1. Write a function that prints the numbers from 1 to 100. But for multiples of three, print "Fizz" instead of the number, and for the multiples of five, print "Buzz". For numbers that are multiples of both three and five, print "FizzBuzz".

Code:

function fizzBuzz(num){

console.log(num);

for (var i = 0; i <= 100; i++){

if (i % 5 === 0 && i % 3 === 0){

console.log(‘fizzbuzz’);

} else if (i % 5 === 0) {

console.log(‘buzz’);

} else if (i % 3 === 0){

console.log(‘fizz’);

} else {

console.log(i);

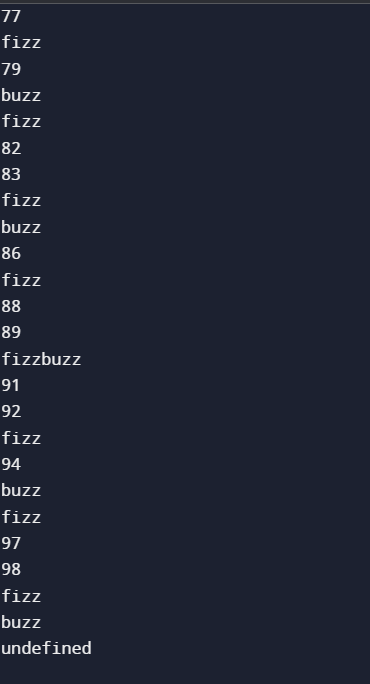
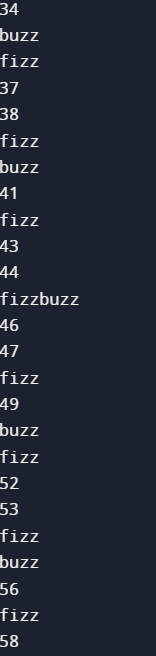
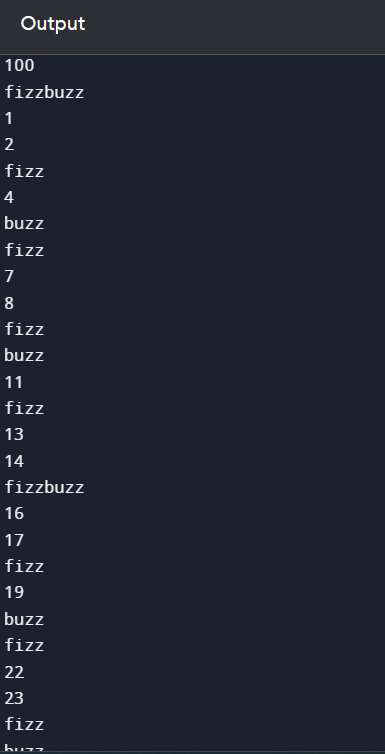
}

}

}

console.log(fizzBuzz(100));

output:



1. Write a function that takes a string input representing a simple arithmetic expression (only addition and subtraction) and returns the result.

Code:

const operator = prompt('Enter operator ( either + or - ): ');

const number1 = parseFloat(prompt('Enter first number: '));

const number2 = parseFloat(prompt('Enter second number: '));

let result;

if (operator == '+') {

result = number1 + number2;

}

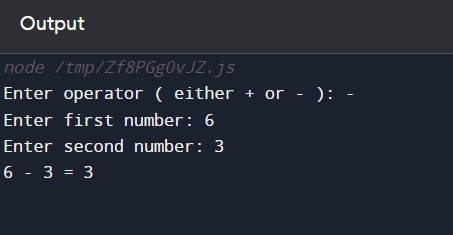
else (operator == '-') {

result = number1 - number2;

}

console.log(`${number1} ${operator} ${number2} = ${result}`);

OUTPUT:



1. Write a function that takes a nested array and returns a flattened array.

Code:

function flattenArray(arr) {

let result = [];

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

if (Array.isArray(arr[i])) {

result = result

.concat(flattenArray(arr[i]));

} else {

result

.push(arr[i]);

}

}

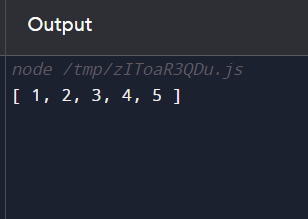
return result;

}

const nestedArray = [[1, 2], [3, [4, 5]]];

console.log(flattenArray(nestedArray));

OUTPUT:



1. Write a function that checks if two given strings are anagrams of each other.

Code:



OUTPUT:



1. Write a function that takes an array and returns a new array with duplicates removed.

Code:

function removeDuplicates(arr, n){

if (n==0 || n==1)

return n;

var temp = new Array(n);

var j = 0;

for (var i=0; i<n-1; i++)

if (arr[i] != arr[i+1])

temp[j++] = arr[i];

temp[j++] = arr[n-1];

for (var i=0; i<j; i++)

arr[i] = temp[i];

return j;

}

var arr = [1, 2, 2, 3, 4, 4, 4, 5, 5];

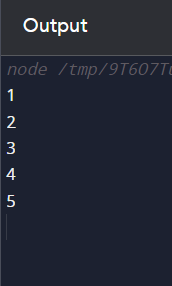
var n = arr.length;

n = removeDuplicates(arr, n);

for (var i=0; i<n; i++)

document.write( arr[i]+" ");

OUTPUT:



6.Write a function that takes a string and capitalizes the first letter of each word in the string.

Code:

function capital\_letter(str){

str = str.split(" ");

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

str[i] = str[i][0].toUpperCase() + str[i].substr(1)

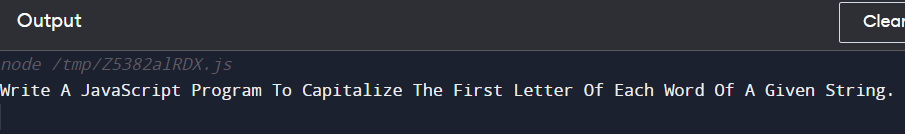
}

return str.join(" ");

}

console.log(capital\_letter("Write a JavaScript program to capitalize the first letter of each word of a given string."));

OUTPUT:



7. Write a function that generates the first n numbers of the Fibonacci sequence.

Code:

const number = parseInt(prompt('Enter the number of terms: '));

let n1 = 0, n2 = 1, nextTerm;

console.log('Fibonacci Series:');

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

console.log(n1);

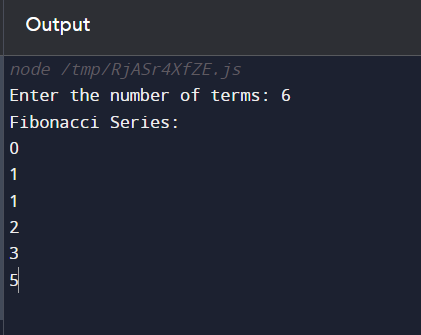
nextTerm = n1 + n2;

n1 = n2;

n2 = nextTerm;

}

OUTPUT:



8.Implement a simple HashMap class with put, get, and remove methods.

Code:

class HashTable {

constructor() {

this.table = new Array(127);

this.size = 0;

}

\_hash(key) {

let hash = 0;

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

hash += key.charCodeAt(i);

}

return hash % this.table.length;

}

set(key, value) {

const index = this.\_hash(key);

if (this.table[index]) {

for (let i = 0; i < this.table[index].length; i++) {

if (this.table[index][i][0] === key) {

this.table[index][i][1] = value;

return;

}

}

this.table[index].push([key, value]);

} else {

this.table[index] = [];

this.table[index].push([key, value]);

}

this.size++;

}

get(key) {

const index = this.\_hash(key);

if (this.table[index]) {

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

if (this.table[index][i][0] === key) {

return this.table[index][i][1];

}

}

}

return undefined;

}

remove(key) {

const index = this.\_hash(key);

if (this.table[index] && this.table[index].length) {

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

if (this.table[index][i][0] === key) {

this.table[index].splice(i, 1);

this.size--;

return true;

}

}

} else {

return false;

}

}

display() {

this.table.forEach((values, index) => {

const chainedValues = values.map(

([key, value]) => `[ ${key}: ${value} ]`

);

console.log(`${index}: ${chainedValues}`);

});

}

}

9.Write a function that filters out even numbers from an array.

Code:

let numbers = [8, 49, 16, 21];

let even = [];

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

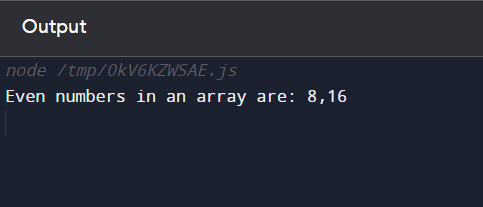
if (numbers[i] % 2 == 0)

even.push(numbers[i]);

}

console.log(`Even numbers in an array are: ${even}`);

OUTPUT:



10.Write a function that converts a given string to title case (capitalizing the first letter of each word).

Code:

function sentenceCase(str) {

if ((str === null) || (str === ''))

return false;

else

str = str.toString();

return str.replace(/\w\S\*/g,

function (txt) {

return txt.charAt(0).toUpperCase() +

txt.substr(1).toLowerCase();

});

}

console.log(sentenceCase('faseehul kalam'));

Output:

