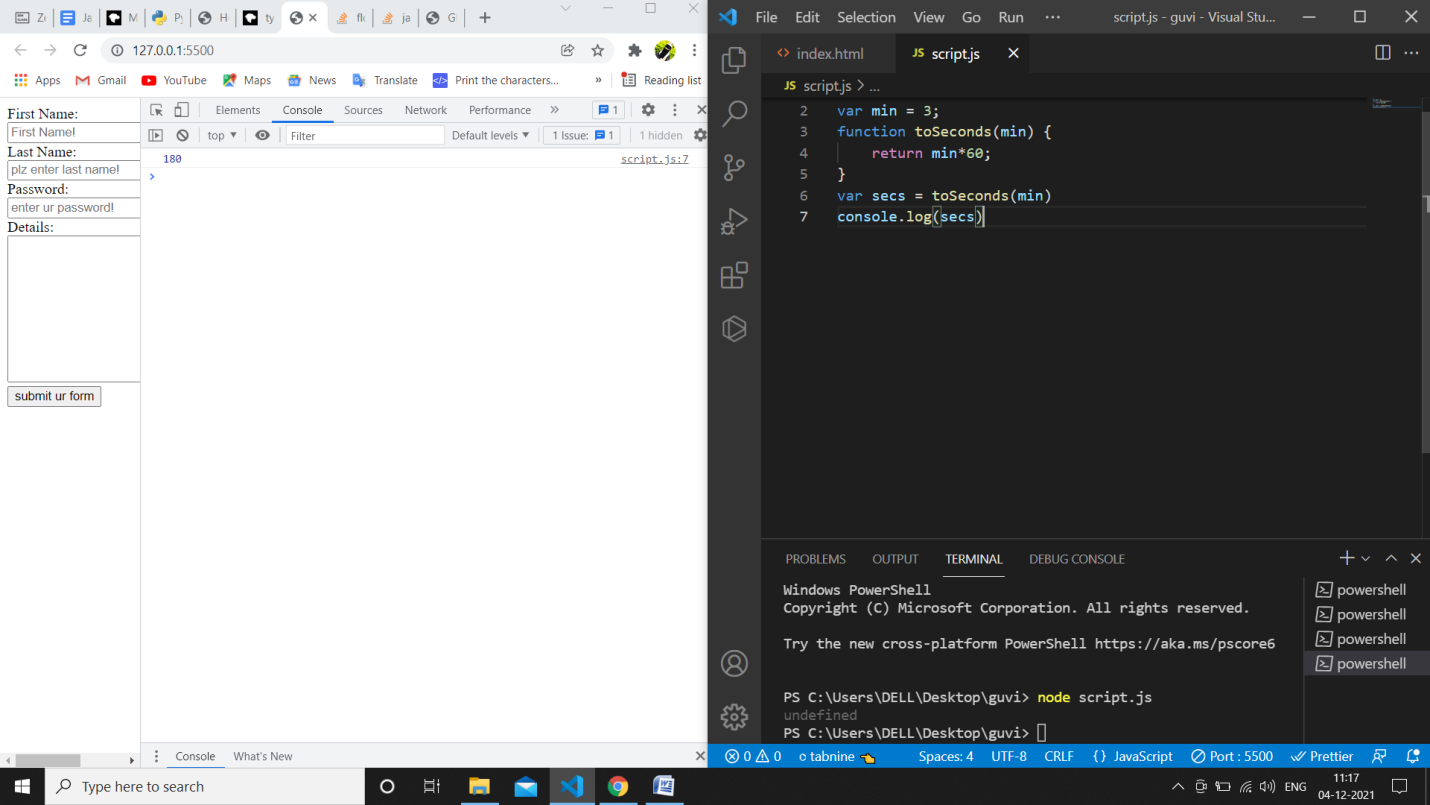
function toSeconds(min) {

    return min\*60;

}

var secs = toSeconds(min)

console.log(secs)



Create a function that takes a string and returns it as an integer.

var mystr = "5";

function toInteger(mystr) {

    if (mystr==mystr){

        return mystr;

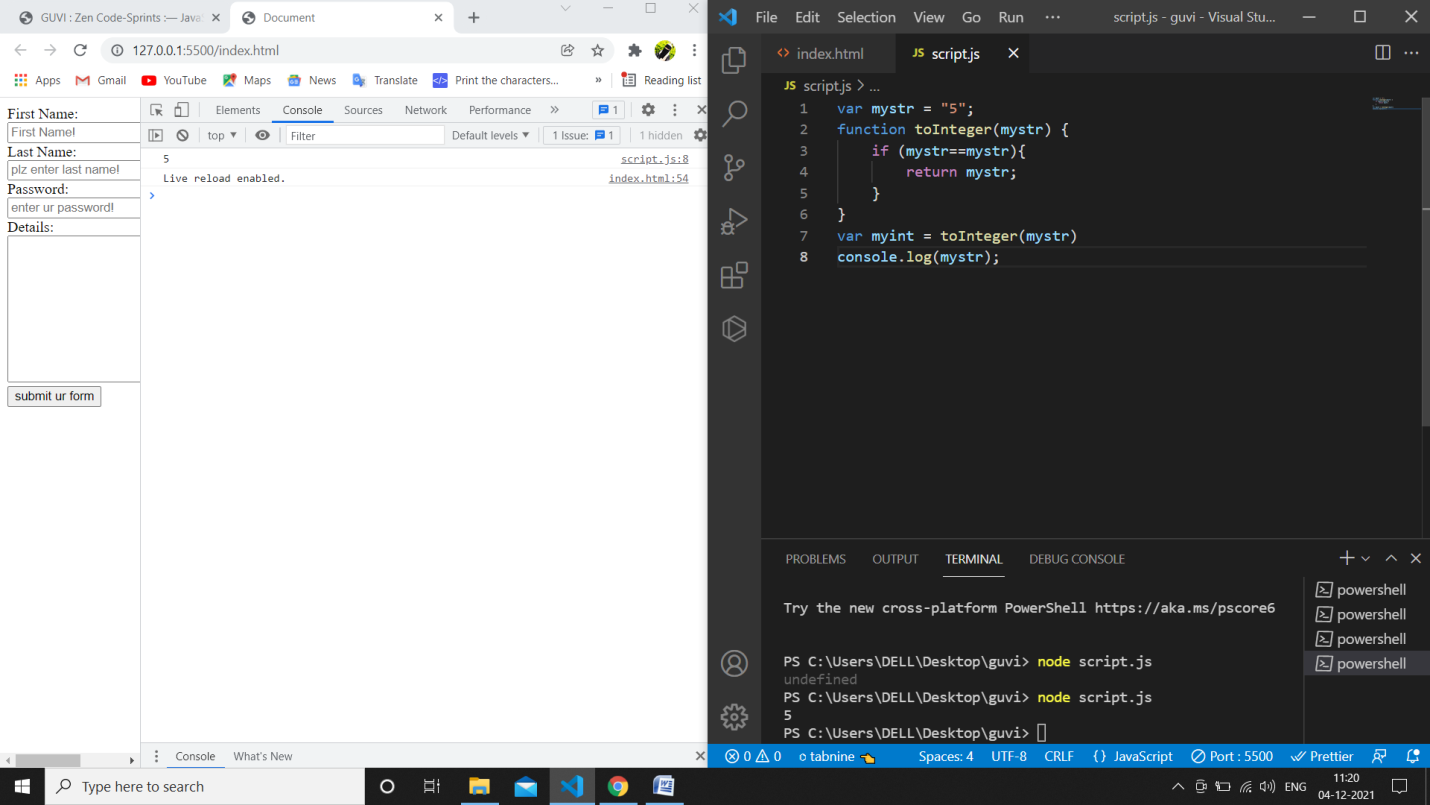
    }

}

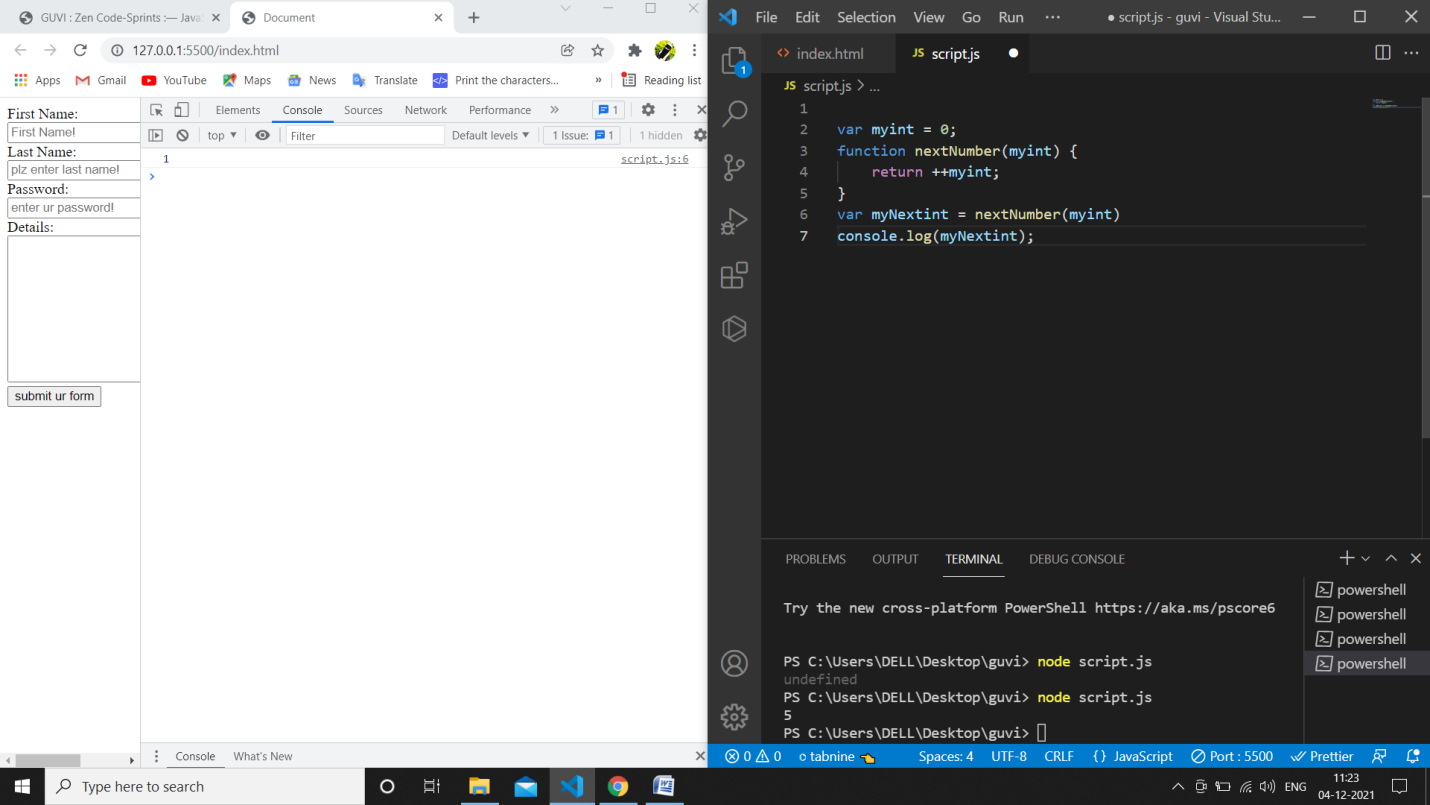
var myint = toInteger(mystr)

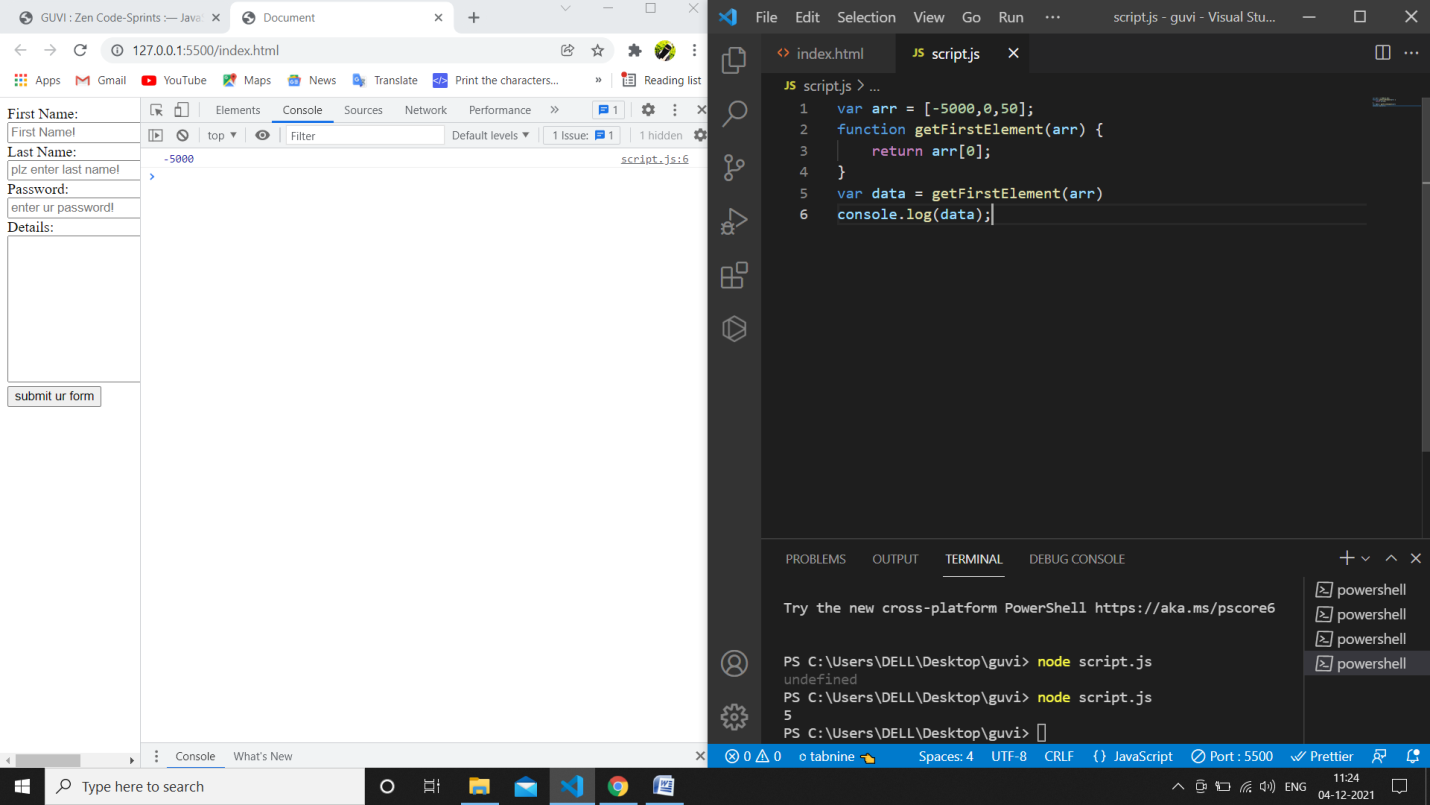
console.log(mystr);

o/p:



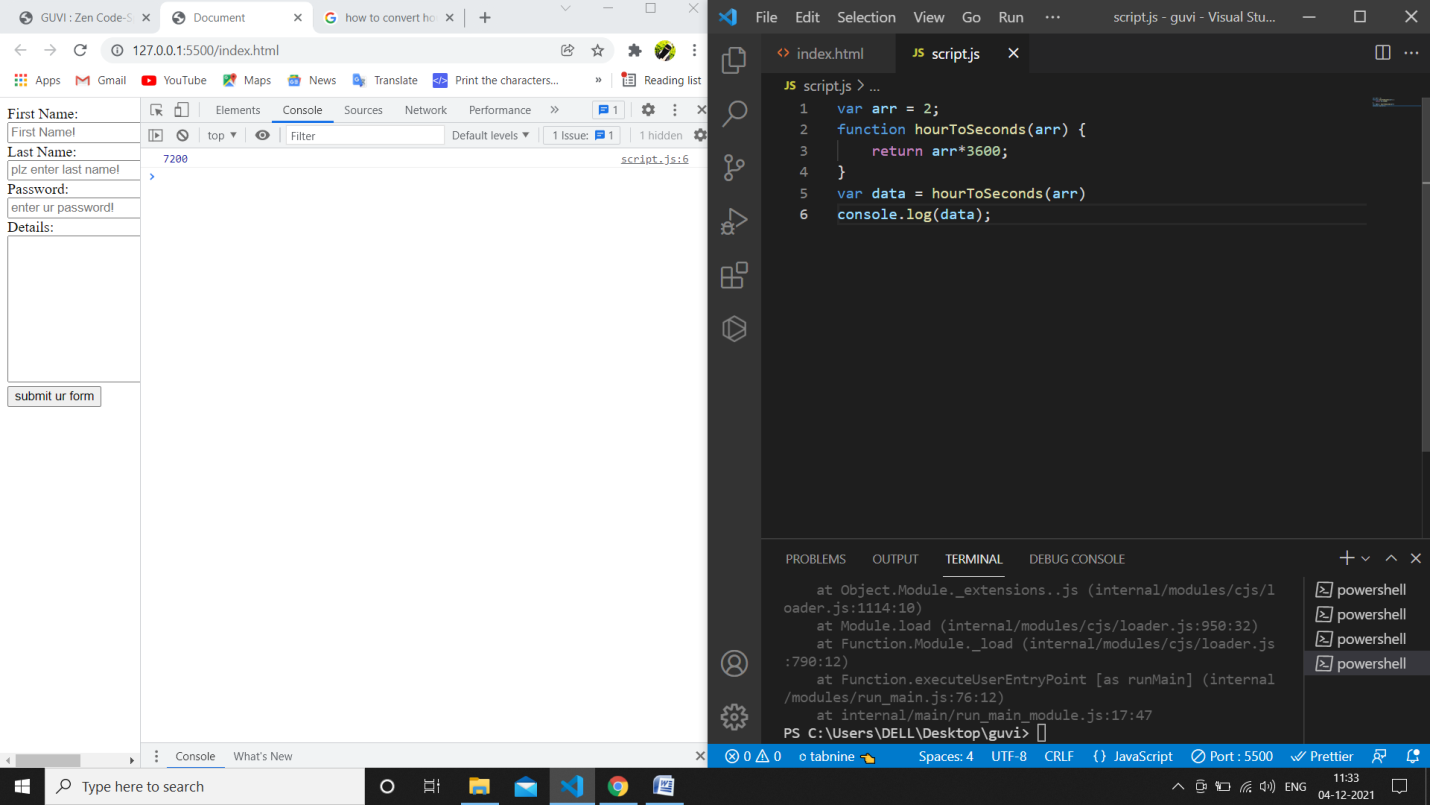
Create a function that takes a number as an argument, increments the number by +1 and returns the result.

Create a function that takes an array and returns the first element.

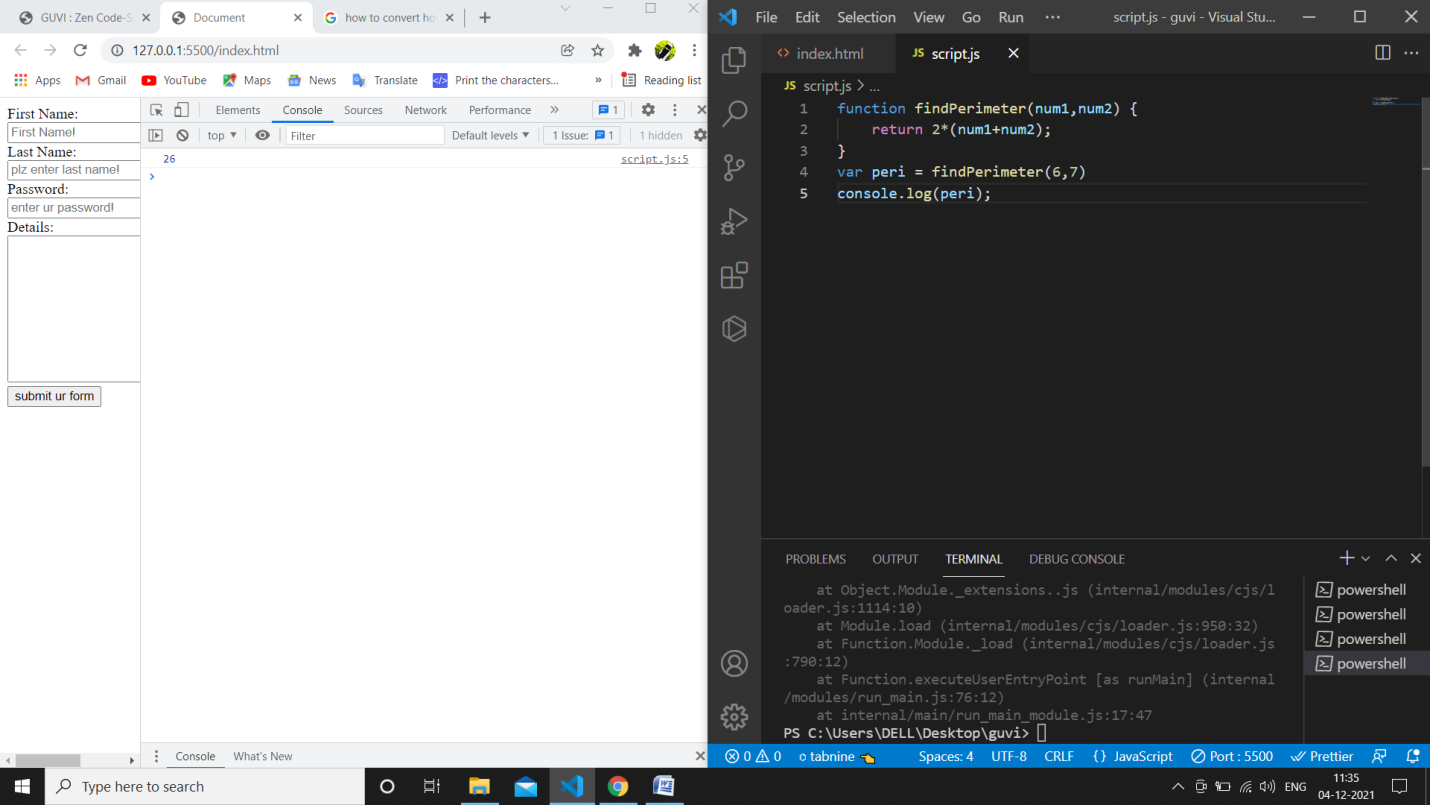


Convert Hours into Seconds

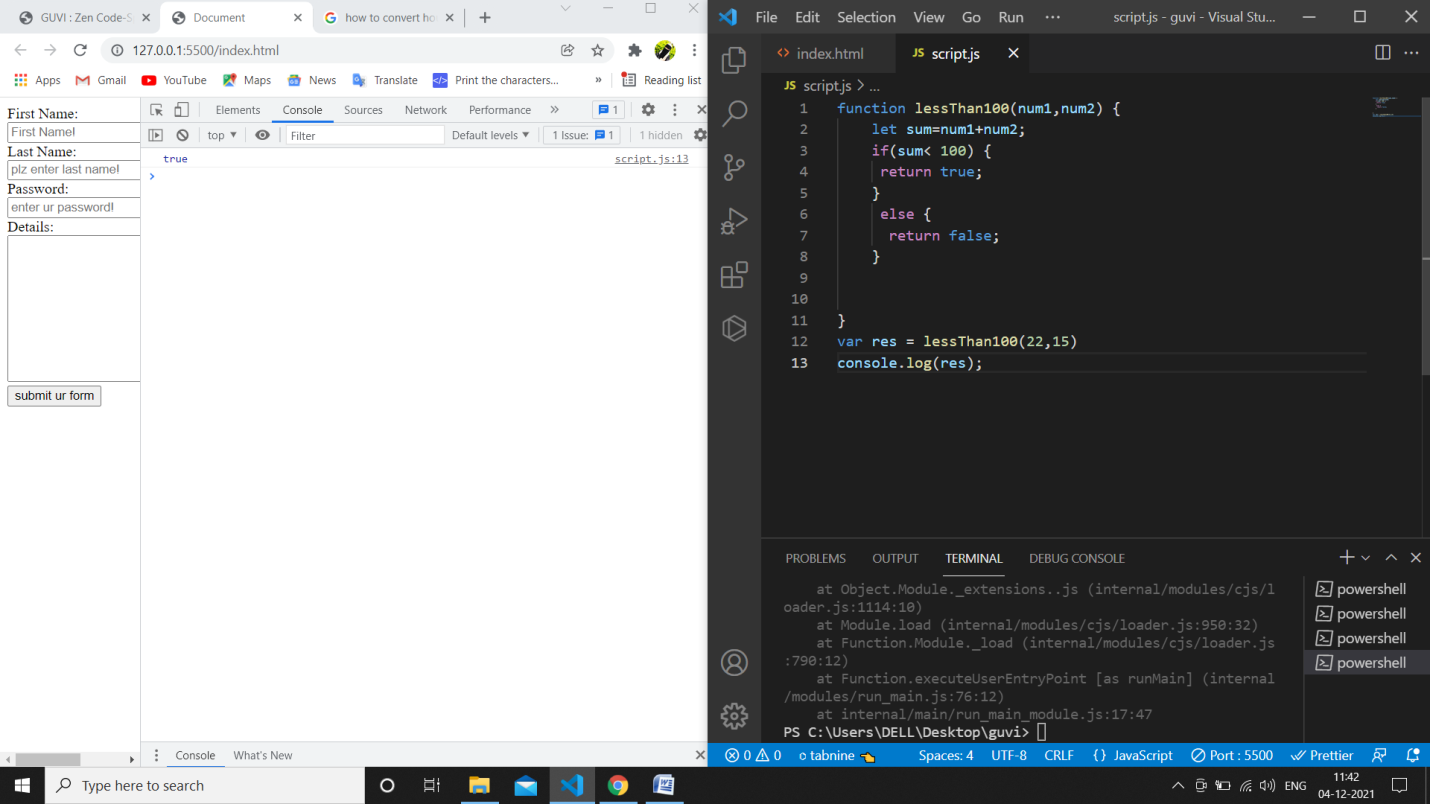
Write a function that converts hours into seconds.



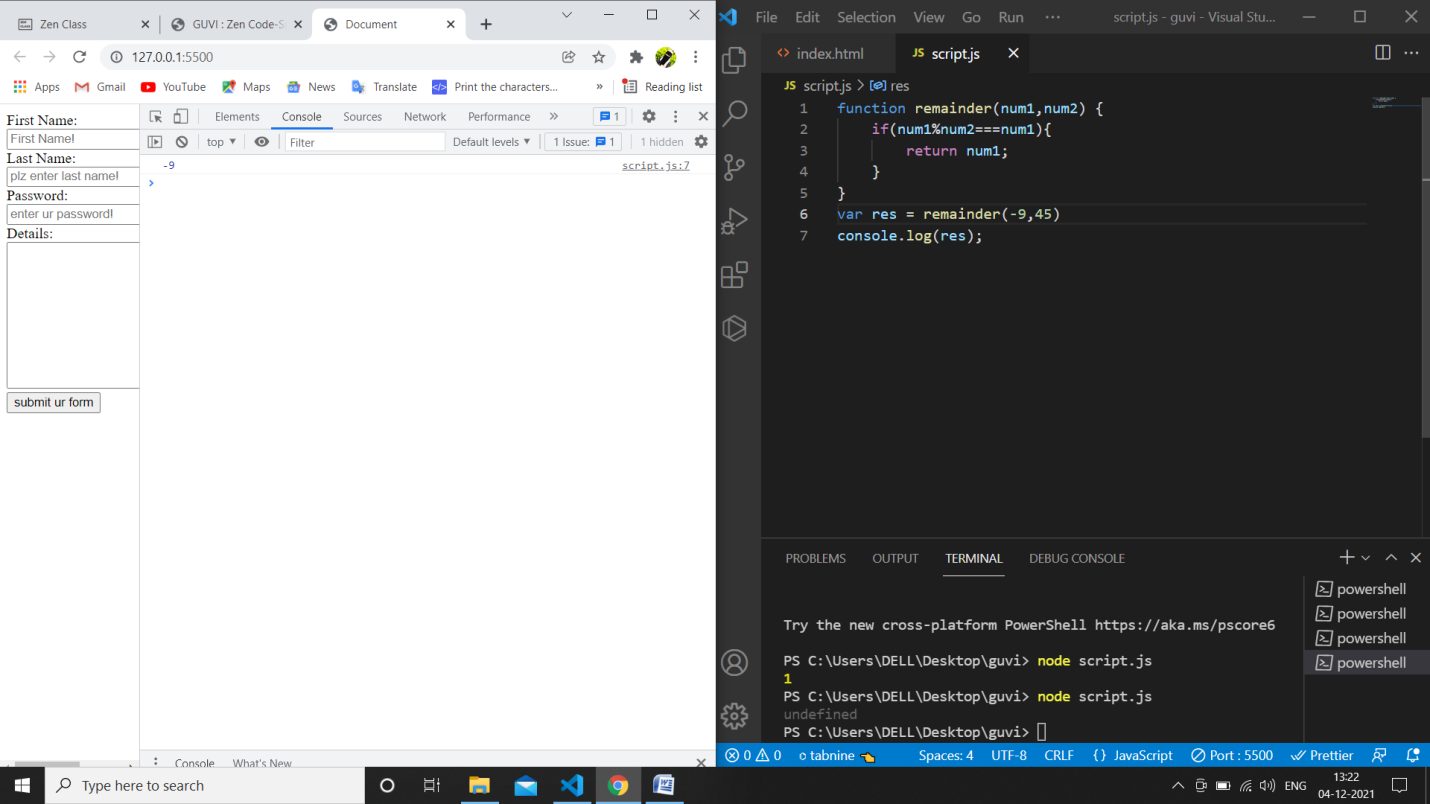
Find the Perimeter of a Rectangle  
Create a function that takes height and width and finds the perimeter of a rectangle.



Less Than 100?  
Given two numbers, return true if the sum of both numbers is less than 100. Otherwise return false.



There is a single operator in JavaScript, capable of providing the remainder of a division operation. Two numbers are passed as parameters. The first parameter divided by the second parameter will have a remainder, possibly zero. Return that value.

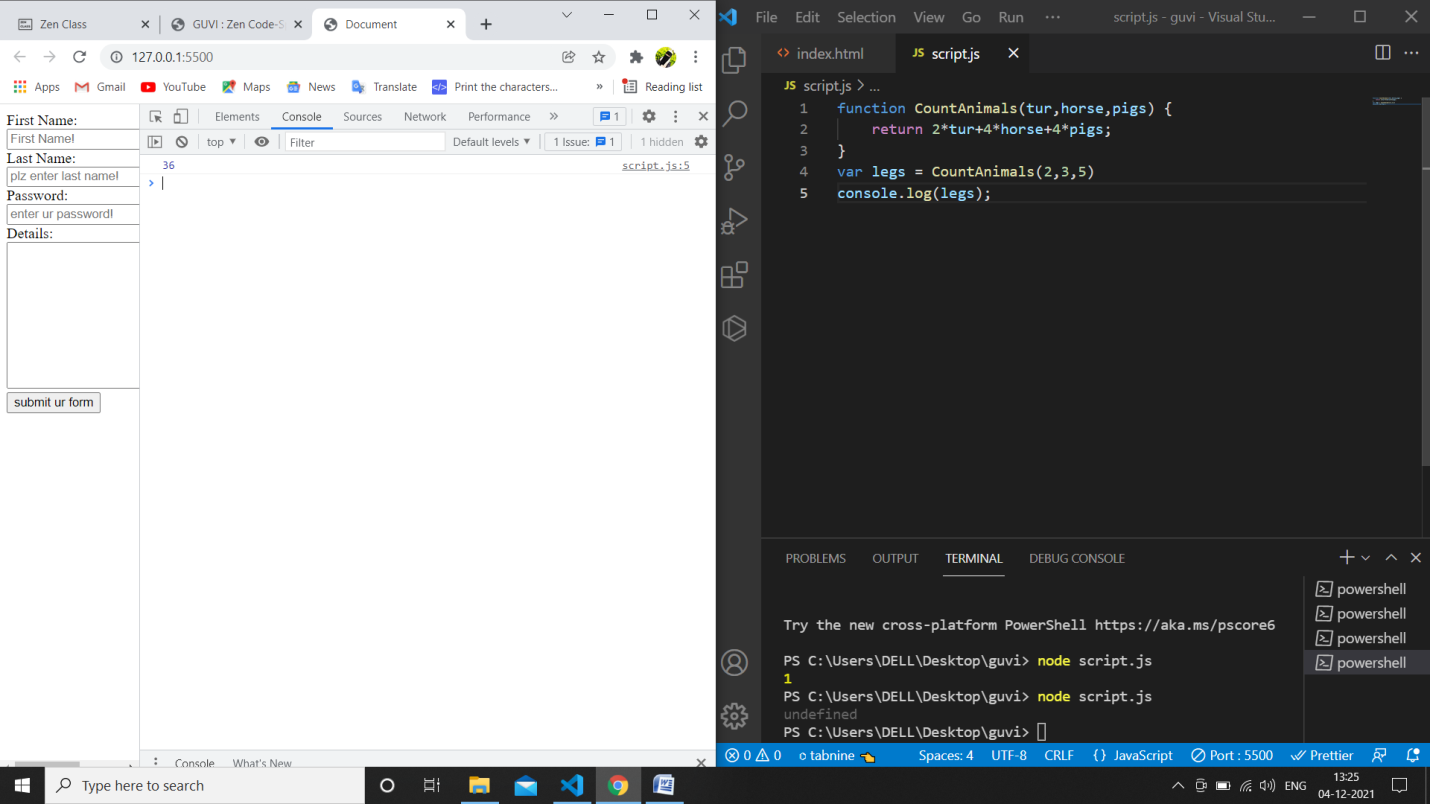


Old macdonald had a farm:

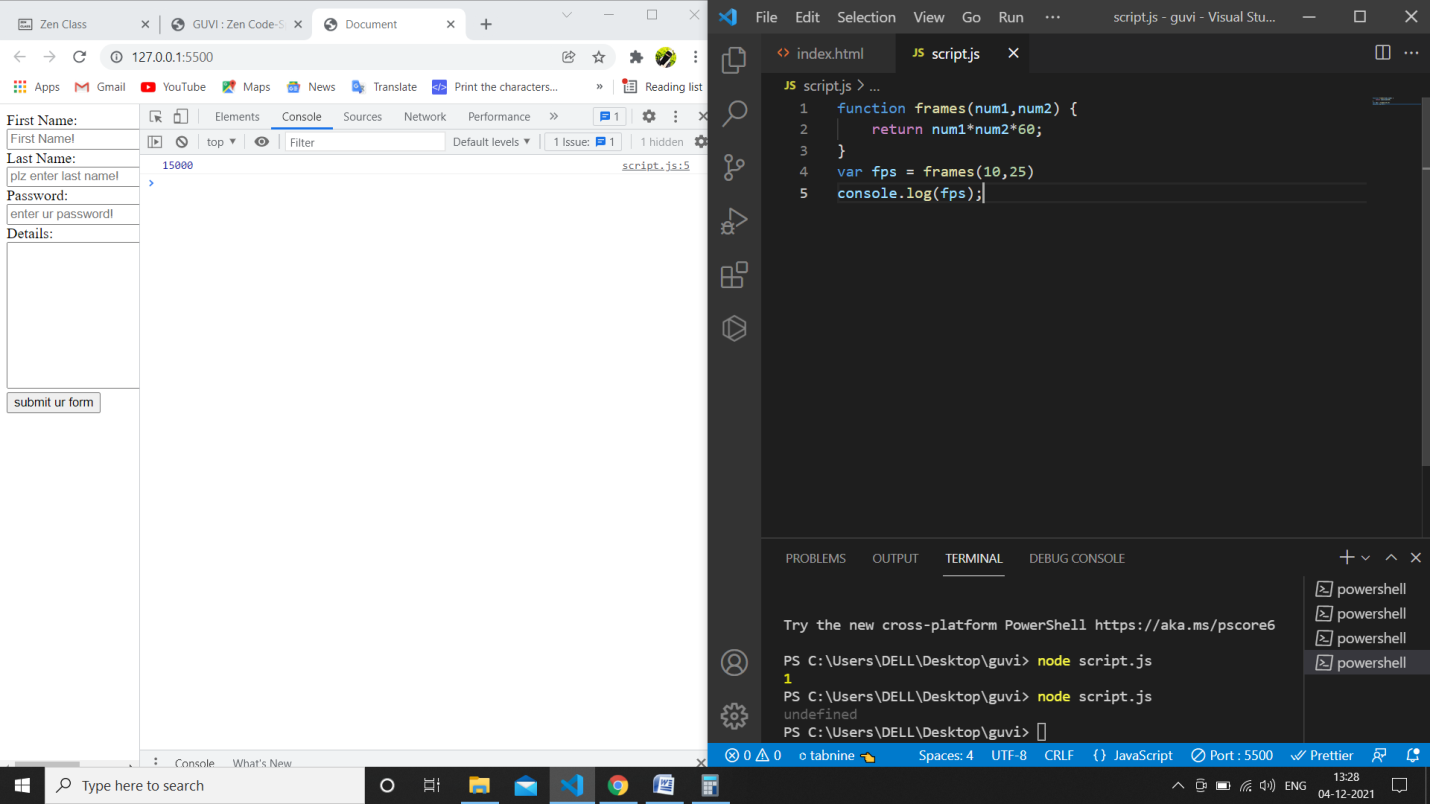
MacDonald is asking you to tell him how many legs can be counted among all his animals. The farmer breeds three species:

turkey = 2 legs  
horse = 4 legs  
pigs = 4 legs

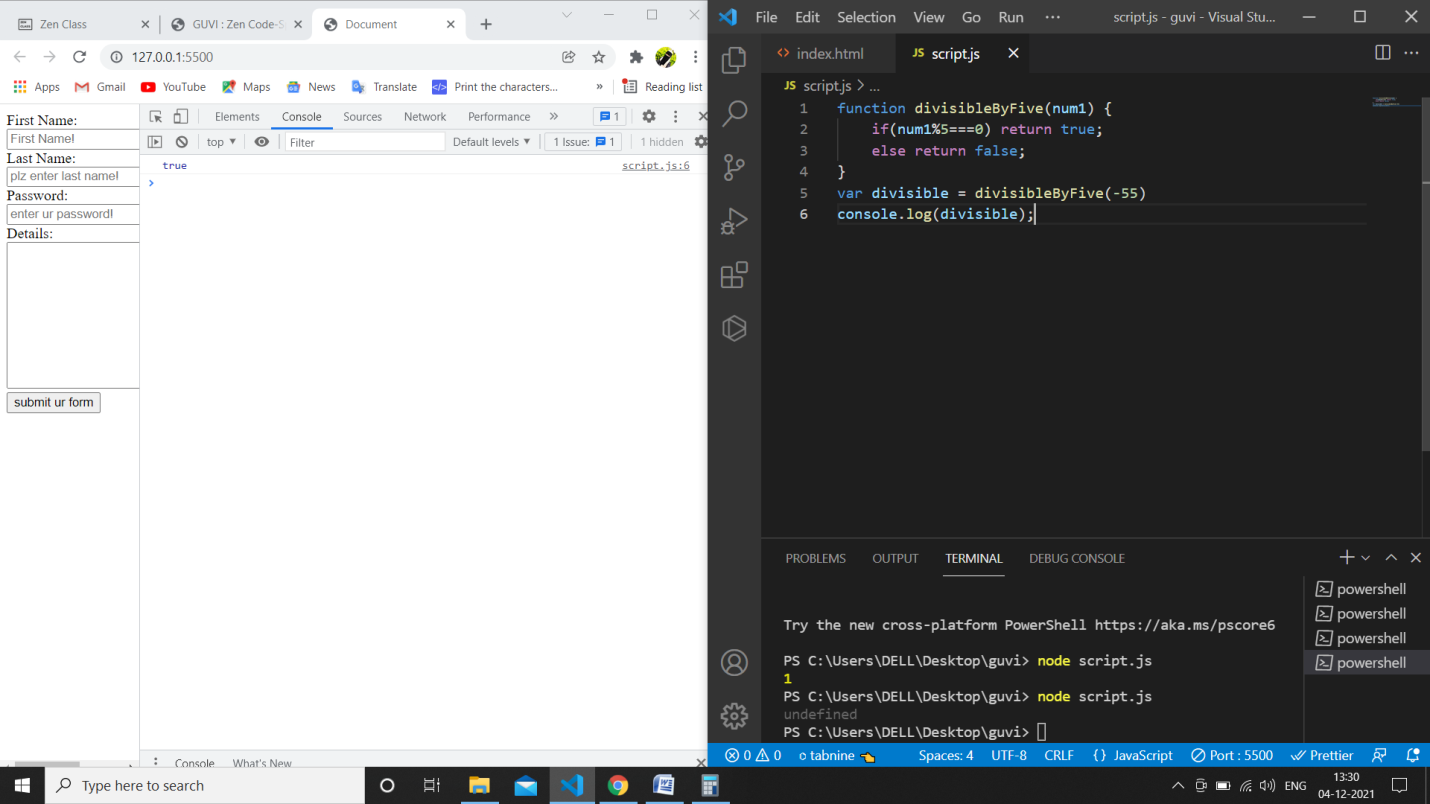
The farmer has counted his animals and he gives you a subtotal for each species. You have to implement a function that returns the total number of legs of all the animals.



Frames Per Second  
Create a function that returns the number of frames shown in a given number of minutes for a certain FPS.

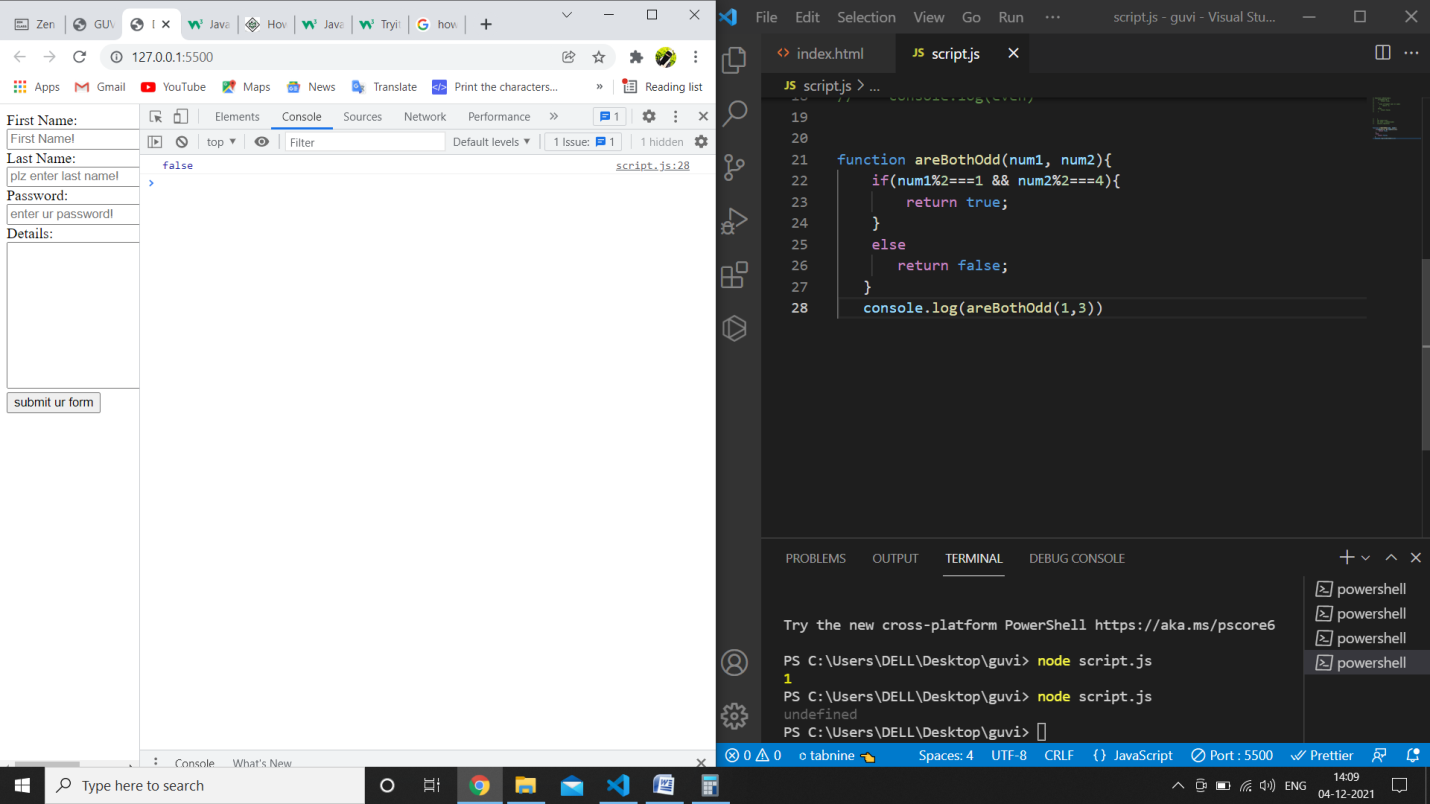


Check if an Integer is Divisible By Five  
Create a function that returns true if an integer is evenly divisible by 5, and false otherwise.

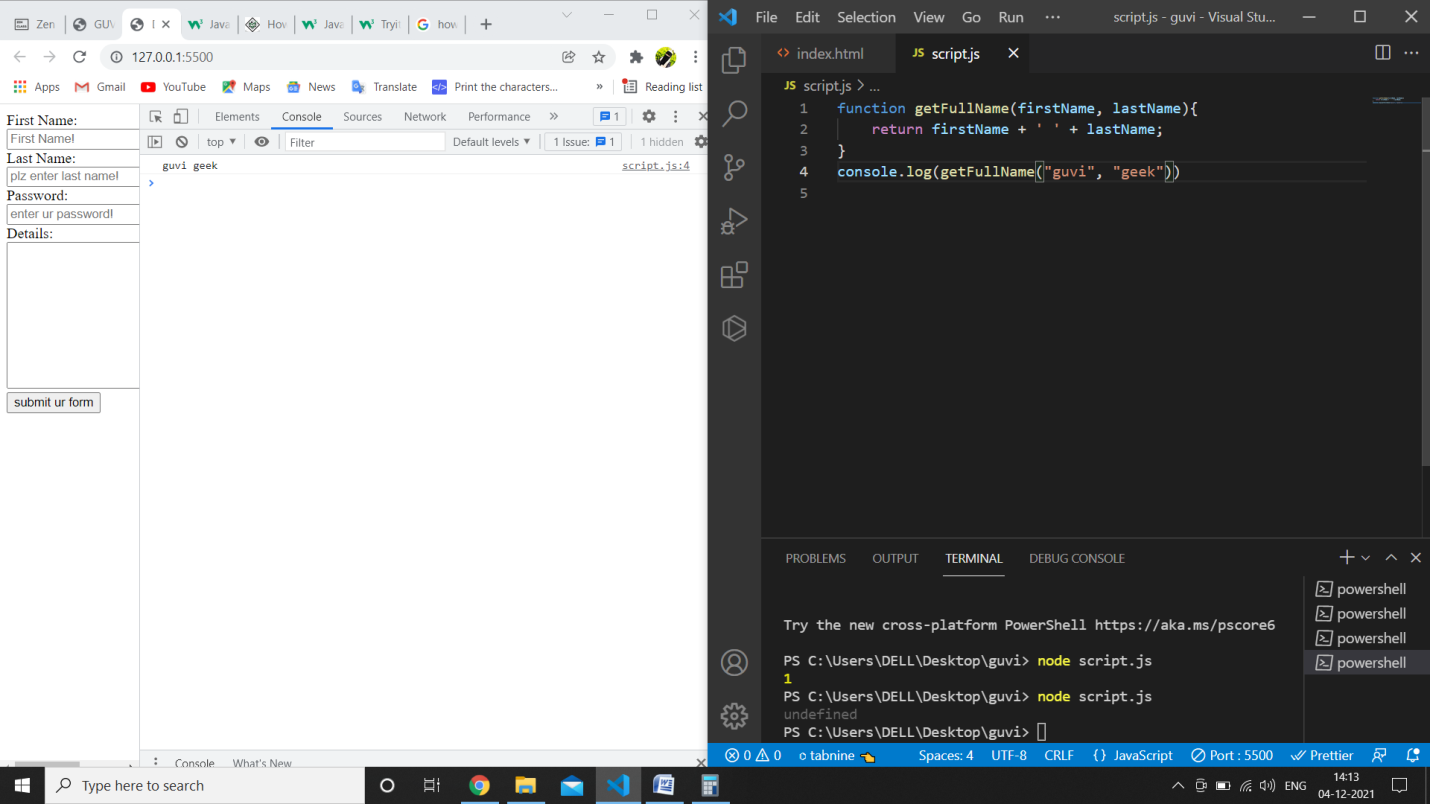


Write a function called “isEven”.  
Given a number, “isEven” returns whether it is even.

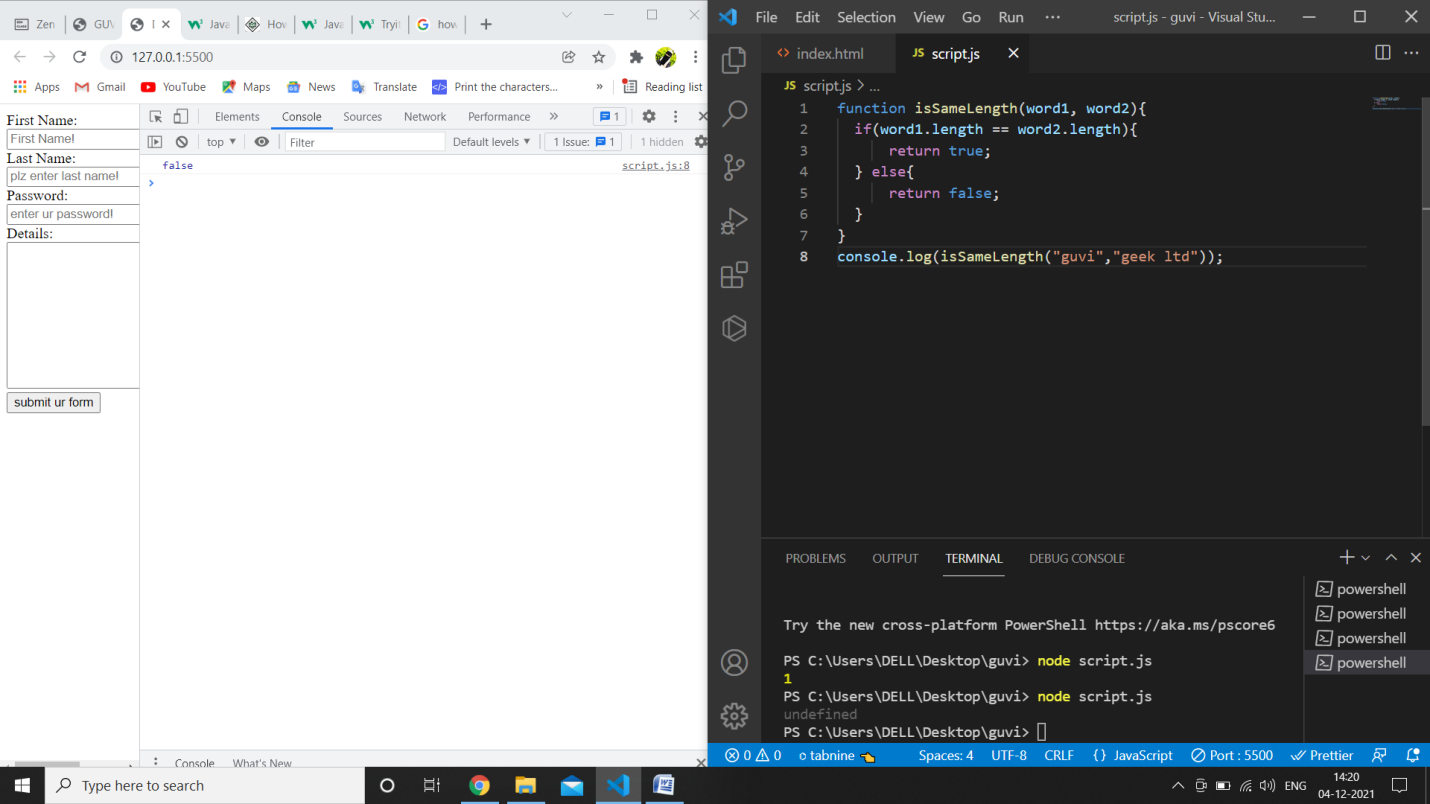
Write a function called “areBothOdd”.  
Given 2 numbers, “areBothOdd” returns whether or not both of the given numbers are odd.



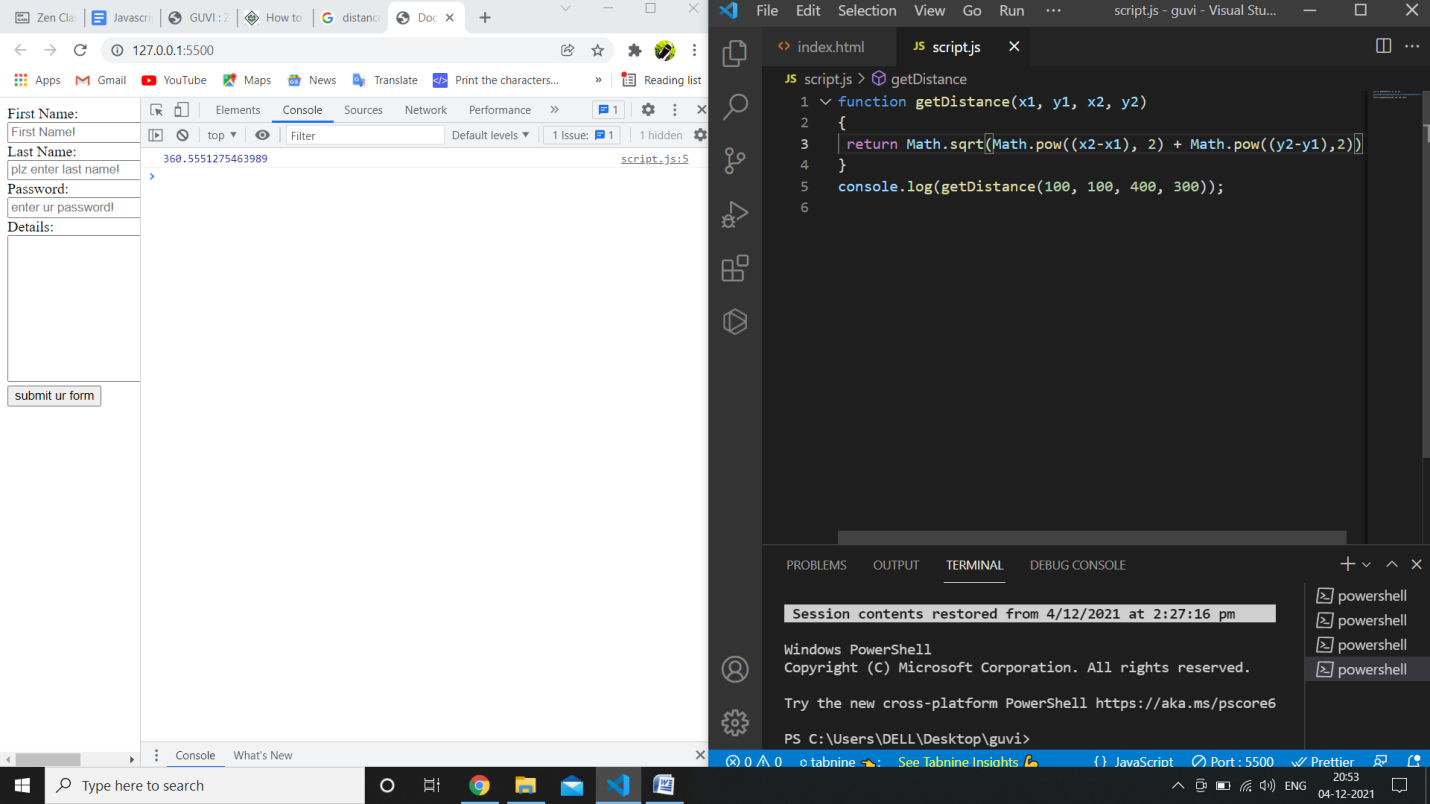
Write a function called “getFullName”.  
Given a first and a last name, “getFullName” returns a single string with the given first and last names separated by a single space.



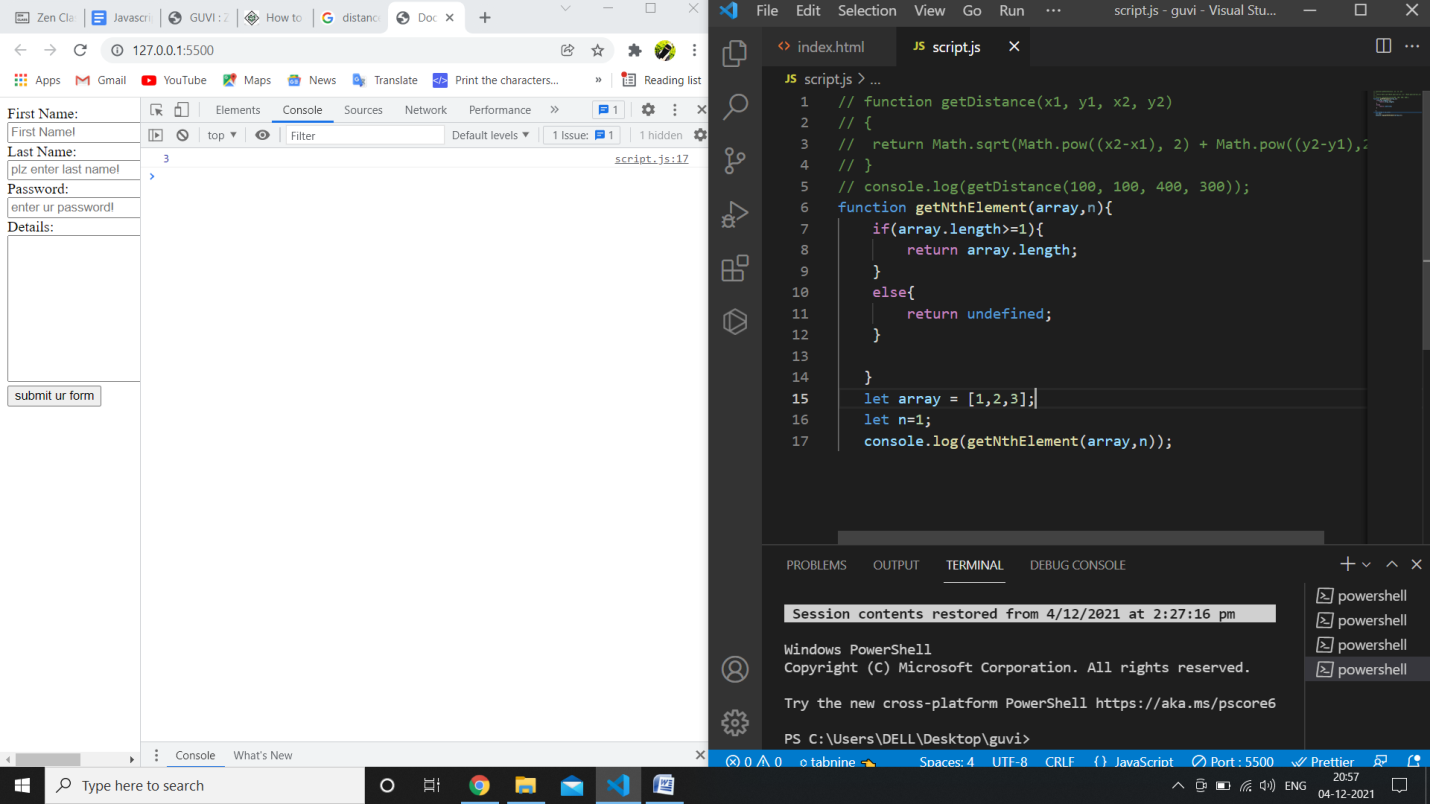
Write a function called “getLengthOfWord”.  
Given a word, “getLengthOfWord” returns the length of the given word.



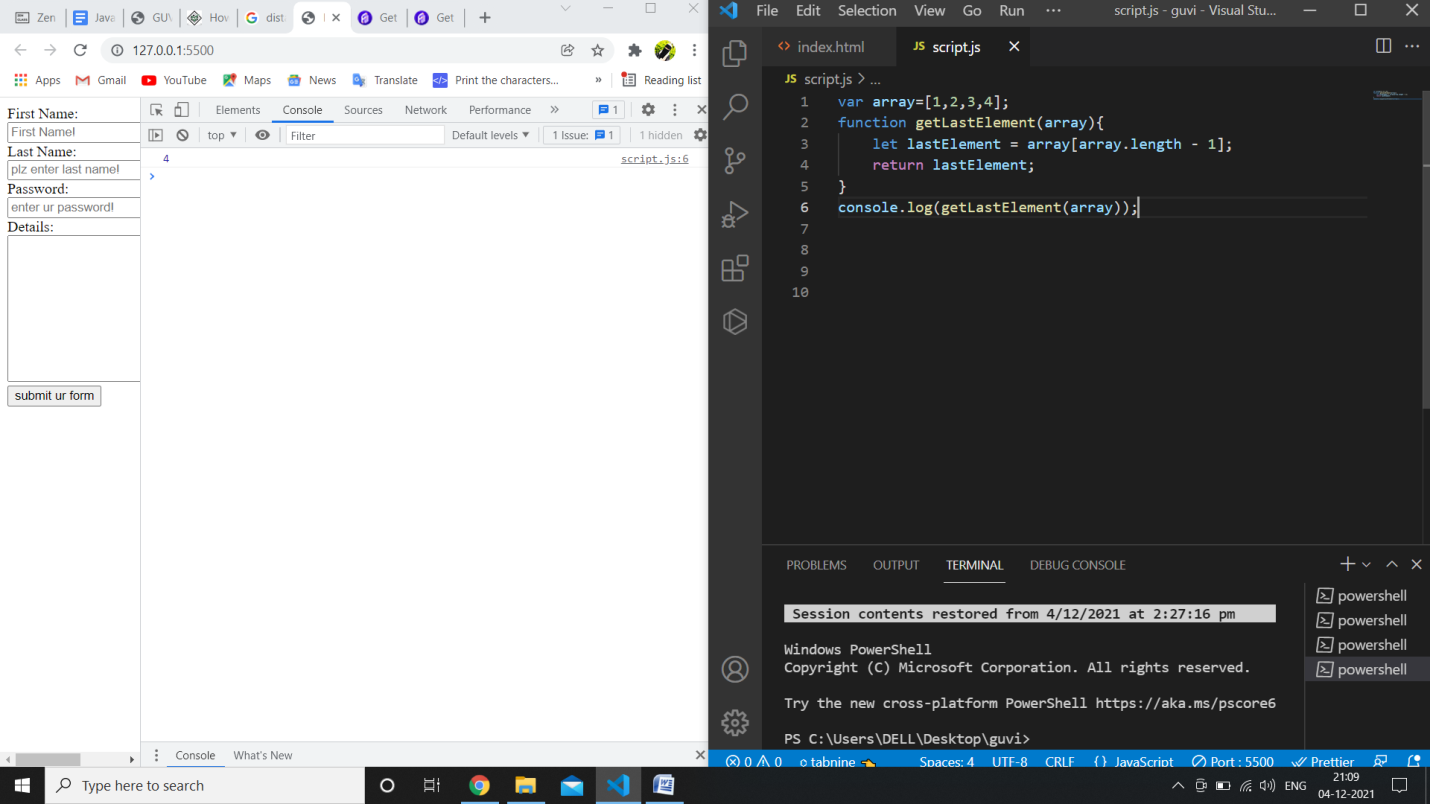
Create a function to calculate the distance between two points defined by their x, y coordinates



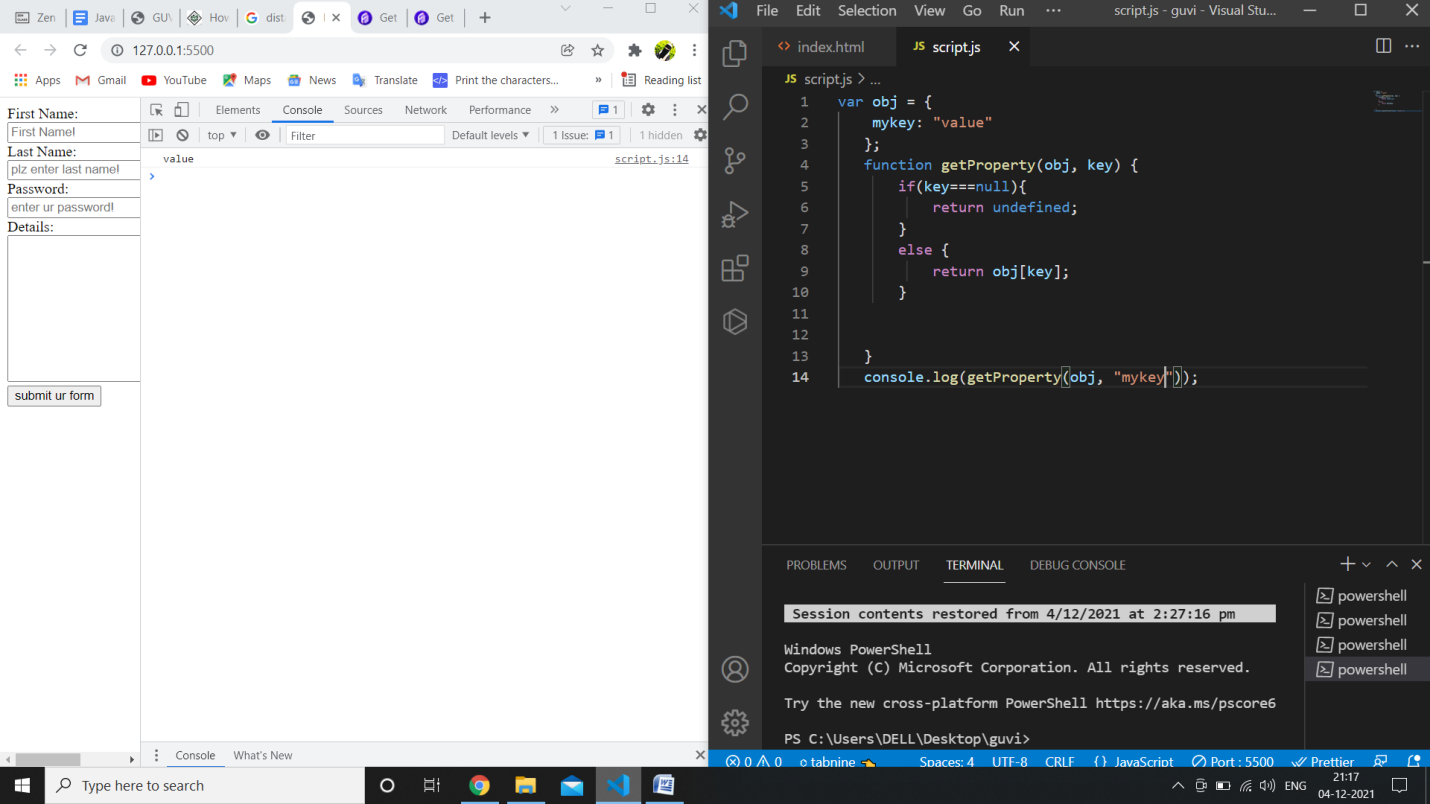
Write a function called “getNthElement”.  
Given an array and an integer, “getNthElement” returns the element at the given integer, within the given array. If the array has a length of 0, it should return ‘undefined’.  
Input:  
getNthElement([1, 3, 5], 1);  
Output:  
3



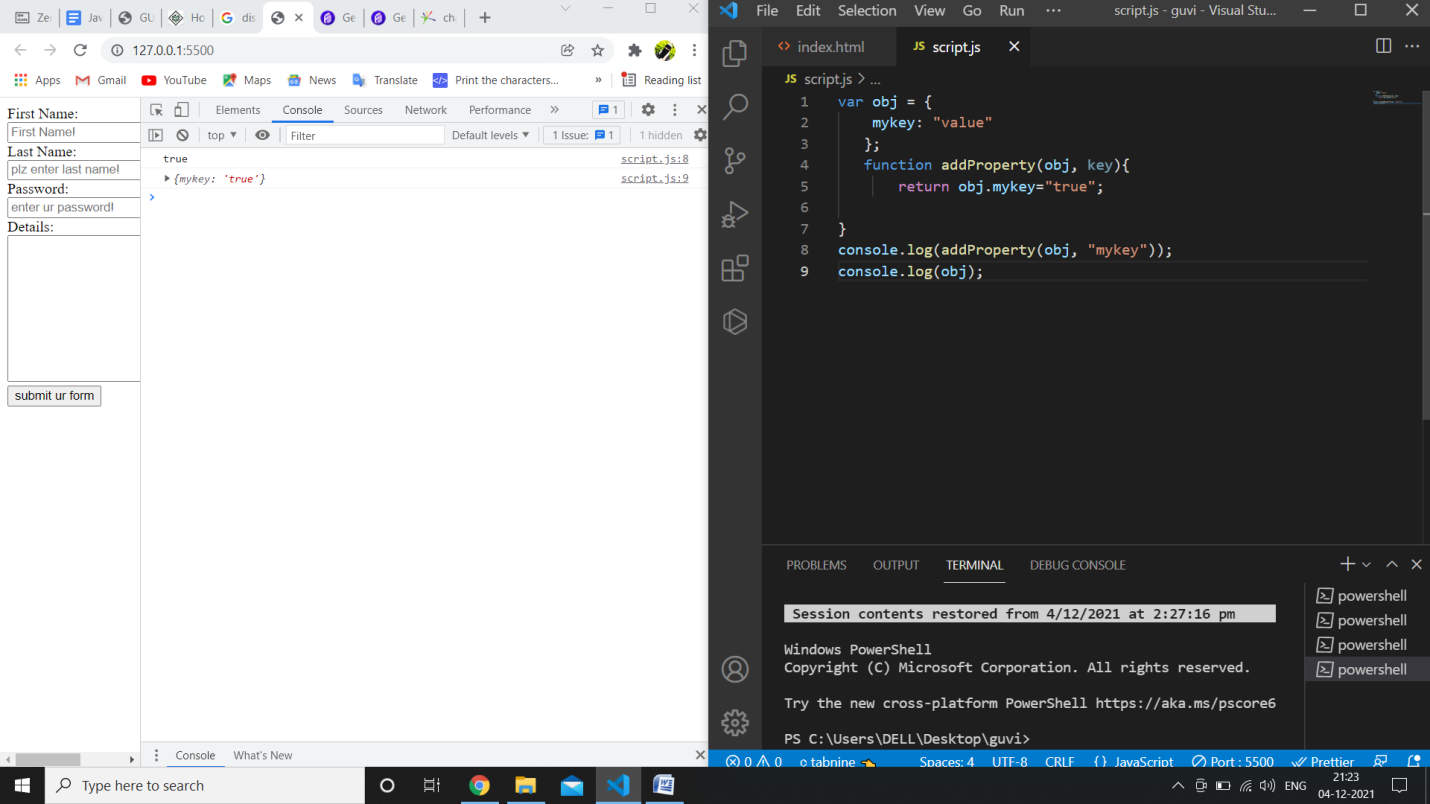
//last element



Write a function called “getProperty”.  
Given an object and a key, “getProperty” returns the value of the property at the given key. If there is no property at the given key, it should return undefined.



Write a function called “addProperty”.  
Given an object and a key, “addProperty” adds a new property on the given object with a value of true.



Write a function called “removeProperty”.  
Given an object and a key, “removeProperty” removes the given key from the given object.

---------------

Return an array, where the first element is the count of positives numbers and the second element is sum of negative numbers.

var arr = [-5, 10, -3, 12, -9, 5, 90, 0, 1];

function countPositivesSumNegatives(arr) {

let i;

arr=arr.sort();

console.log(arr);

let sum=0;

let count=0;

let result;

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

    if(arr[i]<0){

        sum=sum+arr[i];

    }

    else if(arr[i]>0){

        count=count+1;

}

}

result=console.log(sum,count);

return result;

}

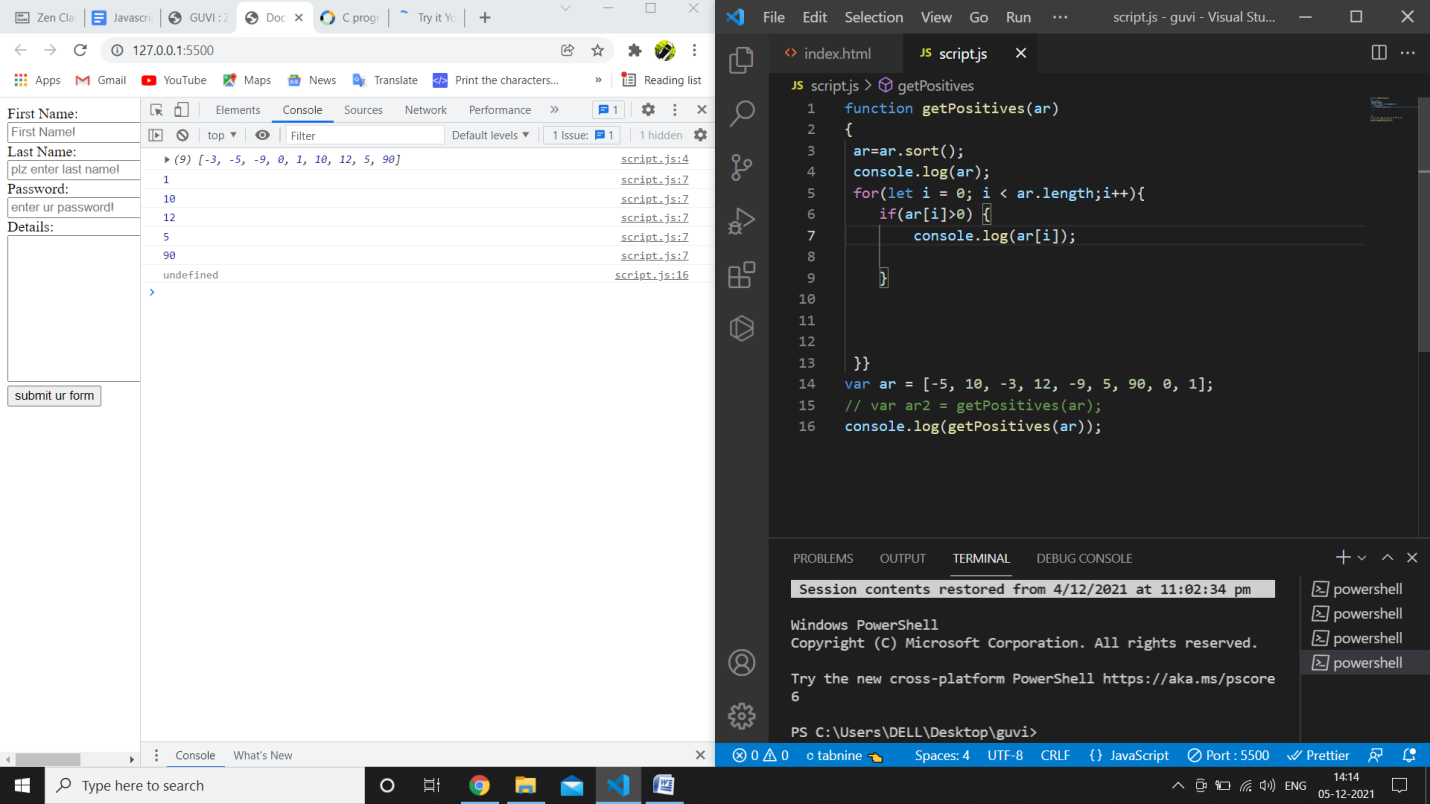
console.log(countPositivesSumNegatives(arr));

o/p:

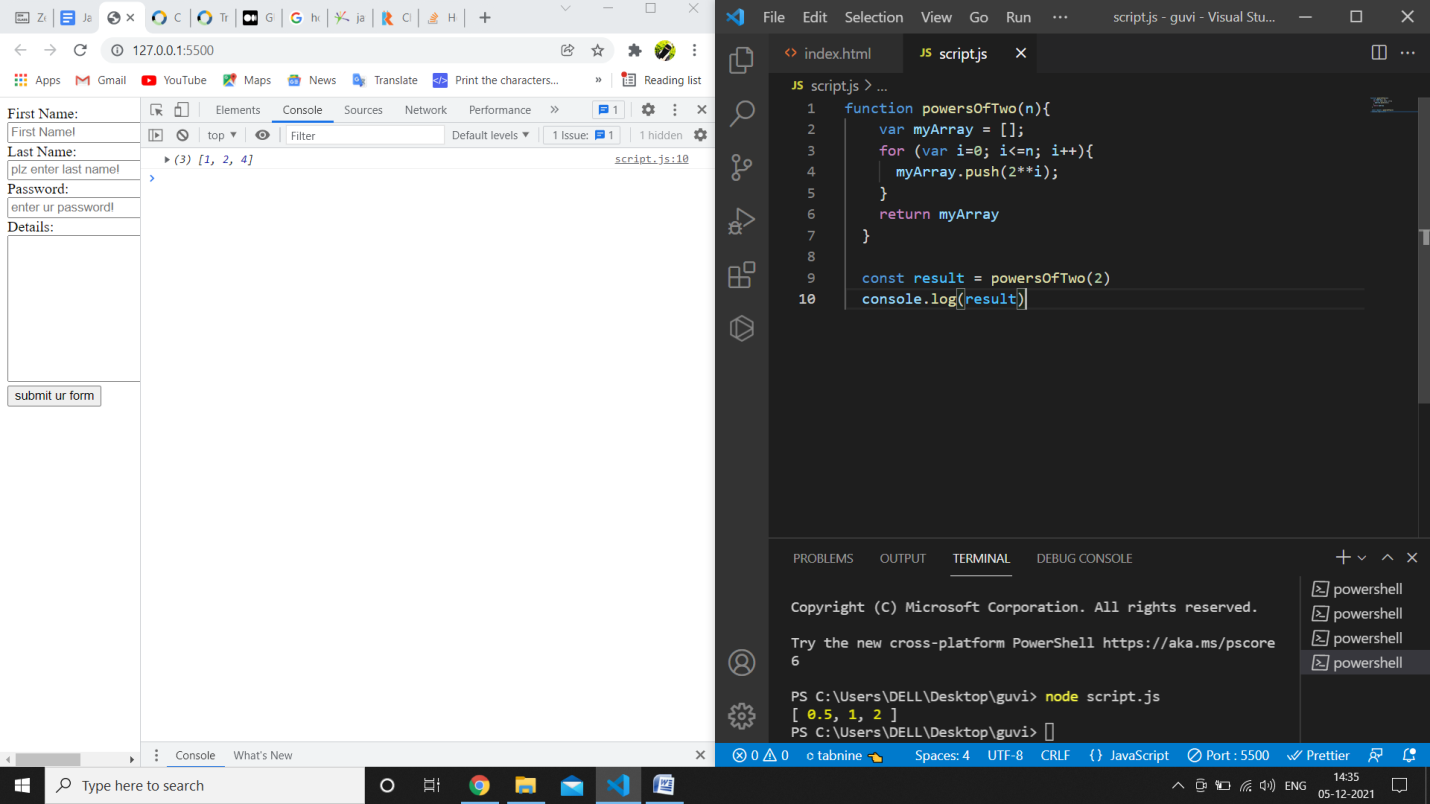
[-3, -5, -9, 0, 1, 10, 12, 5, 90]

-17 5

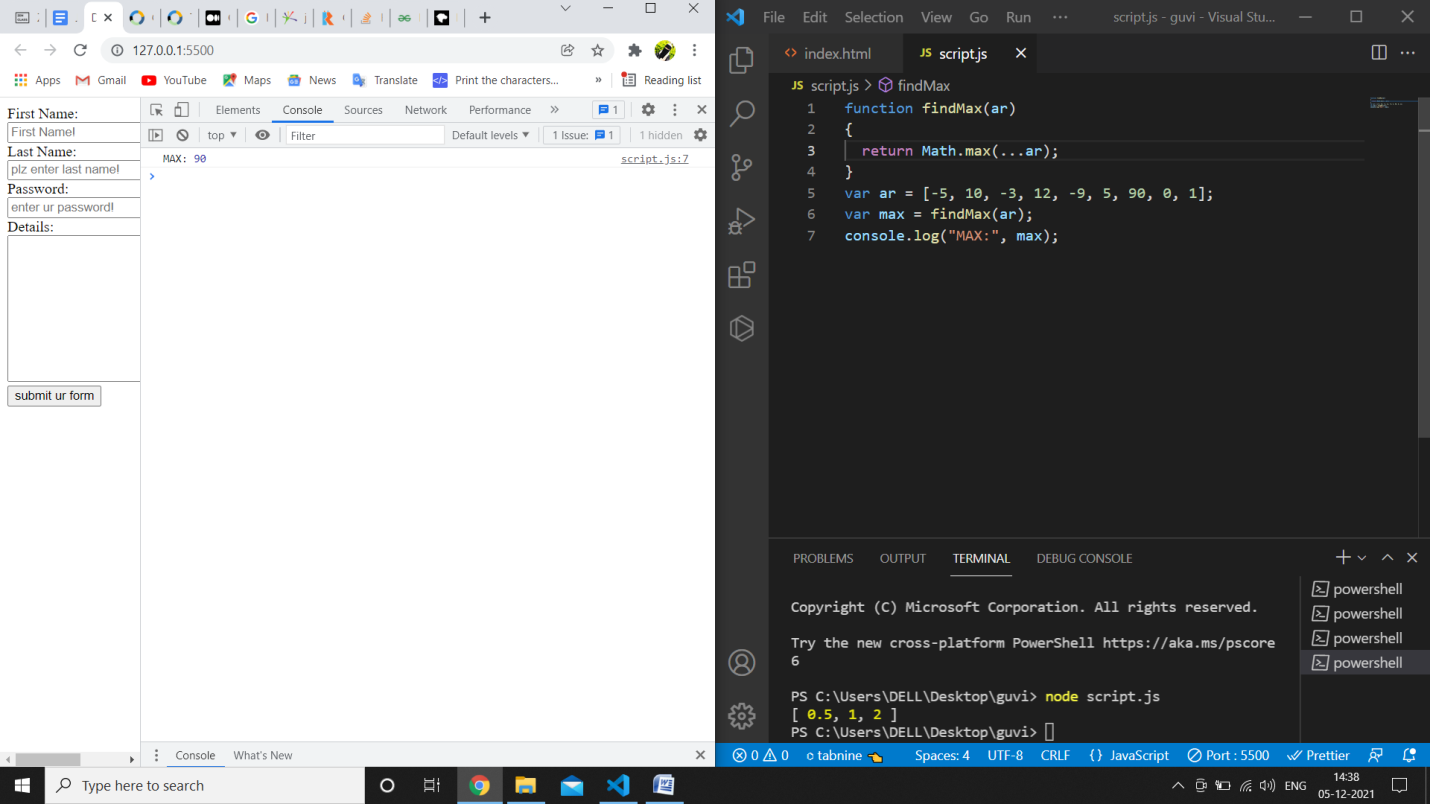
//positive elements



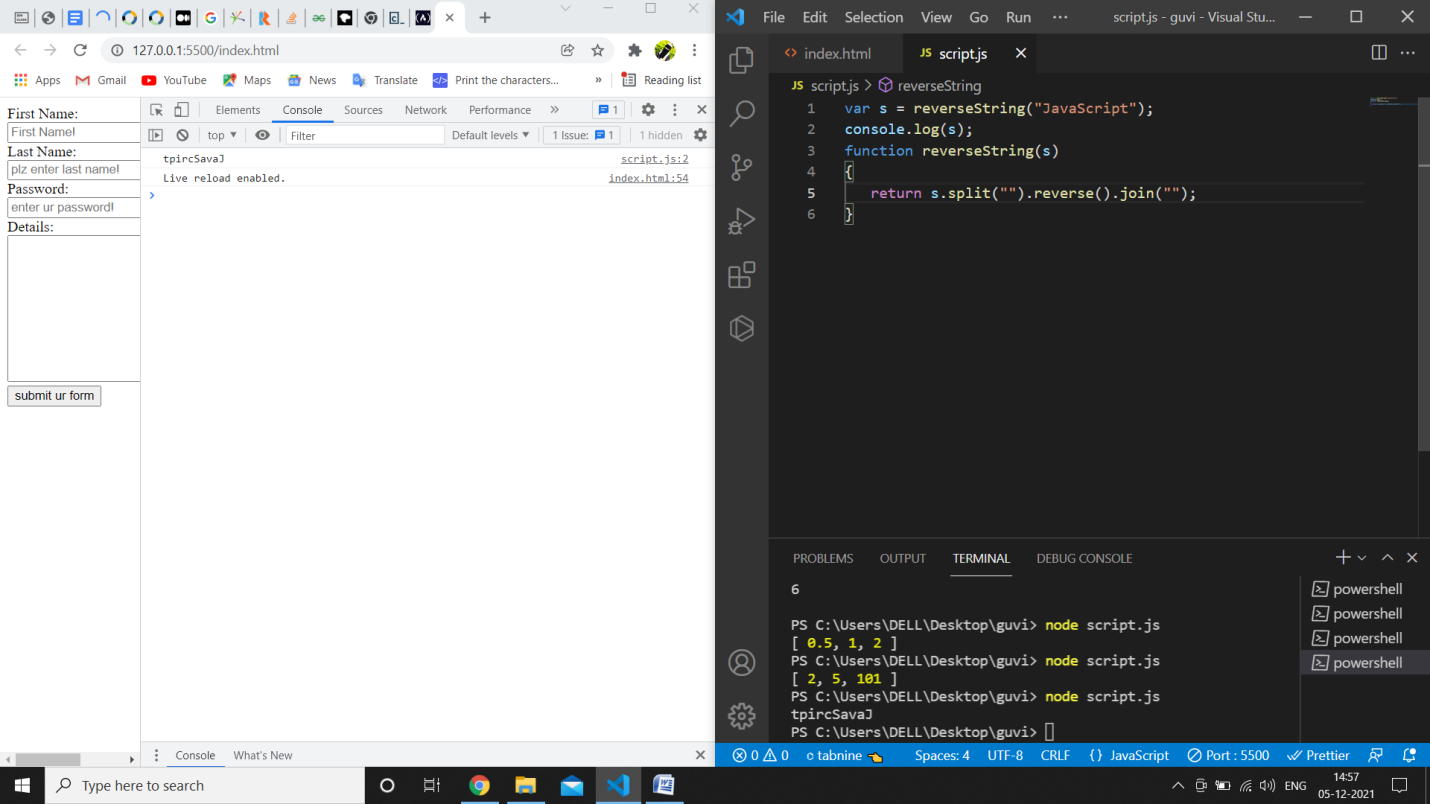
Write a function `powersOfTwo` which will return list of all powers of 2 from 0 to n (where n is an exponent).



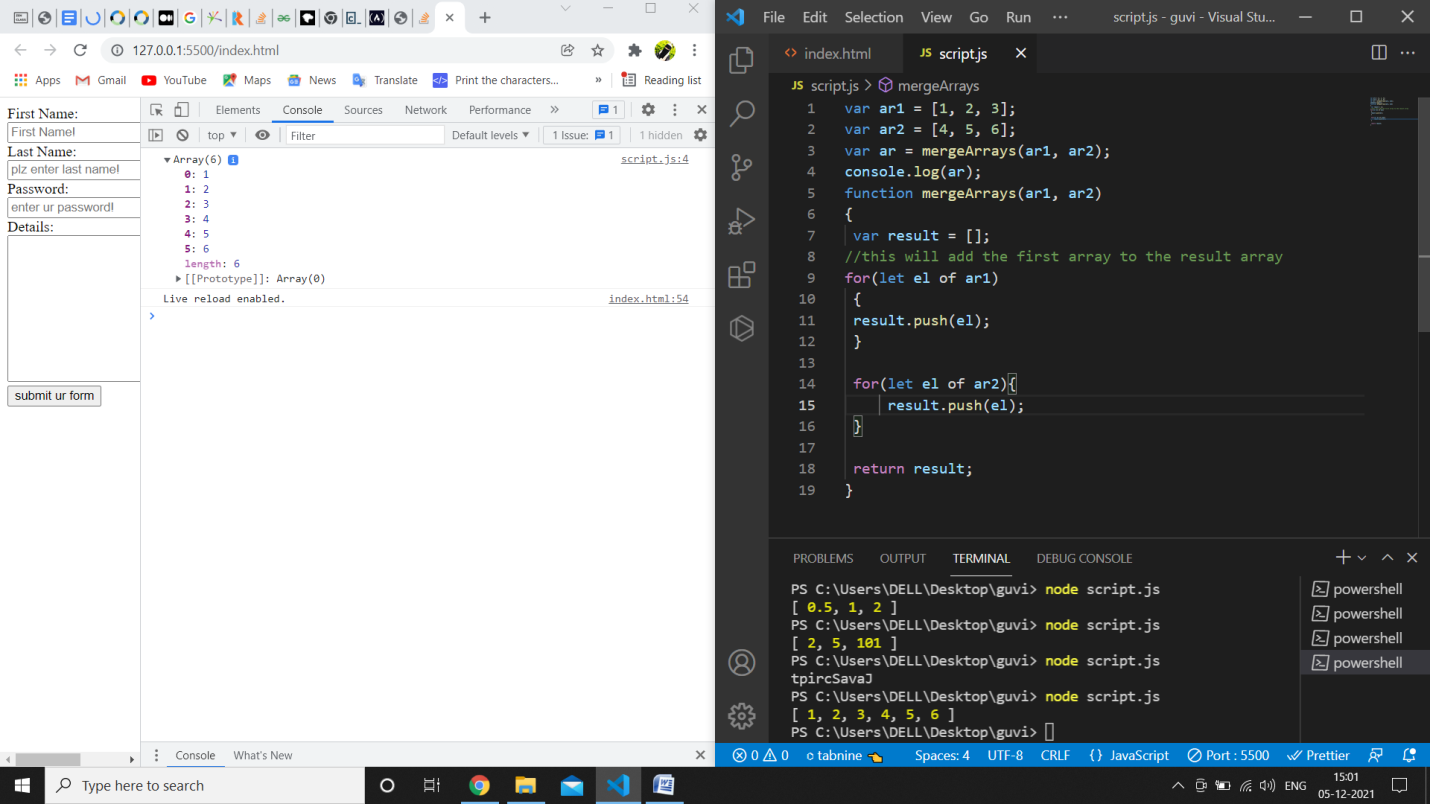
Find the maximum number in an array of numbers



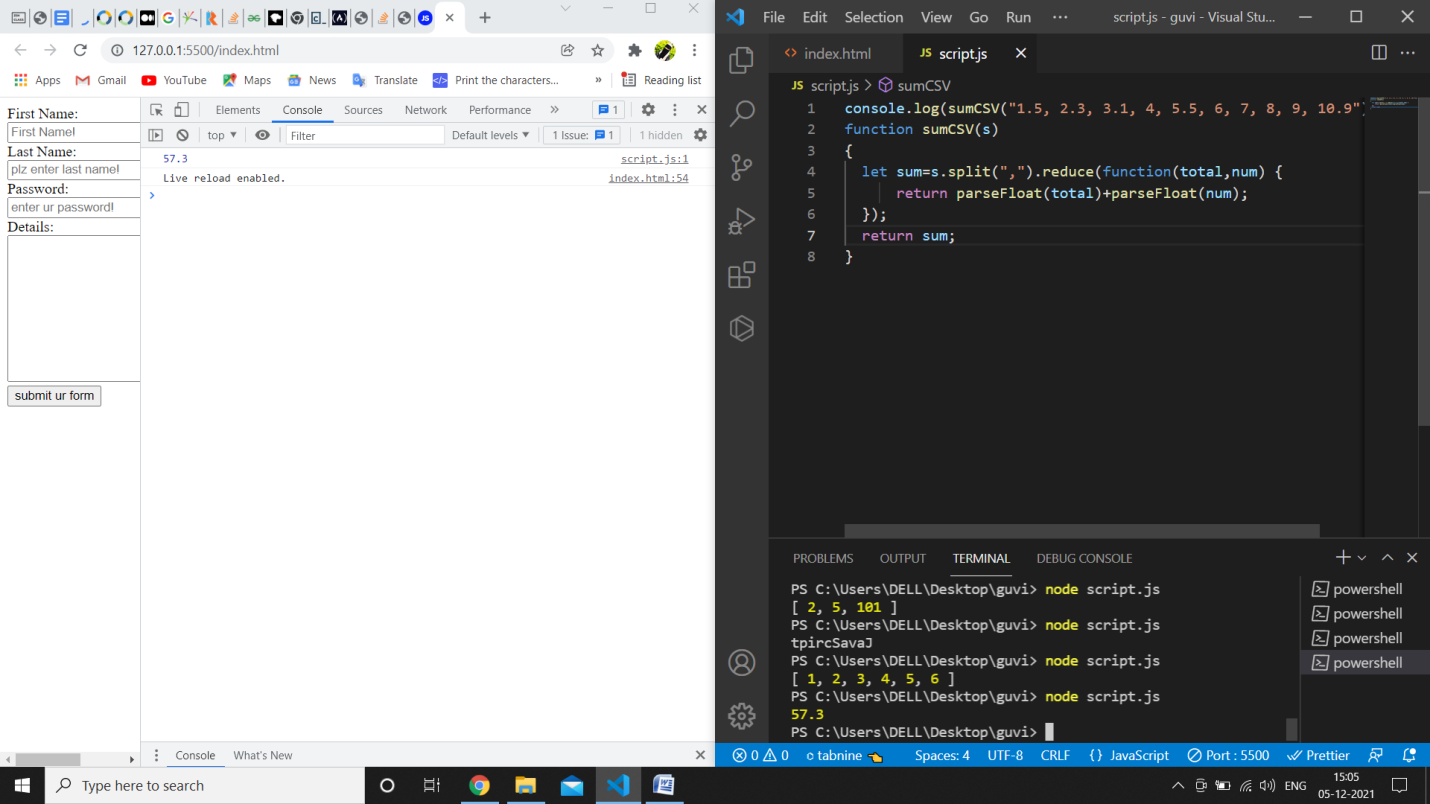
Reverse a string



Create a function that will merge two arrays and return the result as a new array



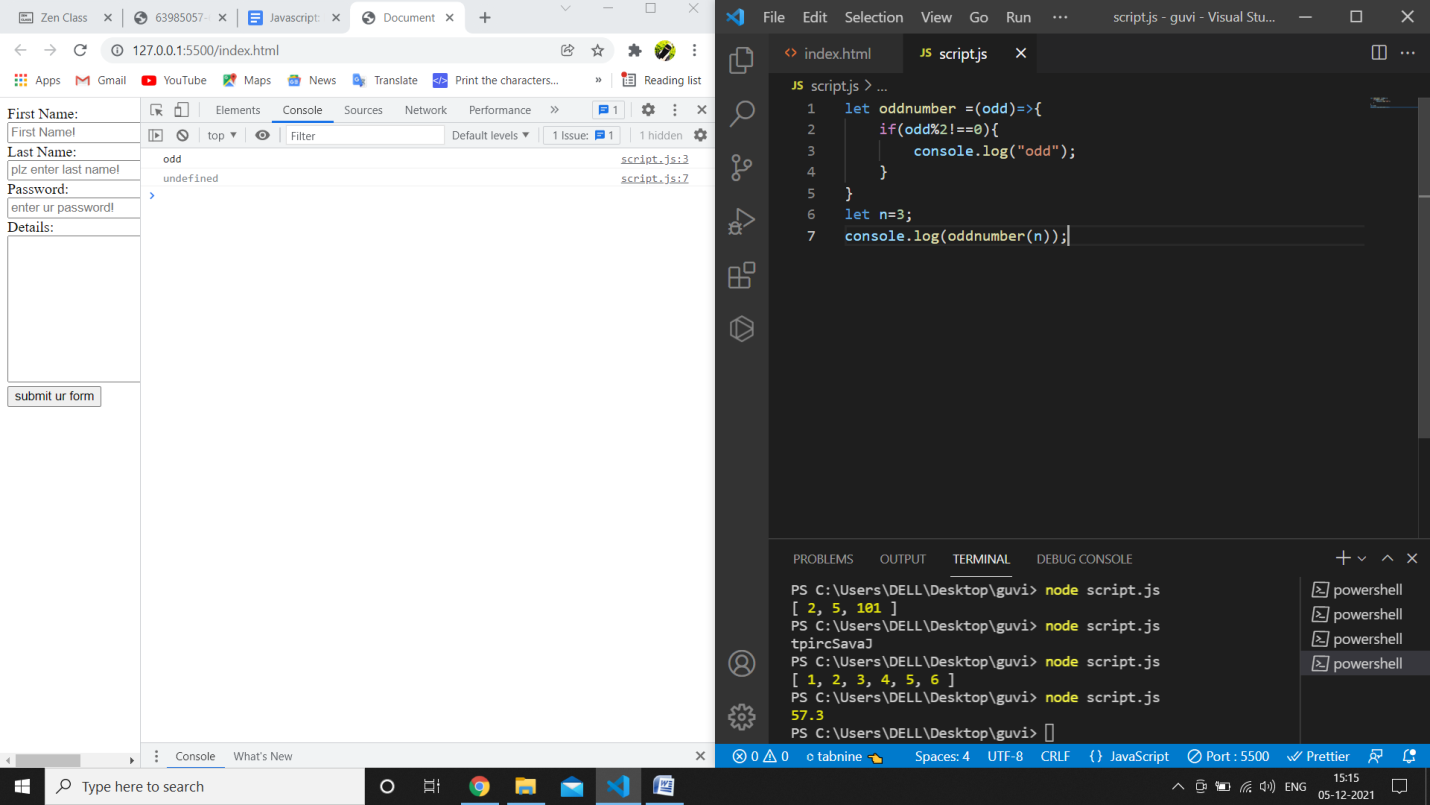
Calculate the sum of numbers received in a comma delimited string



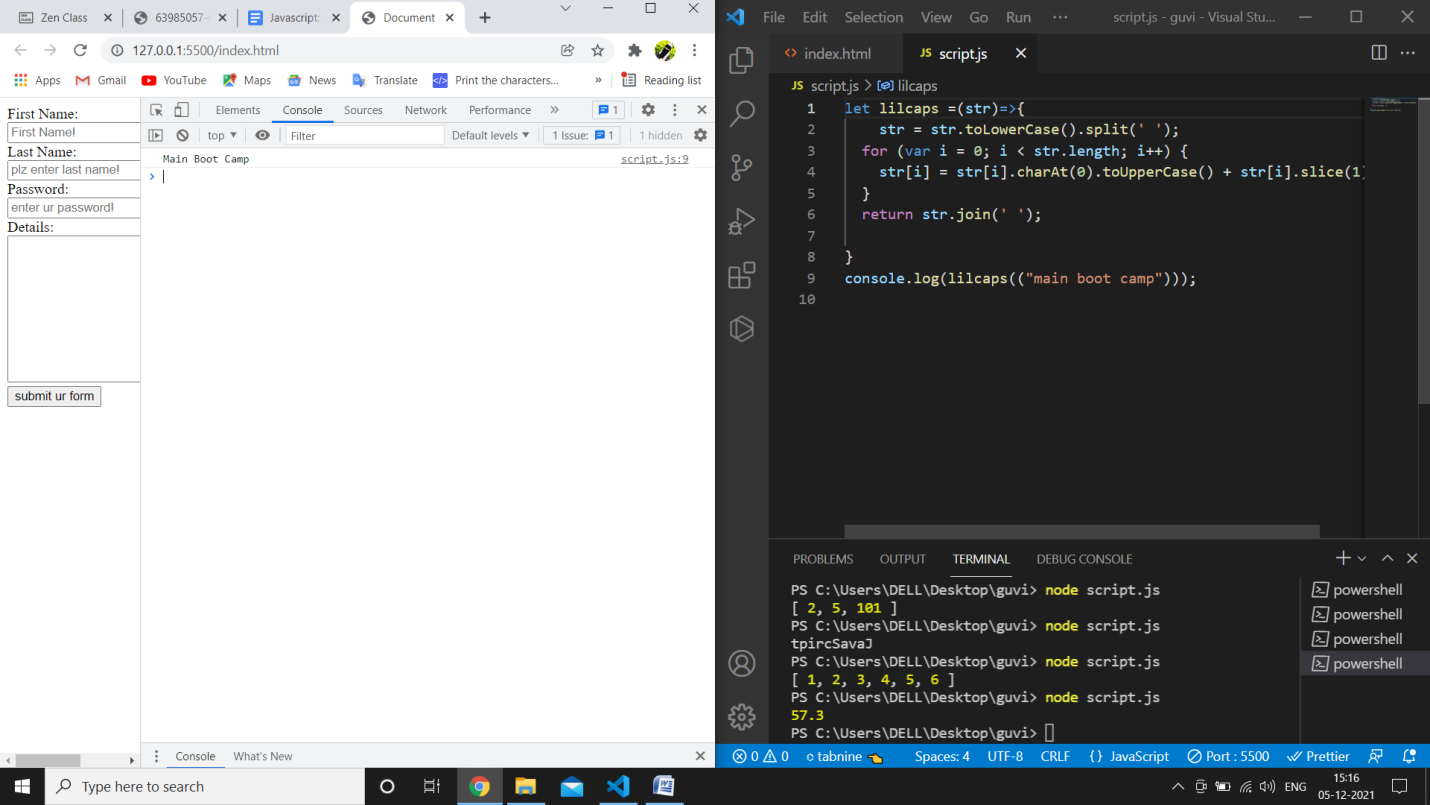
----------------------------------------------------------------------------------------------------------------------

1. Do the below programs in arrow functions
   1. Print odd numbers in an array
   2. Convert all the strings to title caps in a string array
   3. Sum of all numbers in an array
   4. Return all the prime numbers in an array
   5. Return all the palindromes in an array

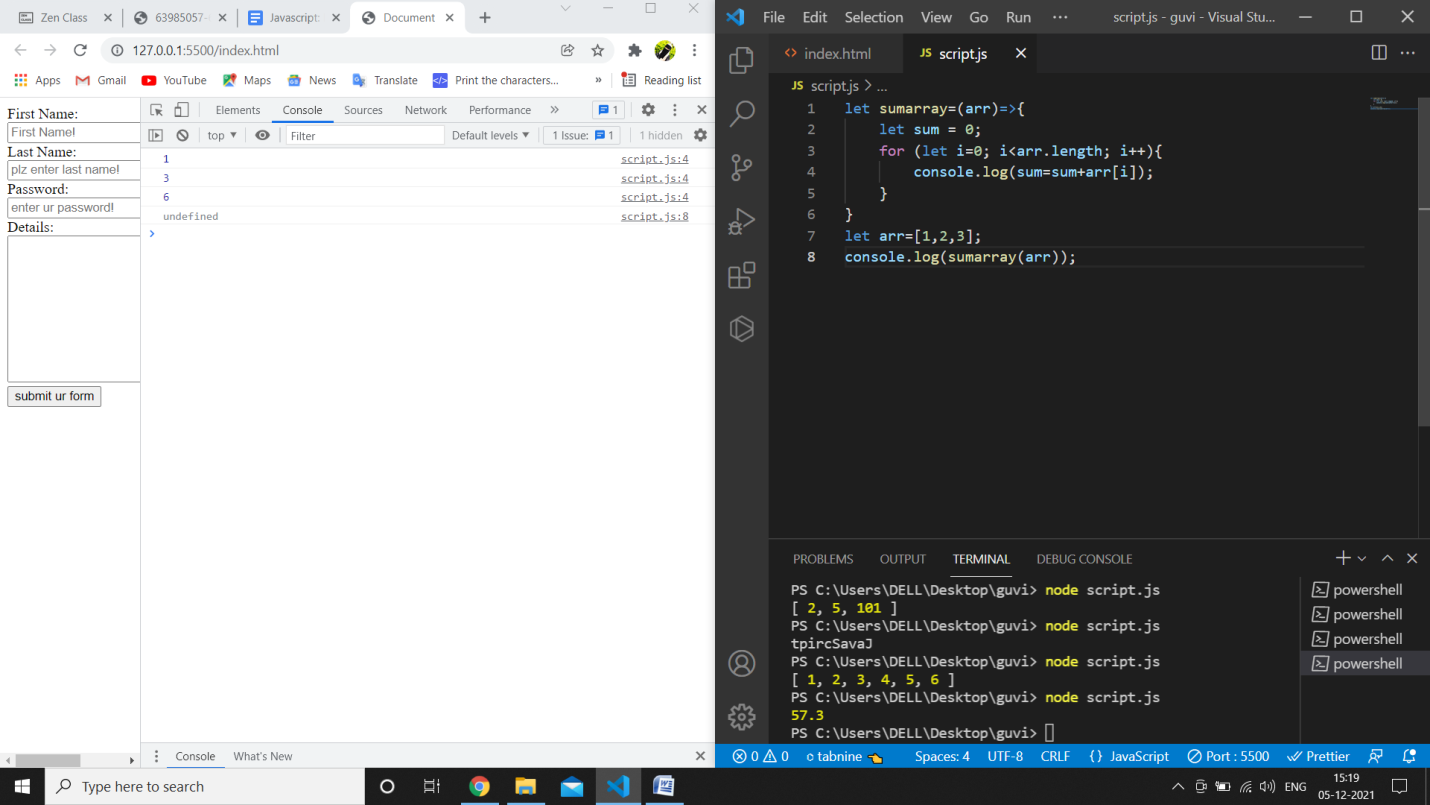
a.code



b.code



c.code



d.code:

let primes = (arr) => {

    return arr.filter((e) => {

      for (let i = 2; i < e; i++) {

        if (e % i === 0) {

          return false;

        }

      }

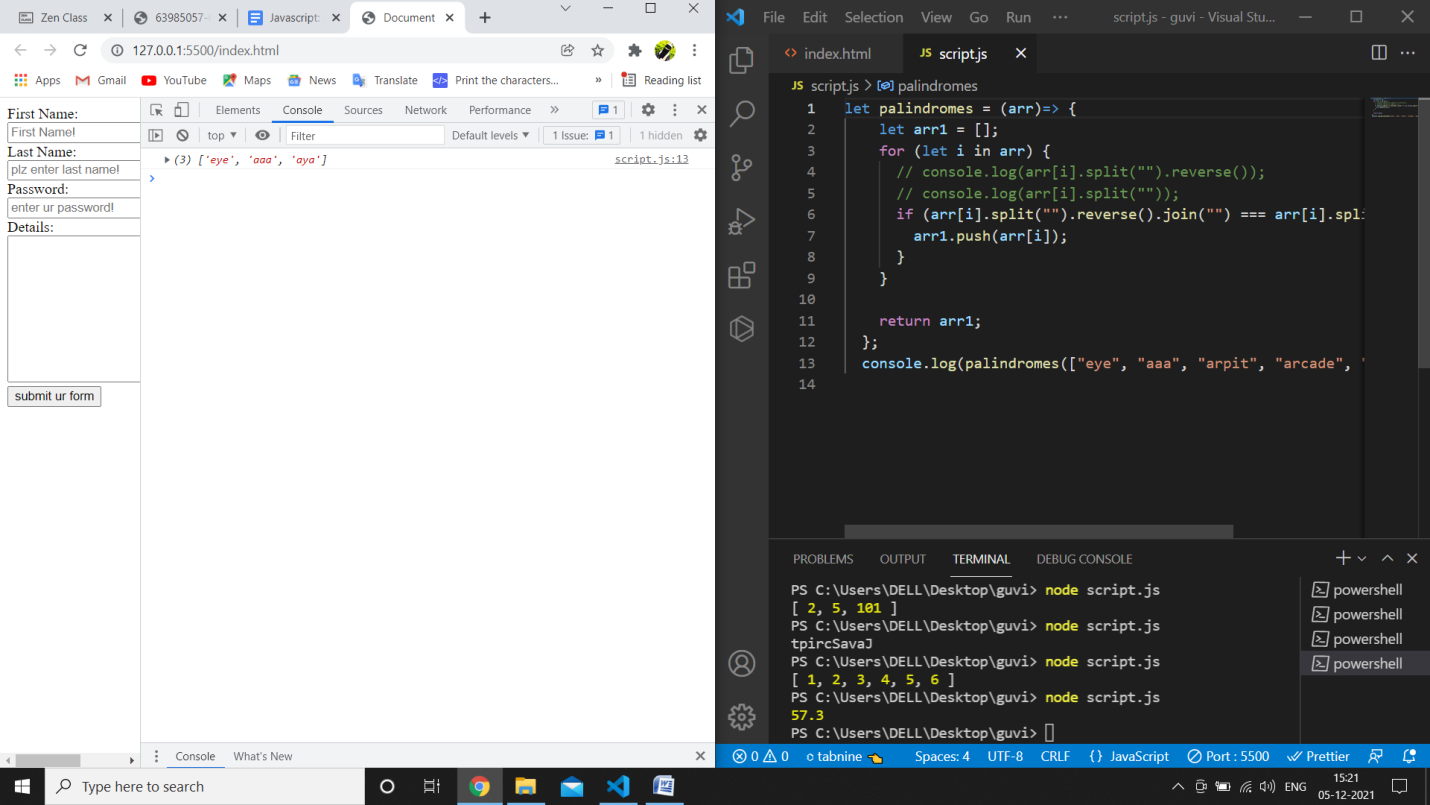
      return e > 1;

    });

  };

  console.log(primes([1, 2, 5, 16, 25, 99, 101]));

e.code:



-----------------------------------------------------------------------------------------------------------------