1. **Do the below programs in anonymous function & IIFE**

a. Print odd numbers in an array

//Print odd numbers in an array using anonymous function

var nums = [1,2,3,4,5,6,7,8,9,10];

//creating two empty arrays for odds and evens

var odds = [];

//var evens = [];

//creating a function with one variable i.e., nothing but anonymous function

var numbers = function(nums) {

//creating for loop for the given array of numbers

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

//giving the condition number divided by 2 if it is not equal to zero push odd else push even

if ((nums[i] % 2) !== 0) {

odds.push(nums[i]);

}

/\*else {

evens.push(nums[i]);

}\*/

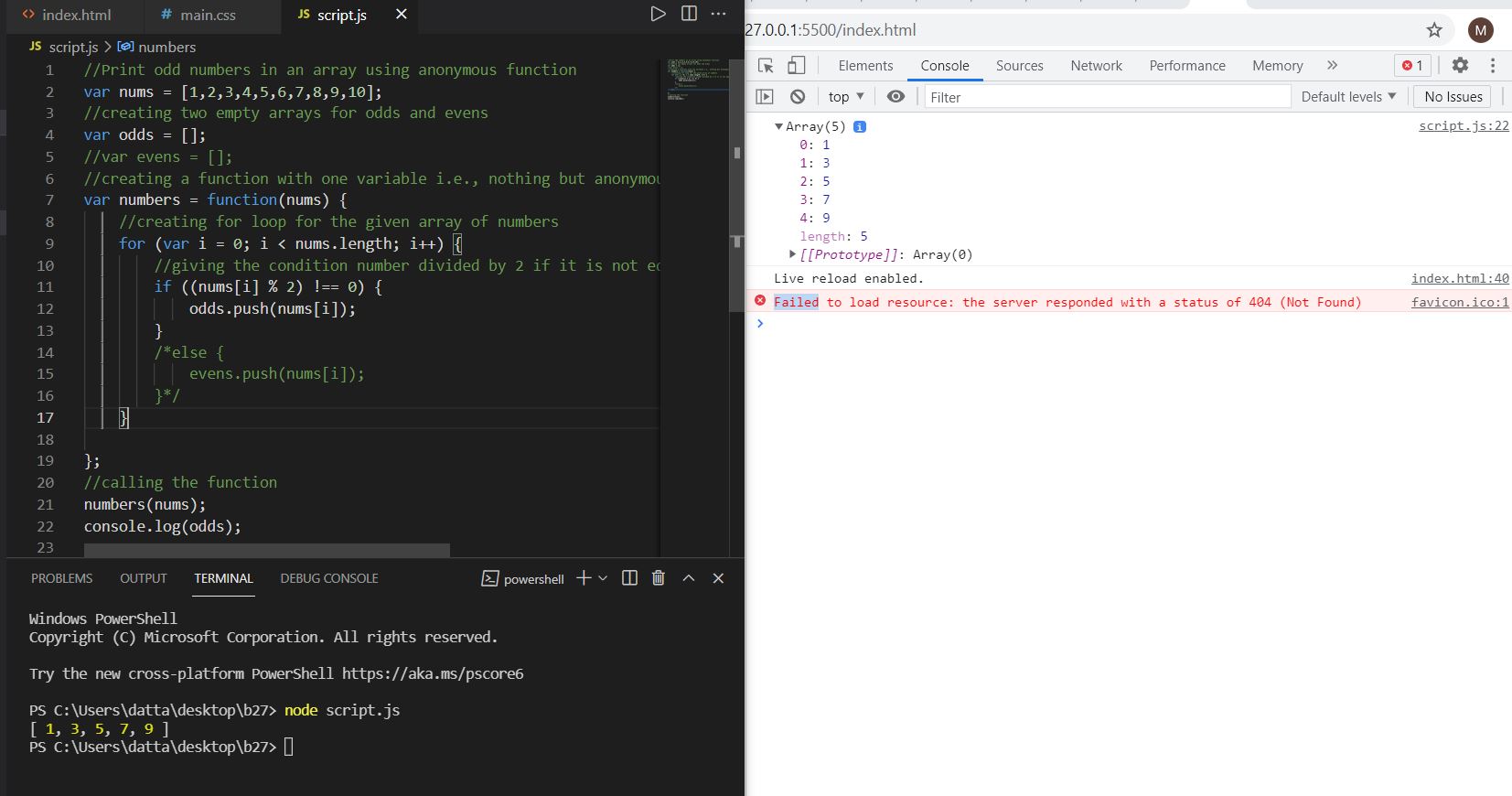
}

};

//calling the function

numbers(nums);

console.log(odds);



//Print odd numbers in an array using IIFE

//creating IIFE function and calling at the end of the directly

(function(nums) {

//creating two empty arrays for odds and evens

var odds = [];

//var evens = [];

//creating for loop for the given array of numbers

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

//giving the condition number divided by 2 if it is not equal to zero push odd else push even

if ((nums[i] % 2) !== 0) {

odds.push(nums[i]);

}

/\*else {

evens.push(nums[i]);

}\*/

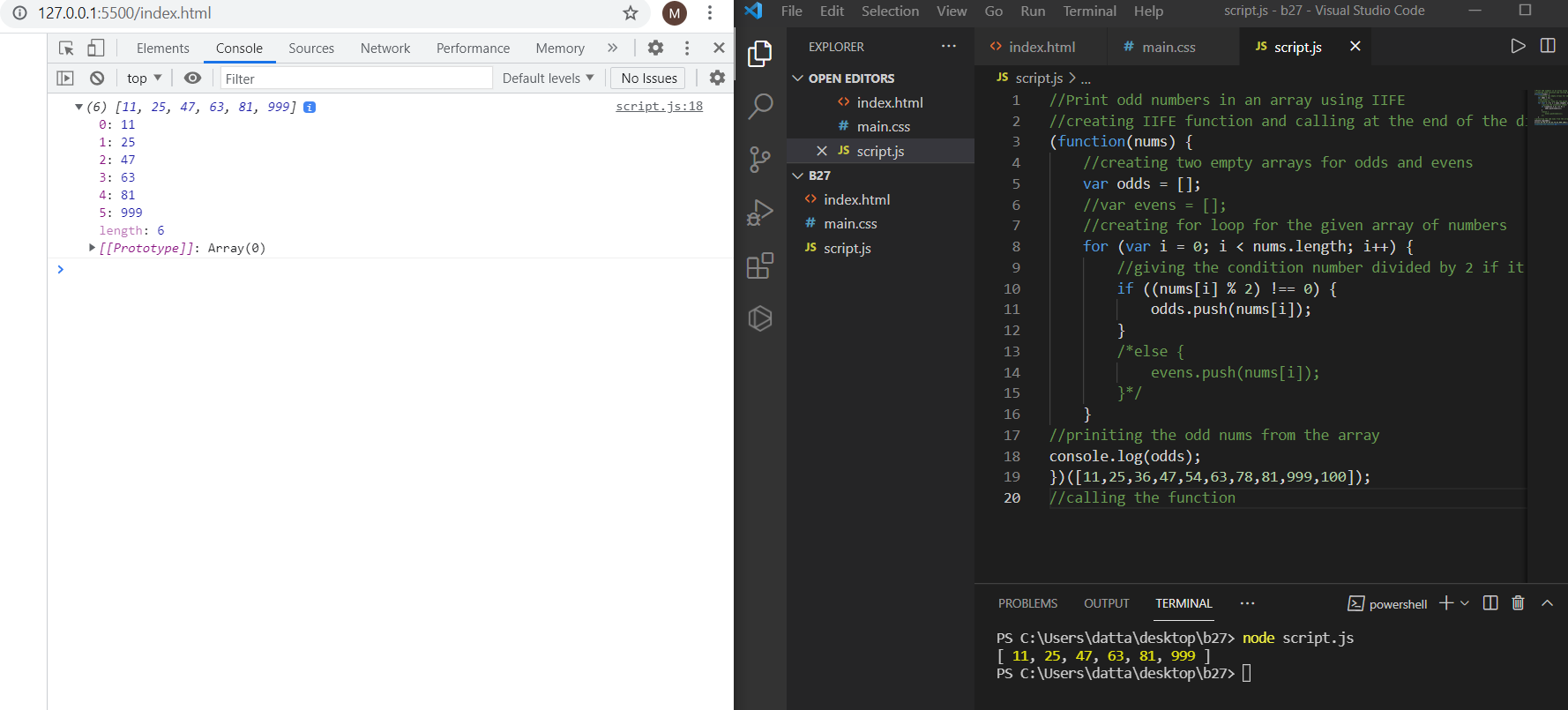
}

//priniting the odd nums from the array

console.log(odds);

})([11,25,36,47,54,63,78,81,999,100]);

//calling the function



b. Convert all the strings to title case in a string array

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

// defining the array of strings to

var arr = ["red", "blue", "black", "white", "pink"];

// creating the variable with name and defining the function in it

var TitleCase = function() {

// creating the for loop over the array and iterating

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

//for every iteration convert to title case and printing every string

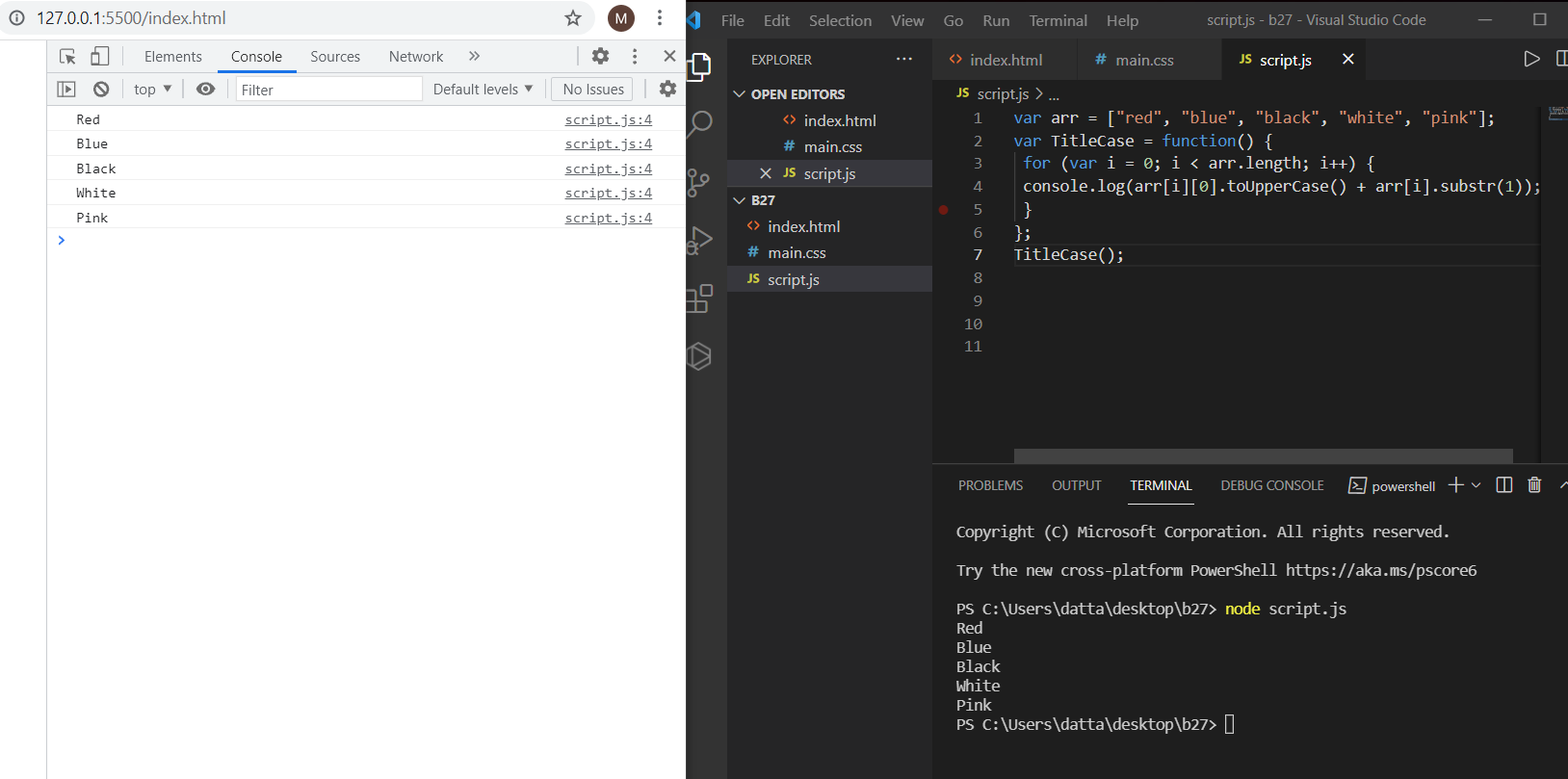
console.log(arr[i][0].toUpperCase() + arr[i].substr(1));

}

};

// calling the function

TitleCase();



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

// defining the array of strings to

var arr = ["red", "blue", "black", "white", "pink"];

// creating the variable with name and defining the function in it

(function() {

// creating the for loop over the array and iterating

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

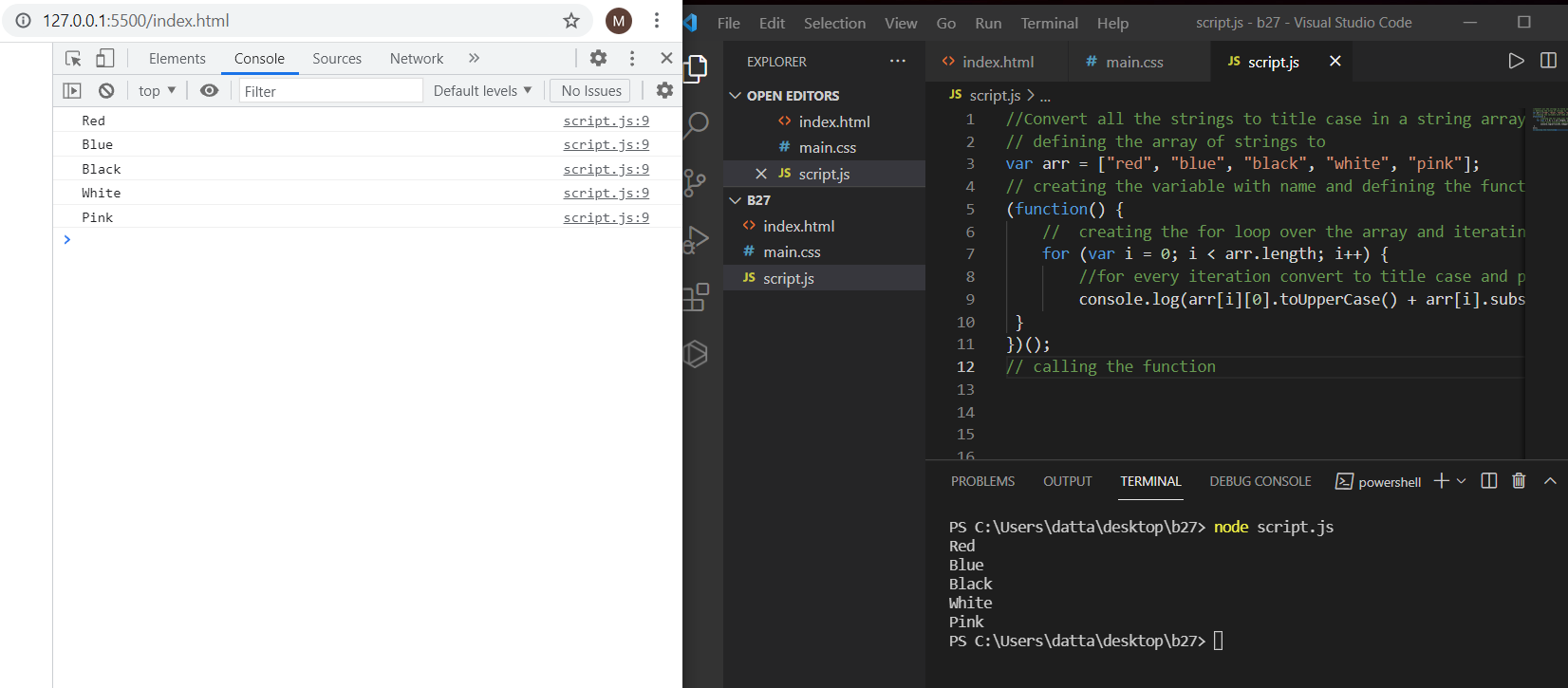
//for every iteration convert to title case and printing every string

console.log(arr[i][0].toUpperCase() + arr[i].substr(1));

}

})();

// calling the function



C. Sum of all numbers in an array

//Sum of all numbers in an array using anonymous function

var nums = [1,2,3,4,5,6,7,8,9,10];

//creating one variable with sum as 0

var sum = 0;

//creating a function with one variable i.e., nothing but anonymous function

var numbers = function(nums) {

//creating for loop for the given array of numbers to iterate all the numbers

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

//adding the values for every iteration and storing in sum variable

sum +=nums[i];

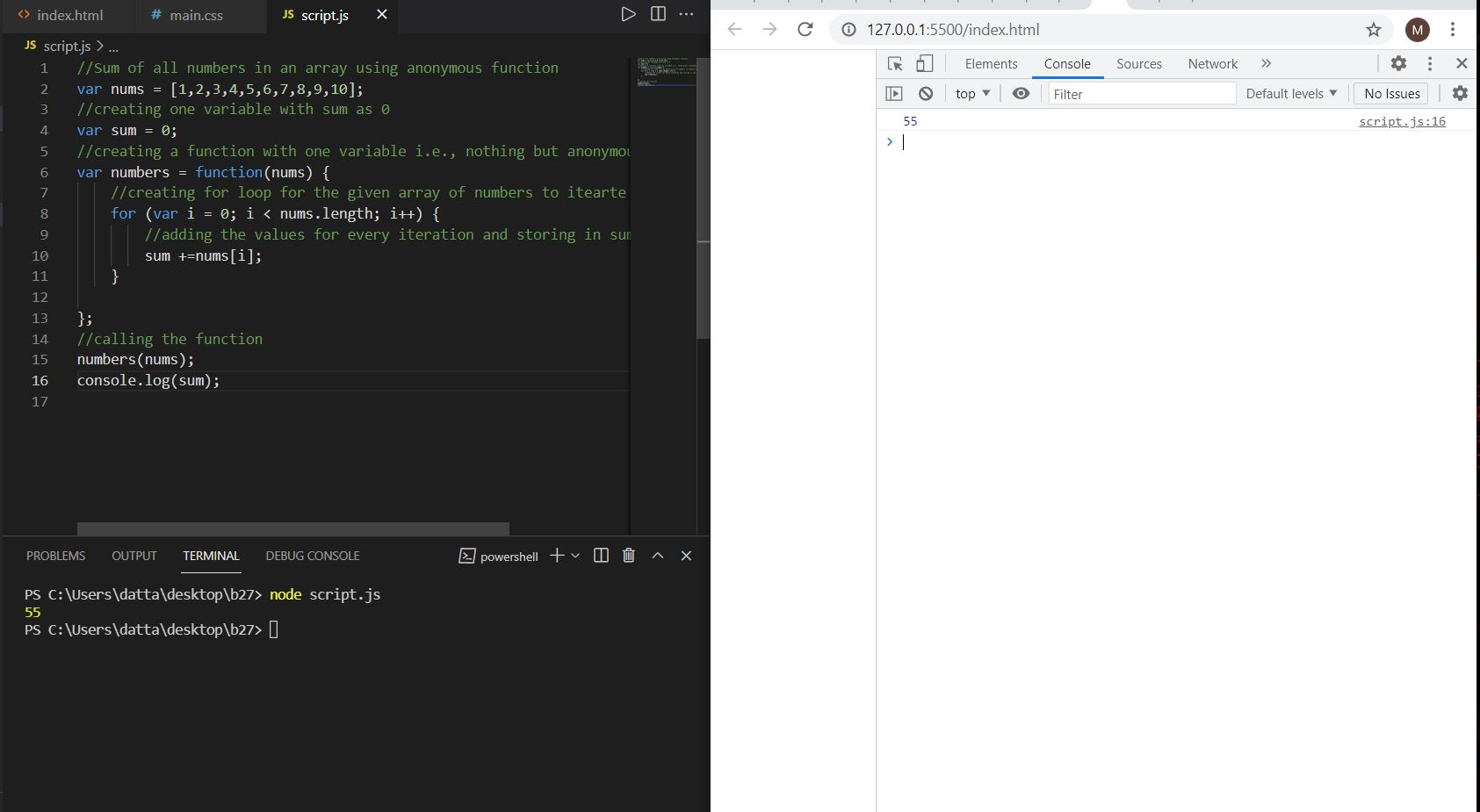
}

};

//calling the function

numbers(nums);

console.log(sum);



//Sum of all numbers in an array using IIFE

(function(nums) {

//creating one variable with sum as 0

var sum = 0;

//creating for loop for the given array of numbers to iterate all the numbers

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

//adding the values for every iteration and storing in sum variable

sum +=nums[i];

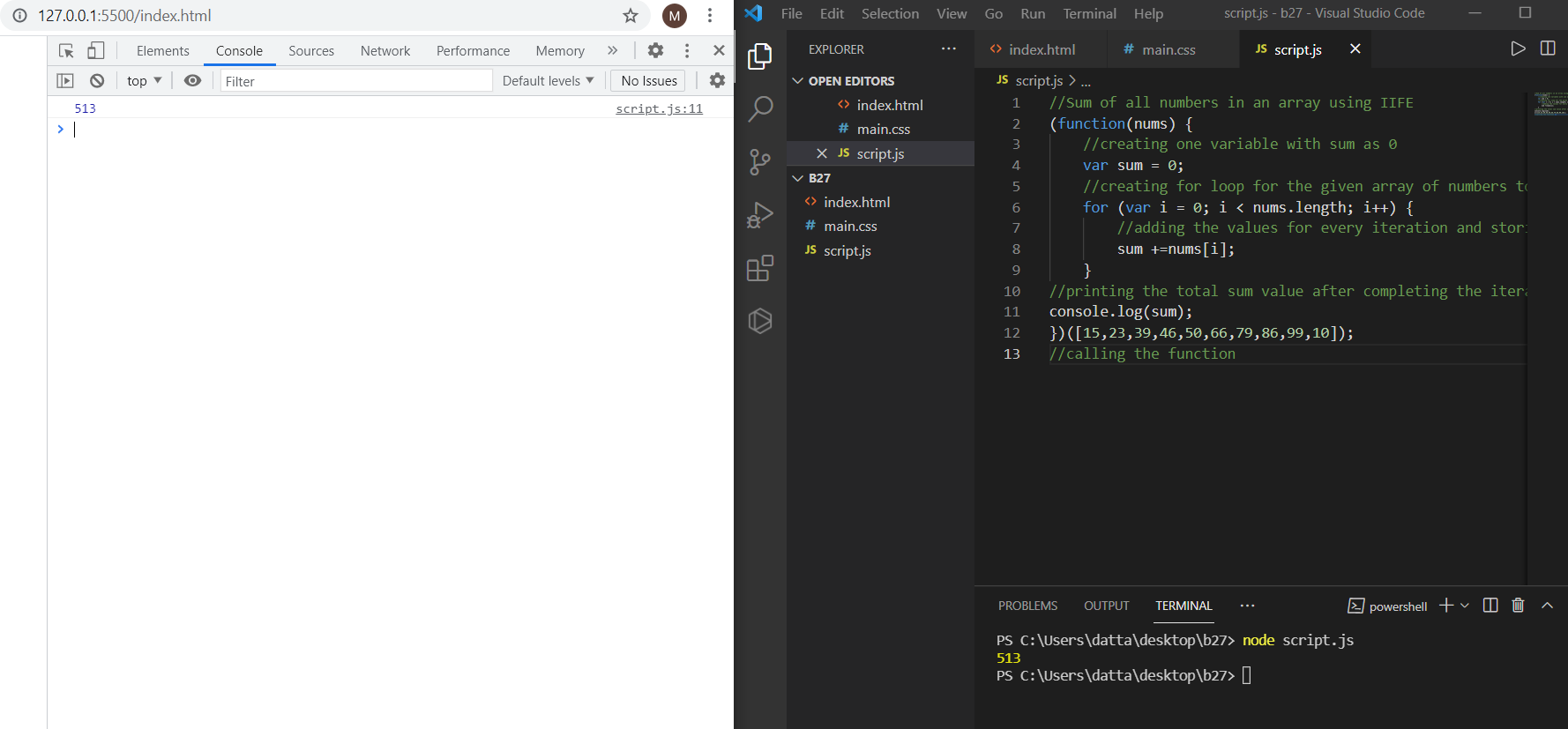
}

//printing the total sum value after completing the iteration over the array

console.log(sum);

})([15,23,39,46,50,66,79,86,99,10]);

//calling the function



d. Return all the prime numbers in an array

//Return all the prime numbers in an array using anonymous function

// creating one array of numbers with variable

const newArray = [1, 3, 2, 5, 10, 11, 13, 15, 16, 18];

//with one variable we are creating function and doing filter after iteration

const myPrimeArray = newArray.filter(function(num){

//creating for loop for checking array of numbers and iterating over it

for (let i = 2; i < num; i++) {

//if condition is equal to zero return false otherwise return true

if (num % i === 0) return false;

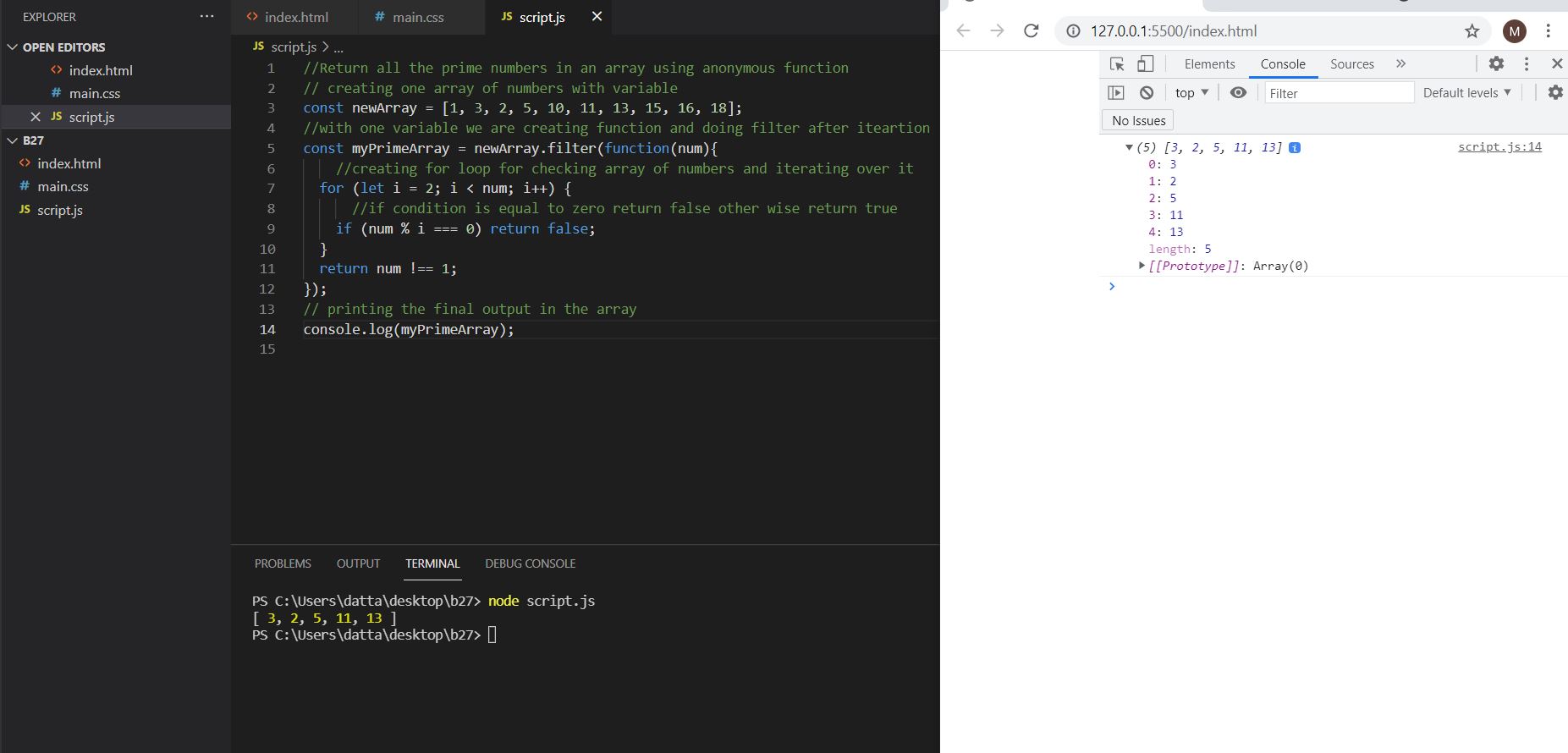
}

return num !== 1;

});

// printing the final output in the array

console.log(myPrimeArray);



//Return all the prime numbers in an array using IIFE

//creating the IIFE and defining the one arugument int it for filtering the array

(function prime(arr){

// for every iteration check the array numbers printing the numbers

console.log("The prime numbers are: " + arr.filter((number) => {

//Iterating over the loop and doing every number to the square root check the numbers

for (var i = 2; i <= Math.sqrt(number); i++) {

// if the number is divided by 2 is equal to zero return the false otherwise returns the true

if (number % i === 0) return false;

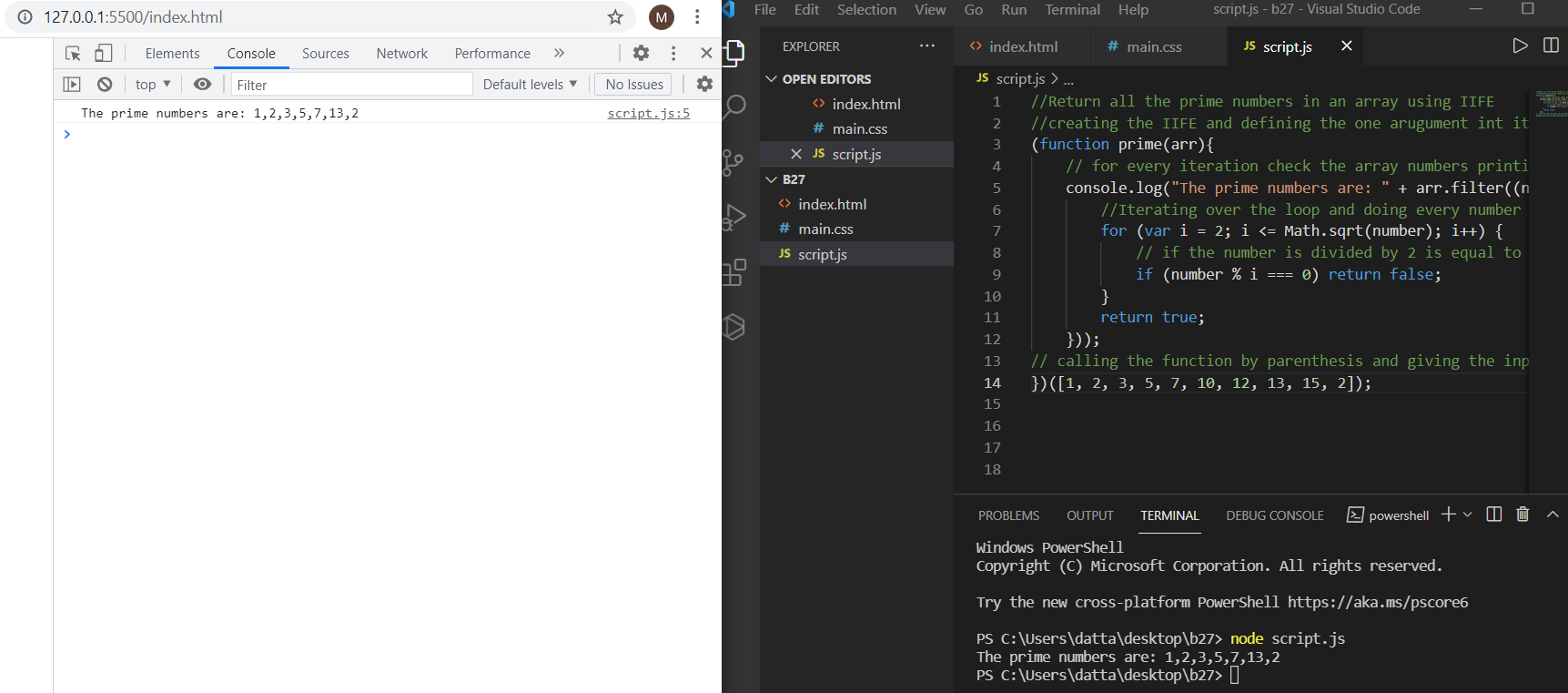
}

return true;

}));

// calling the function by parenthesis and giving the input array in it.

})([1, 2, 3, 5, 7, 10, 12, 13, 15, 2]);



e. Return all the palindromes in an array

//Return all the palindromes in an array using anonymous function

// defining one array with strings and numbers

const arr = ['carecar', 1344, 12321, 'did', 'cannot', 'track', 99, 143341];

// creating one arrow function for checking the array

const isPalindrome = function(el) {

// converting each number or word to string

const str = String(el);

let i = 0;

let j = str.length - 1;

// writing while loop for iterating over array

while(i < j) {

// writing if condition string of i is equal to string og j then if condition executes and return true otherwise return false

if(str[i] === str[j]) {

i++;

j--;

}

else {

return false;

}

}

return true;

};

// created another arrow function for filtering the array after checking the array

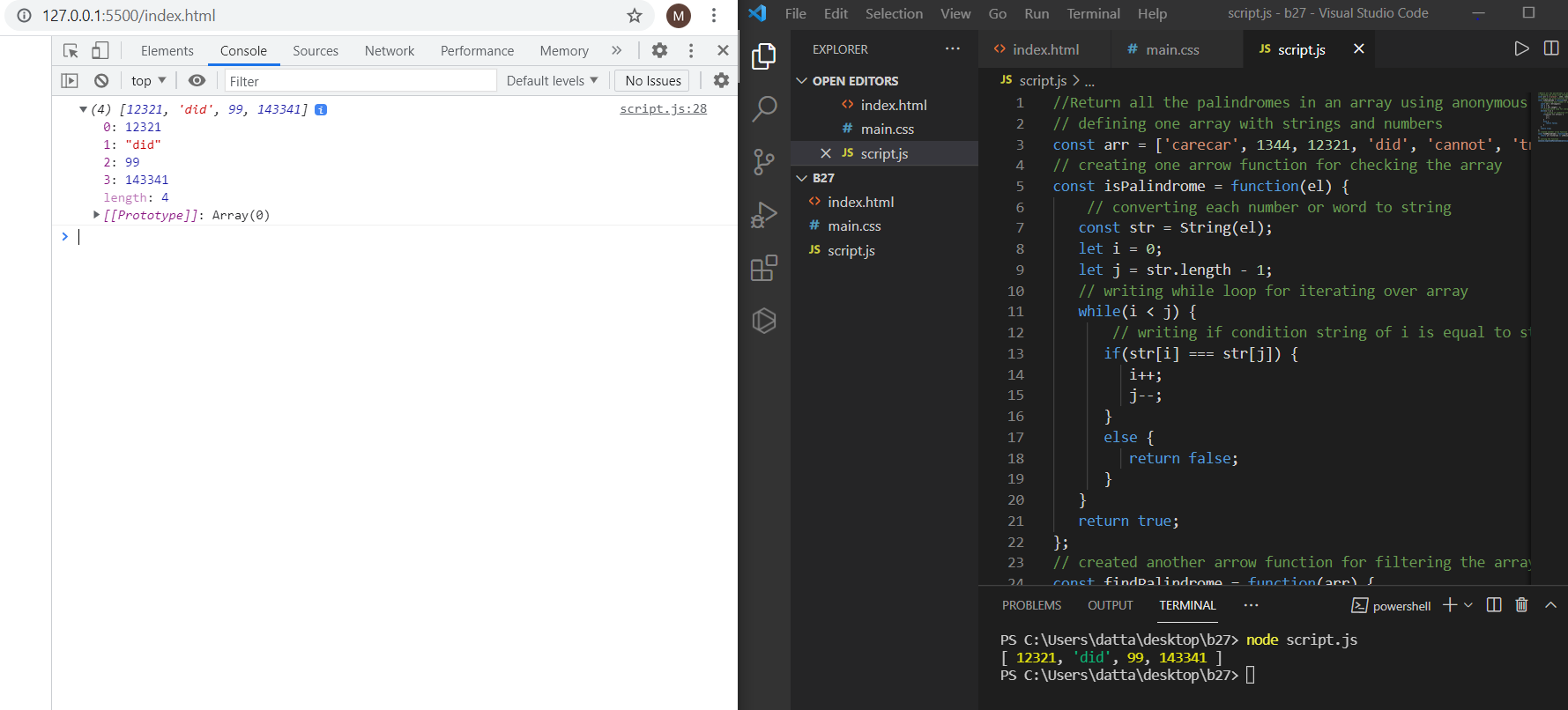
const findPalindrome = function(arr) {

return arr.filter(el => isPalindrome(el));

};

// calling the function

console.log(findPalindrome(arr));



//Return all the palindromes in an array using IIFE

// creating one arrow function for checking the array

(function(el) {

// for every iteration of checking it will check, filter the string and print it.

console.log("The palindromes are: " + el.filter(function(el) {

// converting each number or word to string

const str = String(el);

let i = 0;

let j = str.length - 1;

// writing while loop for iterating over array

while(i < j) {

// writing if condition string of i is equal to string og j then if condition executes and return true otherwise return false

if(str[i] === str[j]) {

// it will increment and decrement the i and j if the condition is true

i++;

j--;

}

else {

return false;

}

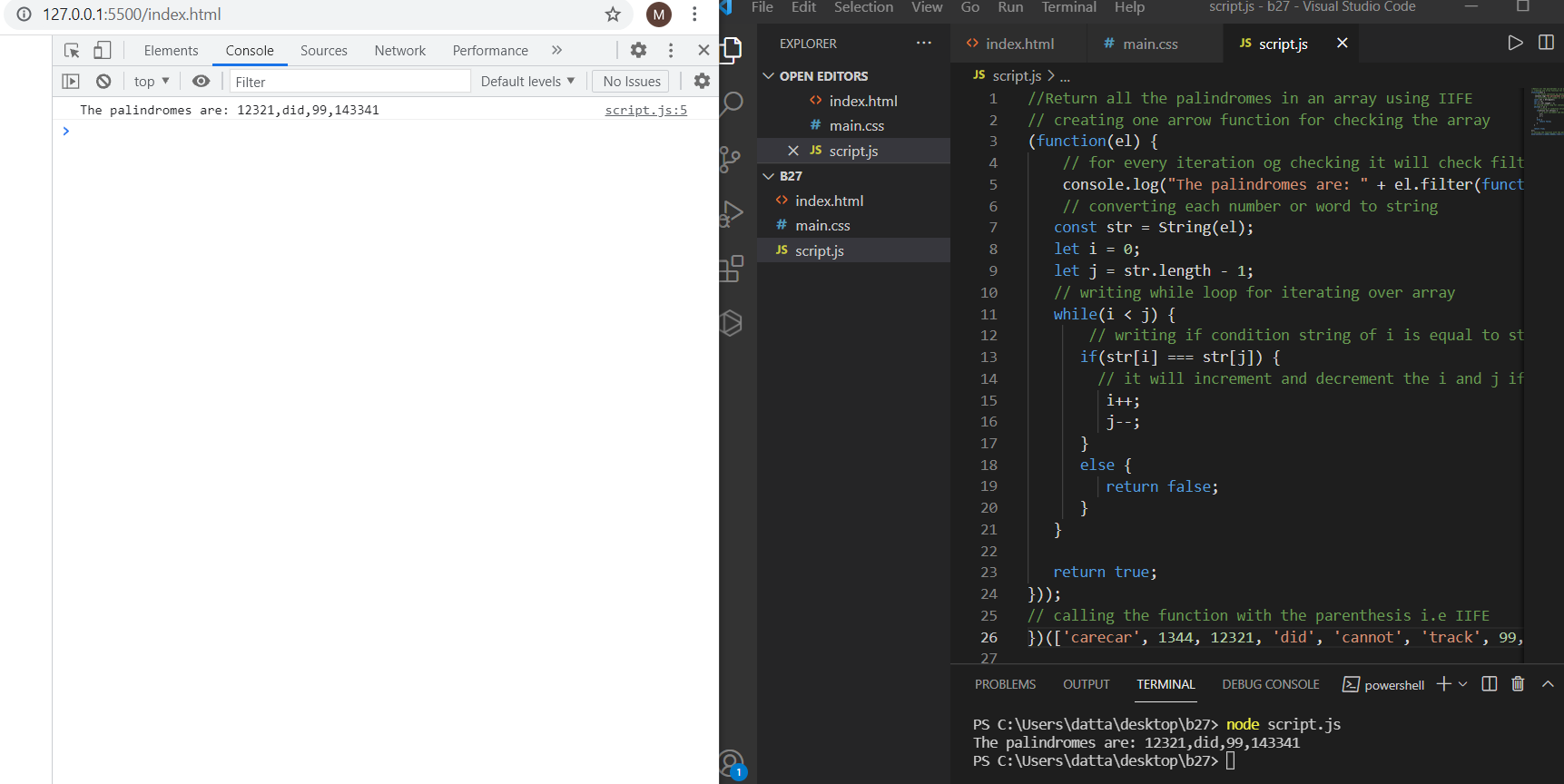
}

return true;

}));

// calling the function with the parenthesis i.e IIFE

})(['carecar', 1344, 12321, 'did', 'cannot', 'track', 99, 143341]);



f. Return median of two sorted arrays of same size

// A Simple Merge based O(n) solution to find median of two sorted arrays of same size

/\* This function returns median of ar1[] and ar2[].

Assumptions in this function:

Both ar1[] and ar2[] are sorted arrays

Both have n elements \*/

let sortedArray=function getMedian(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;

}

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

var ar1 = [1, 19, 25, 36, 43];

var ar2 = [2, 18, 23, 38, 55];

var n1 = ar1.length;

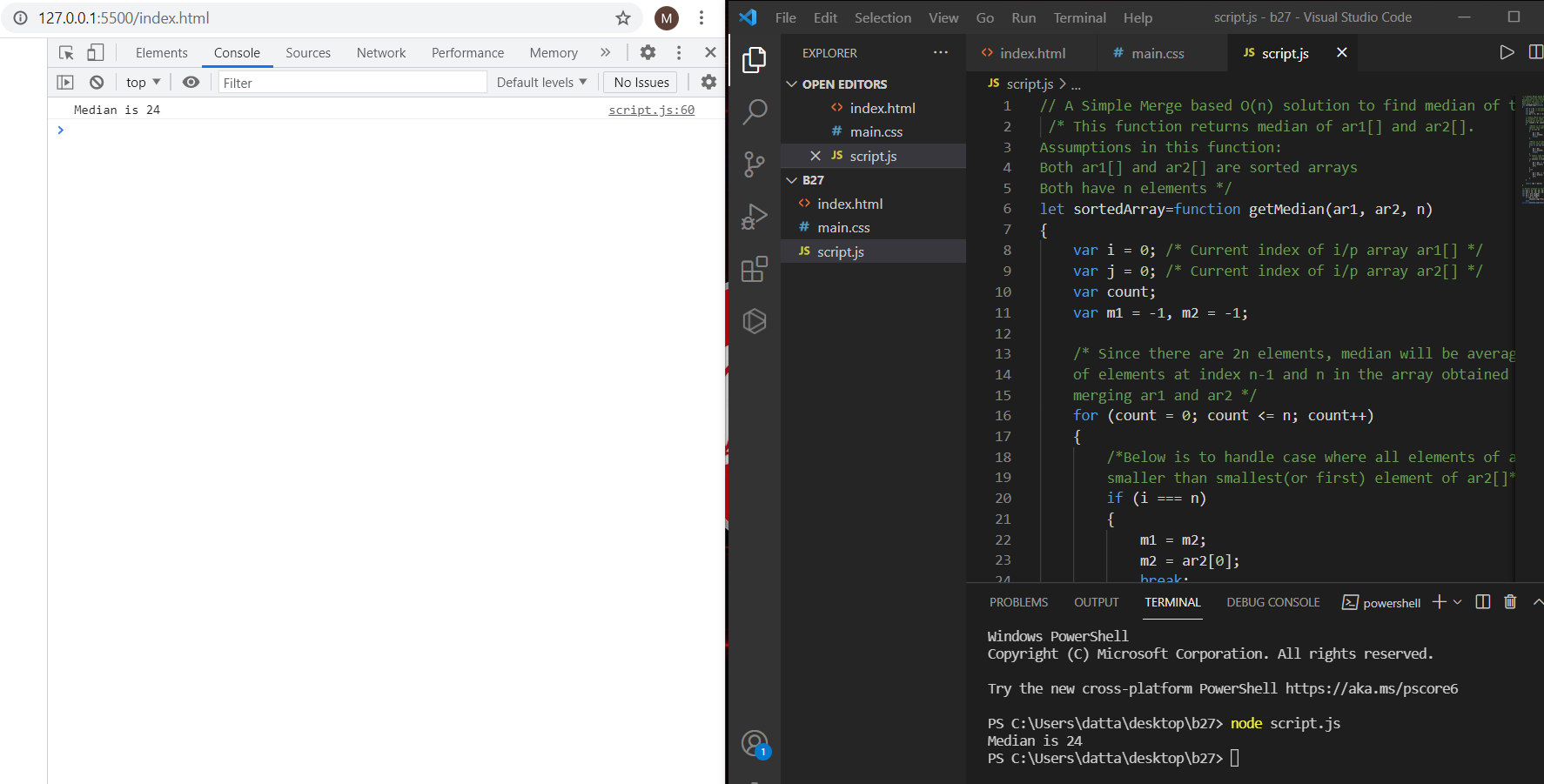
var n2 = ar2.length;

if (n1 === n2)

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

else

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



g. Remove duplicates from an array

//Remove duplicates from an array using anonymous function

//defining the one array with one variable

let chars = ['A', 'B', 'A', 'C', 'B', 'C', 'D', 'E'];

//with the filter and anonymous function we are removing duplicates in an array

//In the function we are defining two variables one is for holding the value and other for index number.

let uniqueChars = chars.filter(function(c, index){

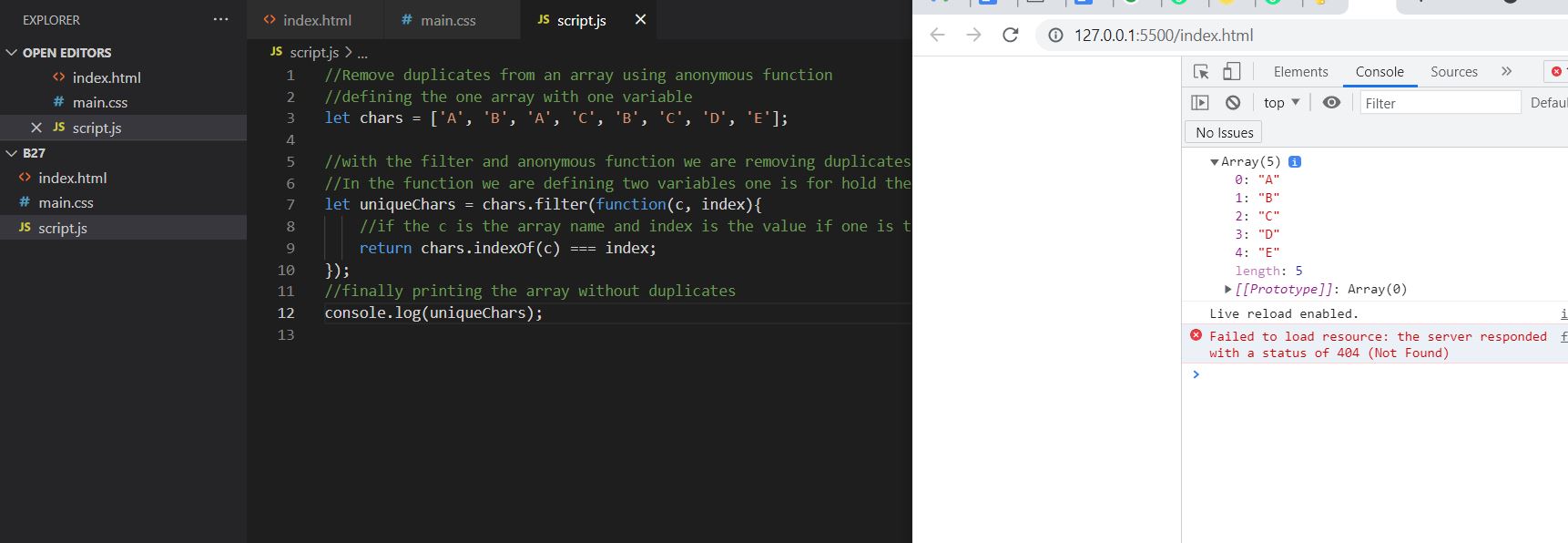
//if the c is the array name and index is the value if one is their it will come true otherwise it will come false

return chars.indexOf(c) === index;

});

//finally printing the array without duplicates

console.log(uniqueChars);



//Removing all duplicates from an Array using anonymous function

//Defining the function with one variable

var colors = function(arr){

var newArr = [];

//creating for loop for the array length

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

// if the same word is came it will not push the string to the array otherwise it will push the each string to the newarray

if(newArr.indexOf(arr[i]) === -1) {

newArr.push(arr[i]);

}

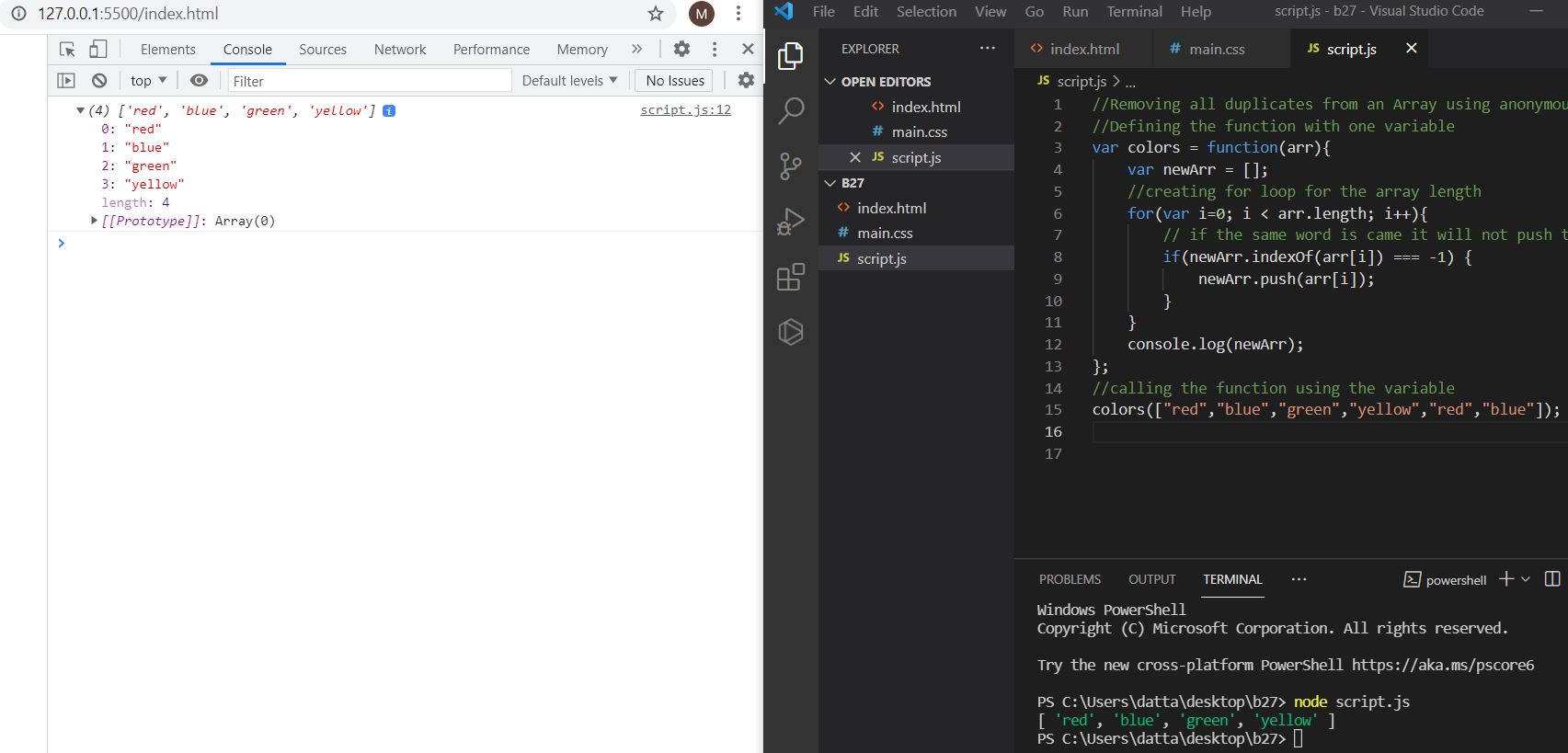
}

console.log(newArr);

};

//calling the function using the variable

colors(["red","blue","green","yellow","red","blue"]);



//Removing all duplicates from an Array using IIFE

//Defining the array with one variable

var colors = ["red","blue","green","yellow","red","blue"];

//defining the IIFE

(function(){

var newArr = [];

//creating for loop for the array length

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

// if the same word is came it will not push the string to the array otherwise it will push the each string to the newarray

if(newArr.indexOf(colors[i]) === -1) {

newArr.push(colors[i]);

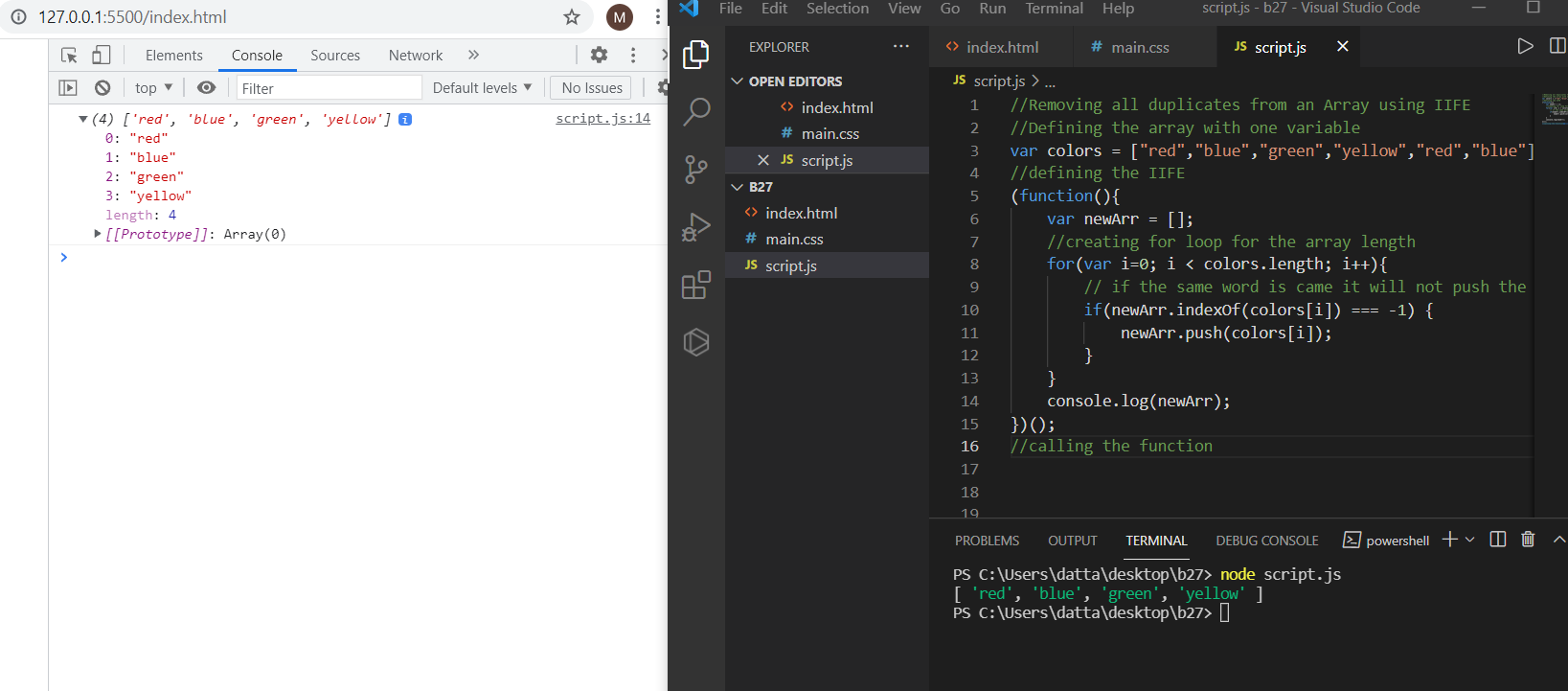
}

}

console.log(newArr);

})();

//calling the function by parenthesis



**h. Rotate an array by k times**

//Rotate an array by k times using anonymous function

// Creating one variable and defining the array

let arr = [4, 8, 10, 6, 11, 6, 1, 9, 10, 12, 13];

// defining the k value with one variable

let k = 7;

// by using the array length counting the array with the k value it is doing division and giving remainder

k = arr.length % k;

// defining the the function

let karray = function(num) {

// Iterating over the array and taking the values

for (let i = 0; i < k + 1; i++) {

// by using unshift it will do right rotation and by using pop it will remove las adding value

arr.unshift(arr.pop());

}

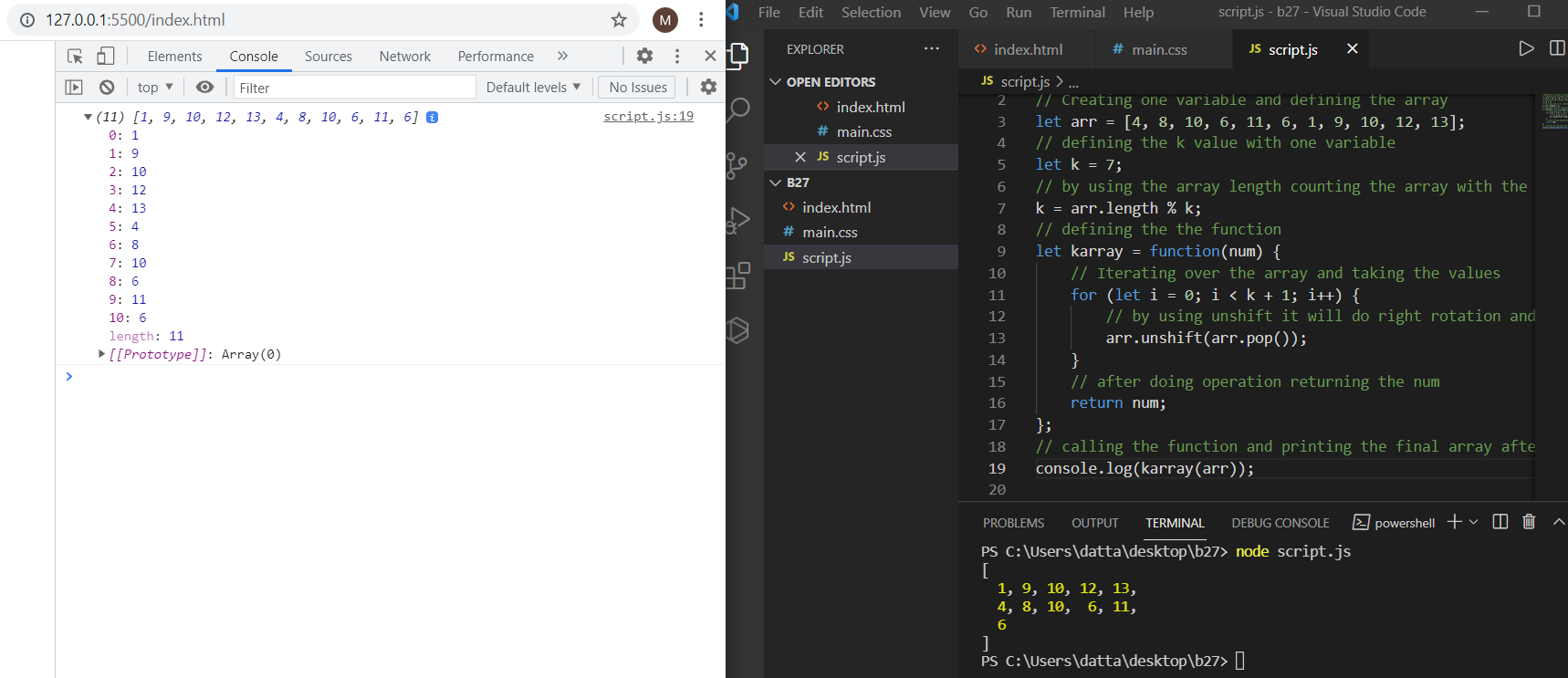
// after doing operation returning the num

return num;

};

// calling the function and printing the final array after k times rotation

console.log(karray(arr));

****

//Rotate an array by k times using IIFE

// Creating one variable and defining the array

let arr = [4, 8, 10, 6, 11, 6, 1, 9, 10, 12, 13];

// defining the k value with one variable

let k = 7;

// by using the array length counting the array with the k value it is doing division and giving remainder

k = arr.length % k;

// defining the the function

(function(num) {

// Iterating over the array and taking the values

for (let i = 0; i < k + 1; i++) {

// by using unshift it will do right rotation and by using pop it will remove las adding value

arr.unshift(arr.pop());

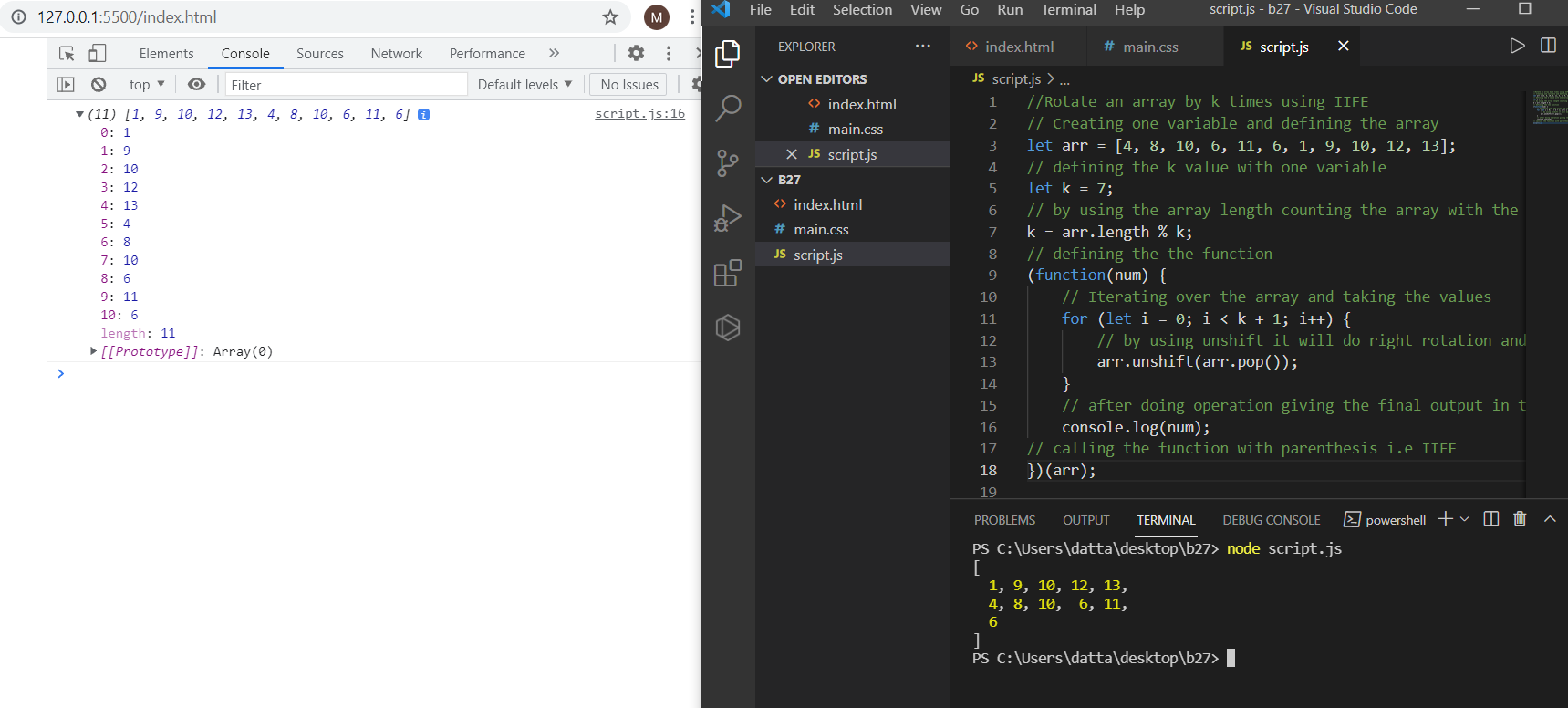
}

// after doing operation giving the final output in the print format

console.log(num);

// calling the function with parenthesis i.e IIFE

})(arr);

****