**GLOBAL\_TREND (ASSESSMENT)**

Q1. Write a function \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_”FizzBuzz”.

**Code:**

function show(){

    for (let i = 0; i < 101; i++) {

        if(i%3==0 && i%5==0){

            console.log("FizzBuzz")

        }

        else if(i%3==0){

            console.log("Fizz")

        }

        else if(i%5==0){

            console.log("Buzz")

        }

        else{

            console.log(i)

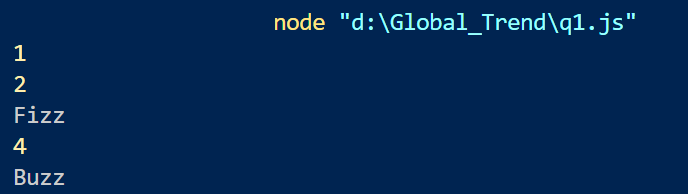
        }

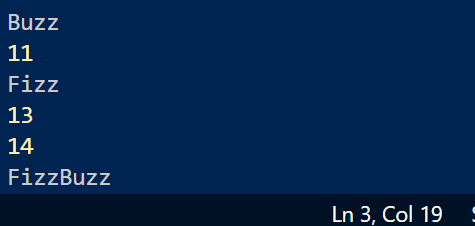
    }

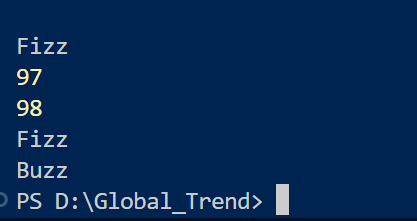
}

show()

**output:**

****

****

****

PS D:\Global\_Trend> nod

node "d:\Global\_Trend\q1.js"

1

2

Fizz

4

Buzz

Fizz

7

8

Fizz

Buzz

11

Fizz

13

14

FizzBuzz

16

17

Fizz

19

Buzz

Fizz

22

23

Fizz

Buzz

26

Fizz

28

29

FizzBuzz

31

32

Fizz

34

Buzz

Fizz

37

38

Fizz

Buzz

41

Fizz

43

44

FizzBuzz

46

47

Fizz

49

Buzz

Fizz

52

53

Fizz

Buzz

56

Fizz

58

59

FizzBuzz

61

62

Fizz

64

Buzz

Fizz

67

68

Fizz

Buzz

71

Fizz

73

74

FizzBuzz

76

77

Fizz

79

Buzz

Fizz

82

83

Fizz

Buzz

86

Fizz

88

89

FizzBuzz

91

92

Fizz

94

Buzz

Fizz

97

98

Fizz

Buzz // Sorry I can’t print screenshort of output because its too big.

Q2.Write a function that take a srtring\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_return the result.

**Code:**

function int(take){

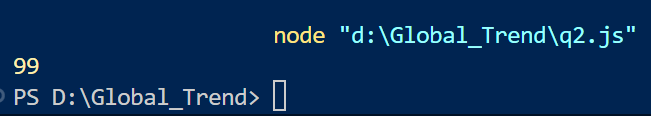
    return eval(take)

}

let str = "67+32"

console.log(int(str))

**Output:**

****

Q3. Write a \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_flattened array.

**Code:**

function flattenArray(nestedArray) {

    let result = [];

    function flatten(arr) {

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

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

                flatten(arr[i]);

            } else {

                result.push(arr[i]);

            }

        }

    }

    flatten(nestedArray);

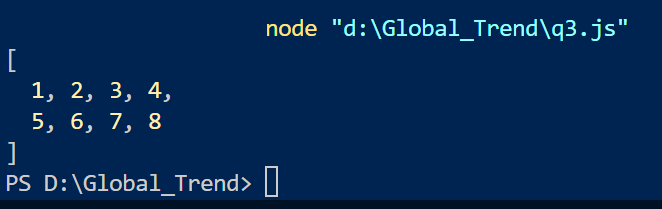
    return result;

}

let nestedArray = [1, [2, [3, 4], 5], 6, [7, 8]];

console.log(flattenArray(nestedArray));

**Output:**

****

Q4. Write a function \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_anagram of each other.

**Code:**

function isAnagram(str1, str2) {

    if (str1.length !== str2.length) {

        return false;

    }

    const sortedStr1 = str1.split('').sort().join('');

    const sortedStr2 = str2.split('').sort().join('');

    return sortedStr1 === sortedStr2;

}

let str1 = "listen";

let str2 = "silent";

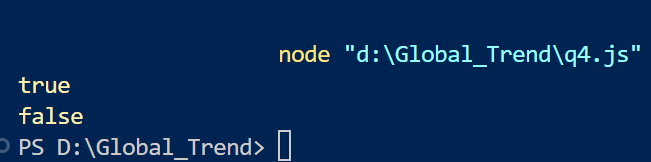
console.log(isAnagram(str1, str2));

str1 = "hello";

str2 = "world";

console.log(isAnagram(str1, str2));

**Output:**

****

Q5.Write a function \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_duplicates removed.

**Code:**

function duplicates(arr){

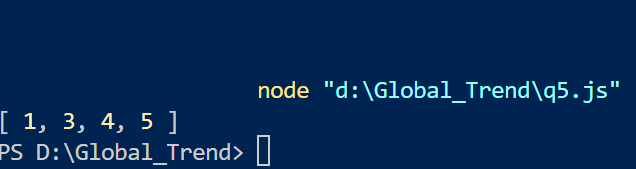
   return[...new Set(arr)]

}

arr= [1,1,1,3,4,5,5]

console.log(duplicates(arr))

**Output:**

****

Q6. Write a function \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_capitalized first word.

**Code:**

function capitalizeFirstWord(str) {

    if (!str) return str;

    const words = str.split(' ');

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

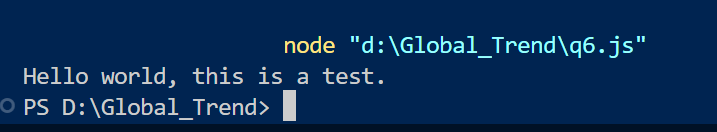
    return words.join(' ');

}

let str = "hello world, this is a test.";

console.log(capitalizeFirstWord(str));

**Output:**

****

Q7 Write a function to \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ n number.

**Code:**

function fibo(n){

    let n1=0

    let n2=1

    let next=0

    for(i=1 ; i<=n ;i++){

        console.log(n1)

        next = n1+n2

        n1=n2

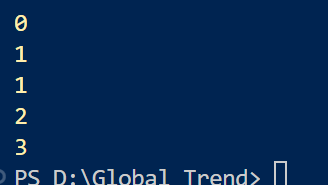
        n2 = next

    }

}

fibo(5)

**Output:**

****

Q8.Impliment a simple hashmap \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_methords.

**Code:**

class HashMap {

    constructor() {

        this.map = {};

    }

    put(key, value) {

        this.map[key] = value;

    }

    get(key) {

        if (key in this.map) {

            return this.map[key];

        } else {

            return null;

        }

    }

    remove(key) {

        if (key in this.map) {

            delete this.map[key];

        }

    }

}

let myMap = new HashMap();

myMap.put("name", "Alice");

myMap.put("age", 25);

console.log(myMap.get("name"));

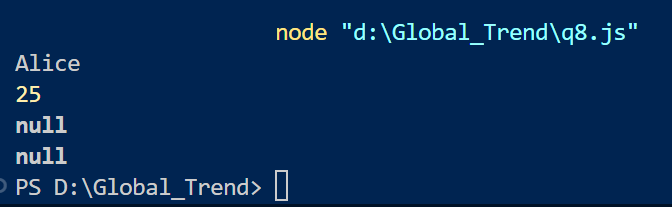
console.log(myMap.get("age"));

console.log(myMap.get("address"));

myMap.remove("age");

console.log(myMap.get("age"));

**Output:**

****

Q9. Write a function \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ array.

**Code:**

function filter(arr) {

    let res = [];

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

        if (arr[i] % 2 === 0) {

            res.push(arr[i]);

        }

    }

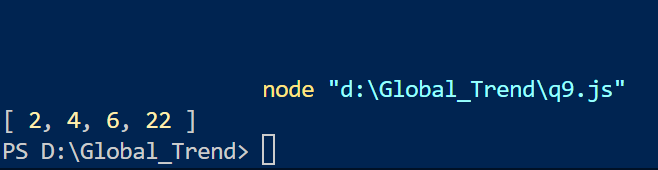
    return res;

}

let arr= [11,2,4,6,7,9,22]

console.log(filter(arr))

**Output:**

****

Q10. Write a function \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ string to title case.

**Code:**

function toTitleCase(str) {

    return str.split(' ').map(word => {

        return word.charAt(0).toUpperCase() + word.slice(1).toLowerCase();

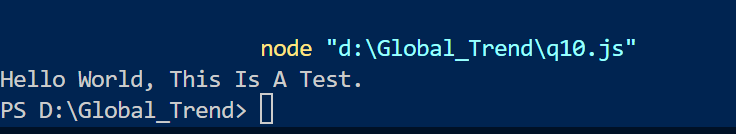
    }).join(' ');

}

let str = "hello world, this is a test.";

console.log(toTitleCase(str));

**Output:**

****