

## DAY - 7

### Exercises: Level 1

1. Declare a function *fullName* and it print out your full name.

```
const fullName = () => 'Priyanka Saini';  
console.log(fullName());
```

2. Declare a function *fullName* and now it takes *firstName*, *lastName* as a parameter and it returns your full - name.

```
const fullName = (firstName, lastName) => firstName + " " +  
lastName;  
console.log(fullName('Priyanka', 'Saini'));
```

3. Declare a function *addNumbers* and it takes two two parameters and it returns sum.

```
const addNumber = (num1, num2) => num1+num2;  
addNumber(78, 90);
```

4. An area of a rectangle is calculated as follows:  $area = length \times width$ . Write a function which calculates *areaOfRectangle*.

```
const areaOfRectangle = (length, width) => length * width;  
areaOfRectangle(20, 30);
```

5. A perimeter of a rectangle is calculated as follows:  $perimeter = 2 \times (length + width)$ . Write a function which calculates *perimeterOfRectangle*.

```
const perimeterOfRectangle = (length, width) => 2*(length+width);  
perimeterOfRectangle(20, 30);
```

6. A volume of a rectangular prism is calculated as follows:  $volume = length \times width \times height$ . Write a function which calculates *volumeOfRectPrism*.

```
const volumeOfRectPrism = (length, width, height) =>  
length*width*height;  
volumeOfRectPrism(20,30,40);
```

7. Area of a circle is calculated as follows:  $area = \pi \times r \times r$ . Write a function which calculates *areaOfCircle*

```
const areaOfCircle = (radius) => Math.PI * radius * radius;  
areaOfCircle(60);
```

8. Circumference of a circle is calculated as follows:  $circumference = 2\pi r$ . Write a function which calculates *circumOfCircle*

```
const circumOfCircle = (radius) => Math.PI * 2 * radius;  
circumOfCircle(60);
```

9. Density of a substance is calculated as follows:  $density = mass/volume$ . Write a function which calculates *density*.

```
const density = (mass, volume) => mass/volume;
density(60, 20);
```

10. Speed is calculated by dividing the total distance covered by a moving object divided by the total amount of time taken. Write a function which calculates a speed of a moving object, *speed*.

```
const speed = (distance, time) => distance/time;
speed(60, 20);
```

11. Weight of a substance is calculated as follows:  $weight = mass \times gravity$ . Write a function which calculates *weight*.

```
const weight = (mass, gravity = 9.81) => mass*gravity + ' N';
weight(100);
weight(100, 1.62);
```

12. Temperature in oC can be converted to oF using this formula:  $oF = (oC \times 9/5) + 32$ . Write a function which convert oC to oF *convertCelsiusToFahrenheit*.

```
const convertCelsiusToFahrenheit = (tempInCelsius) =>
(tempInCelsius * (9/5)) + 32;
convertCelsiusToFahrenheit(100);
```

13. Body mass index(BMI) is calculated as follows:  $bmi = weight \text{ in Kg} / (height \times height) \text{ in } m^2$ . Write a function which calculates *bmi*. BMI is used to broadly define different weight groups in adults 20 years old or older. Check if a person is *underweight*, *normal*, *overweight* or *obese* based the information given below.

- The same groups apply to both men and women.
- *Underweight*: BMI is less than 18.5
- *Normal weight*: BMI is 18.5 to 24.9
- *Overweight*: BMI is 25 to 29.9
- *Obese*: BMI is 30 or more

```
const bmi = (weight, height) => {
  let bmi1 = (weight / ((height * height) / 10000)).toFixed(2);
  if (bmi1 < 18.5) {
    return 'Underweight';
  }
  else if (bmi1 > 18.5 && bmi1 < 24.5) {
    return 'Normal Weight';
  }
}
```

```

    }
    else if(bmi1>25 && bmi1<29.9) {
        return 'Ovewrweight';
    }
    else if(bmi1>30) {
        return 'Obese';
    }
}
console.log(bmi(50, 161));

```

14. Write a function called *checkSeason*, it takes a month parameter and returns the season: Autumn, Winter, Spring or Summer.

```

const season = (month) => {
    if(month === 'december' || month === 'january' || month ===
'february') return 'Winter Season';
    else if(month === 'march' || month === 'april' || month ===
'may') return 'Spring Season';
    else if(month === 'june' || month === 'july' || month ===
'august') return 'Summer Season';
    else if(month === 'september' || month === 'october' || month
=== 'november') return 'Autumn Season';
}
console.log(season('june'));

```

15. Math.max returns its largest argument. Write a function findMax that takes three arguments and returns their maximum with out using Math.max method.

```

console.log(findMax(0, 10, 5))
10
console.log(findMax(0, -10, -2))
0

```

```

const findMax = (num1, num2, num3) => {
    if(num1 >= num2 && num1 >= num3) return num1;
    else if (num2 >= num1 && num2 >= num3) return num2;
    else return num3;
}

```

```
console.log(findMax(10,20,40));
```

## Exercises: Level 2

1. Linear equation is calculated as follows:  $ax + by + c = 0$ . Write a function which calculates value of a linear equation, *solveLinEquation*.

```
const solveLinEquation = (equation) => {
  let newEquation = equation;
  let signArray = [];
  let signIndex = [];
  for(var i = 0; i < equation.length; i++) {
    newEquation = newEquation.replace(' ', '');
    if(i == 0) {
      if(newEquation[i] == '-') {
        signArray.push('-');
        signIndex.push(i);
      } else {
        signArray.push('+');
        signIndex.push(i);
      }
    }
    if((newEquation[i] == '+' || newEquation[i] == '-') && i !== 0) {
      let sign = newEquation.charAt(i);
      signArray.push(sign);
      signIndex.push(i);
    }
  }
  //console.log(signArray);
  //console.log(signIndex);
  //console.log(newEquation);
  let a = newEquation.slice(signIndex[0], signIndex[1]-1);
  let b = newEquation.slice(signIndex[1], signIndex[2]-1);
  let c = newEquation.slice(signIndex[2],
newEquation.indexOf("="));
  console.log(a,b,c);

  // When x = 0;
  y = -(c)/b;
  console.log(`y=${y}`);

  // When y = 0;
  x = -(c)/a;
  console.log(`x=${x}`);
}
solveLinEquation("5x + 3y - 30 = 0")
```

2. Quadratic equation is calculated as follows:  $ax^2 + bx + c = 0$ . Write a function which calculates value or values of a quadratic equation, *solveQuadEquation*.

```
console.log(solveQuadratic()) // {0}
console.log(solveQuadratic(1, 4, 4)) // {-2}
console.log(solveQuadratic(1, -1, -2)) // {2, -1}
console.log(solveQuadratic(1, 7, 12)) // {-3, -4}
console.log(solveQuadratic(1, 0, -4)) //{2, -2}
console.log(solveQuadratic(1, -1, 0)) //{1, 0}
```

```
const solveQuadEquation = (a=0, b=0, c=0) => {
  let discriminant = b * b - 4 * a * c;
  let root1, root2;
  // condition for real and different roots
  if (discriminant > 0) {
    root1 = (-b + Math.sqrt(discriminant)) / (2 * a);
    root2 = (-b - Math.sqrt(discriminant)) / (2 * a);
    console.log(`The roots of quadratic equation are ${root1}
and ${root2}`);
  }
  // condition for real and equal roots
  else if (discriminant == 0) {
    root1 = root2 = -b / (2 * a);
    console.log(`The roots of quadratic equation are ${root1}
and ${root2}`);
  }
}
solveQuadEquation(1,-1,-2)
```

3. Declare a function name *printArray*. It takes array as a parameter and it prints out each value of the array.

```
let array = [1,2,3,4,5];
let printArray = (arr) => {
  for(var i=0; i<arr.length; i++){
    console.log(arr[i]);
  }
};
printArray(array);
```

4. Write a function name *showDateTime* which shows time in this format: 08/01/2020 04:08 using the Date object.

```
showDateTime()
08/01/2020 04:08
```

```
let todayDate = new Date();

function getReadableDate(date,sep) {
  let day = date.getDate();
```

```

let month = date.getMonth()+1;
let year = date.getFullYear();
let hours = date.getHours();
let mins = date.getMinutes();

if(day < 10) day = '0'+day;
if(month < 10) month = '0'+month;
if(hours < 10) hours = '0'+hours;
if(mins < 10) mins = '0'+mins;

return `${day}${sep}${month}${sep}${year} ${hours}:${mins}`;
}
console.log(getReadableDate(todayDate, '/'));

```

5. Declare a function name *swapValues*. This function swaps value of x to y.

```

swapValues(3, 4) // x => 4, y=>3
swapValues(4, 5) // x = 5, y = 4

```

```

let swapValues = (a,b) => {
  let temp = a;
  a = b;
  b = temp;
};

```

6. Declare a function name *reverseArray*. It takes array as a parameter and it returns the reverse of the array (don't use method).

```

console.log(reverseArray([1, 2, 3, 4, 5]))
//[5, 4, 3, 2, 1]
console.log(reverseArray(['A', 'B', 'C']))
//['C', 'B', 'A']

```

```

let reverseArray = (arr) => {
  let left =0;
  let right = arr.length -1;
  while(left < right){
    let temp = arr[left];
    arr[left] = arr[right];
    arr[right] = temp;
    left++;
    right--;
  }
  return arr;
};
reverseArray([1,2,3,4]);
reverseArray([1,2,3,4,5]);

```

7. Declare a function name *capitalizeArray*. It takes array as a parameter and it returns the - capitalizedarray.

```

let capitalizeArray = (arr) => {

```

```

    for(var i=0; i<arr.length;i++){
        arr[i] = arr[i].toUpperCase();
    }
    return arr;
};
capitalizeArray(['s','d','g']);

```

8. Declare a function name *addItem*. It takes an item parameter and it returns an array after adding the item

```

let addItem = (arr,item) => {
    arr.push(item);
    return arr;
};
addItem(['s','d','g'],2);

```

9. Declare a function name *removeItem*. It takes an index parameter and it returns an array after removing an item

```

let removeItem = (arr,index) => {
    arr.splice(index);
    return arr;
};
removeItem(['s','d','g'],2);

```

10. Declare a function name *sumOfNumbers*. It takes a number parameter and it adds all the numbers in that range.

```

let sumOfNumbers = (arr,range) => {
    let sum =0;
    if(range<arr.length){
        for(var i = 0; i<range;i++){
            sum = sum + arr[i];
        }
    }
    return sum;
};
sumOfNumbers([1,1,9,1,1,1],5);

```

11. Declare a function name *sumOfOdds*. It takes a number parameter and it adds all the odd numbers in that - range.

```

let sumOfOdds = (arr,range) => {
    let sum =0;
    if(range<arr.length){
        for(var i = 0; i<range;i++){
            if(arr[i]%2 != 0)
                sum = sum + arr[i];
        }
    }
    return sum;
};

```

```
sumOfOdds ([1,1,8,1,1,1],5) ;
```

12. Declare a function name *sumOfEven*. It takes a number parameter and it adds all the even numbers in that - range.

```
let sumOfEven = (arr,range) => {  
  let sum =0;  
  if(range<arr.length){  
    for(var i = 0; i<range;i++){  
      if(arr[i]%2 == 0)  
        sum = sum + arr[i];  
    }  
  }  
  return sum;  
};  
sumOfEven ([1,1,9,1,1,1],5) ;
```

13. Declare a function name *evensAndOdds* . It takes a positive integer as parameter and it counts number of evens and odds in the number.

```
evensAndOdds(100);  
The number of odds are 50.  
The number of evens are 51.
```

```
let evensAndOdds = (range) => {  
  let evenCount = parseInt(range/2) +1;  
  if(range%2 == 0) var oddCount = parseInt(range/2) ;  
  else var oddCount = evenCount;  
  console.log(`The number of odds are ${oddCount}. \n The  
number of evens are ${evenCount}`;  
};  
evensAndOdds (51) ;
```

14. Write a function which takes any number of arguments and return the sum of the arguments

```
sum(1, 2, 3) // -> 6  
sum(1, 2, 3, 4) // -> 10
```

```
let sumOfNumbers = (arr) => {  
  let sum =0;  
  for(var i = 0; i<arr.length;i++){  
    sum = sum + arr[i];  
  }  
  return sum;  
};  
sumOfNumbers ([1,1,9,1,1,1]) ;
```

15. Write a function which generates a *randomUserIp*.

```
const randomUserIp = () => {  
  let IP = "";  
  for(var i = 0; i < 4; i++) {
```



```

    randomNumber = Math.floor(Math.random() * 256);
    if(i !== 3)
        IP += `${randomNumber}.`;
    else
        IP += `${randomNumber}`;
    }
    console.log(IP);
}
randomUserIp();

```

16. Write a function which generates a *randomMacAddress*

```

const randomUserIp = () => {
    let IP = "";
    let str = "ABCDEF0987654321";
    for(var i = 0; i < 6; i++) {
        for(var j = 0; j < 2; j++) {
            randomNumber = Math.floor(Math.random() * str.length);
            IP += str[`${randomNumber}`];
        }
        if(i !== 5)
            IP += '-';
    }
    console.log(IP);
}
randomUserIp();

```

17. Declare a function name *randomHexaNumberGenerator*. When this function is called it generates a random hexadecimal number. The function return the hexadecimal number.

```

console.log(randomHexaNumberGenerator());
'#ee33df'

```

```

const randomHexaNumberGenerator = () => {
    let randomHexCode = "";
    let str = "abcdef1234567890";
    for(var i = 0; i < 6; i++) {
        let randomNumber = Math.floor(Math.random() * str.length);
        randomHexCode += str[`${randomNumber}`];
    }
    console.log(`#${randomHexCode}`);
}
randomHexaNumberGenerator();

```

18. Declare a function name *userIdGenerator*. When this function is called it generates seven character id. The function return the id.

```
console.log(userIdGenerator());  
41XTDbE
```

```
const userIdGenerator = () => {  
  let id = "";  
  let str =  
  "abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890";  
  for(var i = 0; i < 7; i++) {  
    let randomNumber = Math.floor(Math.random() * str.length);  
    id += str[`${randomNumber}`];  
  }  
  return id;  
}  
console.log(userIdGenerator());
```

### Exercises: Level 3

1. Modify the *userIdGenerator* function. Declare a function name *userIdGeneratedByUser*. It doesn't take any parameter but it takes two inputs using `prompt()`. One of the input is the number of characters and the second input is the number of ids which are supposed to be generated.

```
userIdGeneratedByUser()  
'kcsy2  
SMFYb  
bWmeq  
ZXOYh  
2Rgxf  
,  
  
userIdGeneratedByUser()  
'1GCSgPLMaBAVQZ26  
YD7eFwNQKNs7qXaT  
ycArC5yrRupyG00S  
UbGxOFI7UXSWAyKN  
dIV0SSUTgAdKwStr  
,
```

```
const userIdGeneratedByUser = () => {  
  let id = "";  
  let str =  
  "abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890  
  ";  
  let len = prompt("Enter No. of Characters: ");  
  let no = prompt("Enter No. of ID's to Generate");  
  for(var i = 0; i < no; i++) {
```

```

    id = "";
    for(let j = 0; j < len; j++) {
        let randomNumber = Math.floor(Math.random() *
str.length);
        id += str[`${randomNumber}`];
    }
    console.log(id);
}
}
userIdGeneratedByUser();

```

2. Write a function name *rgbColorGenerator* and it generates rgb colors.

```

rgbColorGenerator()
rgb(125,244,255)

```

```

const rgbColorGenerator = () => {
    let randomRGB = "";
    for(let i = 0; i < 3; i++) {
        let randomNumber = Math.floor(Math.random() * 256);
        if(i == 2) randomRGB += randomNumber;
        else {
            randomRGB += randomNumber+', ';
        }
    }
    console.log(`rgb(${randomRGB})`);
}
rgbColorGenerator()

```

3. Write a function ***arrayOfHexaColors*** which return any number of hexadecimal colors in an array.

```

const arrayOfHexaColors = (count) => {
    let randomHexCode = "";
    let hexArray = [];
    let str = "abcdef1234567890";
    for(let j = 0; j < count; j++) {
        randomHexCode = "";
        for(var i = 0; i < 6; i++) {
            let randomNumber = Math.floor(Math.random() *
str.length);
            randomHexCode += str[`${randomNumber}`];
        }
        hexArray.push(`#${randomHexCode}`);
    }
    return hexArray;
}
console.log(arrayOfHexaColors(5));

```

4. Write a function ***arrayOfRgbColors*** which return any number of RGB colors in an array.

```
const arrayOfRgbColors = (count) => {
  let randomRGB;
  let rgbArray = [];
  for(let j = 0; j < count; j++) {
    randomRGB = "";
    for(let i = 0; i < 3; i++) {
      let randomNumber = Math.floor(Math.random() * 256);
      if(i == 2) randomRGB += randomNumber;
      else {
        randomRGB += randomNumber+', ';
      }
    }
    rgbArray.push(`rgb(${randomRGB})`);
  }
  return rgbArray;
}
console.log(arrayOfRgbColors(2));
```

5. Write a function ***convertHexaToRgb*** which converts hexa color to rgb and it returns an rgb color.

```
const convertHexaToRgb = (hex) => {
  let r = hex.slice(0,2);
  let g = hex.slice(2,4);
  let b = hex.slice(4, 6);
  var aRgb = `rgb(${parseInt(r, 16)}, ${parseInt(g, 16)},
${parseInt(b, 16)})`;
  return aRgb;
}
console.log(convertHexaToRgb("1502BE"));
```

6. Write a function ***convertRgbToHexa*** which converts rgb to hexa color and it returns an hexa color.

```
const convertRgbToHexa = (r,g,b) => {
  let hexR = r.toString(16);
  let hexG = g.toString(16);
  let hexB = b.toString(16);
  if(hexR.length < 2) hexR = `0`+hexR;
  if(hexB.length < 2) hexB = `0`+hexB;
  if(hexG.length < 2) hexG = `0`+hexG;
  var hex = `#${hexR}${hexG}${hexB}`;
  return hex;
}
console.log(convertRgbToHexa(21, 2, 190));
```

7. Write a function ***generateColors*** which can generate any number of hexa or rgb colors.

```
console.log(generateColors('hexa', 3)) // ['#a3e12f', '#03ed55',  
      '#eb3d2b']  
console.log(generateColors('hexa', 1)) // '#b334ef'  
console.log(generateColors('rgb', 3)) // ['rgb(5, 55, 175)', 'rgb(50,  
      105, 100)', 'rgb(15, 26, 80)']  
console.log(generateColors('rgb', 1)) // 'rgb(33,79, 176)'
```

```
const generateColors = (type, count) => {  
  if(type.toLowerCase() == 'hexa') {  
    let randomHexCode = "";  
    let hexArray = [];  
    let str = "abcdef1234567890";  
    for(let j = 0; j < count; j++) {  
      randomHexCode = "";  
      for(var i = 0; i < 6; i++) {  
        let randomNumber = Math.floor(Math.random() *  
str.length);  
        randomHexCode += str[`${randomNumber}`];  
      }  
      hexArray.push(`#${randomHexCode}`);  
    }  
    return hexArray;  
  } else if(type.toLowerCase() == 'rbg'){  
    let randomRGB;  
    let rbgArray = [];  
    for(let j = 0; j < count; j++) {  
      randomRGB = "";  
      for(let i = 0; i < 3; i++) {  
        let randomNumber = Math.floor(Math.random() * 256);  
        if(i == 2) randomRGB += randomNumber;  
        else {  
          randomRGB += randomNumber+', ';  
        }  
      }  
      rbgArray.push(`rgb(${randomRGB})`);  
    }  
    return rbgArray;  
  }  
}  
console.log(generateColors("rbg",5));
```

8. Call your function ***shuffleArray***, it takes an array as a parameter and it returns a shuffled array

```
function shuffle(array) {  
  let currentIndex = array.length,  randomIndex;
```

```

while (currentIndex != 0) {
    randomIndex = Math.floor(Math.random() * currentIndex);
    currentIndex--;
    var temp = array[currentIndex];
    array[currentIndex] = array[randomIndex];
    array[randomIndex] = temp;
}
return array;
}
var arr = [2, 11, 37, 42];
console.log(shuffle(arr));

```

9. Call your function *factorial*, it takes a whole number as a parameter and it return a factorial of the number

```

function factorial(number) {
    let fact = 1;
    while(number > 0) {
        fact = fact*number;
        number--;
    }
    return fact;
}
console.log(factorial(3));

```

10. Call your function *isEmpty*, it takes a parameter and it checks if it is empty or not

```

function isEmpty(str) {
    return (!str || str.length === 0);
}
console.log(isEmpty());

```

11. Call your function *sum*, it takes any number of arguments and it returns the sum.

```

function sum(...arguments) {
    var s = 0;
    for (var i=0; i < arguments.length; i++) {
        s += arguments[i];
    }
    return s;
}
console.log(sum(10,20,30))

```

12. Write a function called *sumOfArrayItems*, it takes an array parameter and return the sum of all the items. Check if all the array items are number types. If not give return reasonable feedback.

```

function sumOfArrayItems(array) {
    var s = 0;
    for (var i=0; i < array.length; i++) {
        if(typeof(array[i]) !== 'number')
            s += array[i];
    }
}

```

```

        else
            return `Array Contains Non Numeric Items`;
    }
    return s;
}
console.log(sumOfArrayItems([10,20,"30"]))

```

13. Write a function called *average*, it takes an array parameter and returns the average of the items. Check if all the array items are number types. If not give return reasonable feedback.

```

function average(array) {
    var s = 0;
    for (var i=0; i < array.length; i++) {
        if(typeof(array[i]) == 'number')
            s += array[i];
        else
            return `Array Contains Non Numeric Items`;
    }
    return s/array.length;
}
console.log(average([10,20,30]))

```

14. Write a function called *modifyArray* takes array as parameter and modifies the fifth item of the array and return the array. If the array length is less than five it return 'item not found'.

```

console.log(modifyArray(['Avocado', 'Tomato', 'Potato','Mango',
'Lemon','Carrot']));

['Avocado', 'Tomato', 'Potato','Mango', 'LEMON', 'Carrot']

console.log(modifyArray(['Google', 'Facebook','Apple',
'Amazon','Microsoft', 'IBM']));

['Google', 'Facebook','Apple', 'Amazon','MICROSOFT', 'IBM']

console.log(modifyArray(['Google', 'Facebook','Apple', 'Amazon']));

'Not Found'

```

```

function modifyArray(array) {
    if(array.length < 5) return `Not Found`;
    array[4] = array[4].toUpperCase();
    return array;
}
console.log(modifyArray(['Avocado', 'Tomato', 'Potato','Mango',
'Lemon','Carrot']));

```

15. Write a function called *isPrime*, which checks if a number is prime number.

```

function isPrime(number) {
    let prime = true;

```

```

    for(i=2; i<number; i++) {
        if(number%i==0) {
            prime = false;
            break;
        }
    }
    if(prime == true && (number != 1 && number != 0))
        console.log(number + " is a Prime Number");
    else
        console.log(number + " is not a Prime Number");
}
console.log(isPrime(2));

```

16. Write a functions which checks if all items are unique in the array.

```

function unique(array) {
    let checkedElements = [];
    for(let i = 0; i < array.length; i++) {
        if(checkedElements.includes(array[i])) return 'Array does
not contain unique Elemets';
        checkedElements.push(array[i]);
    }
    return 'Array contains unique elements';
}
console.log(unique([1,2,3,4]));

```

17. Write a function which checks if all the items of the array are the same data type.

```

function checkDatatype(array) {
    let type = typeof(array[0]);
    for(let i = 0; i < array.length; i++) {
        if(typeof(array[i]) != type) return 'Array does not contain
same datatypes';
    }
    return 'Array contains dame datatypes';
}
console.log(checkDatatype(["1",2,3,4]));

```

18. JavaScript variable name does not support special characters or symbols except \$ or \_ . Write a function **isValidVariable** which check if a variable is valid or invalid variable.

```

function isValidVariable (name) {
    var regex = '^[a-zA-Z_$][a-zA-Z\d_$]*$';
    if((name).match(regex))
        console.log("Valid name");
    else
        console.log("Invalid name");
}
isValidVariable ("Priyanka%");

```



19. Write a function which returns array of seven random numbers in a range of 0-9. All the numbers must be unique.

sevenRandomNumbers()  
[(1, 4, 5, 7, 9, 8, 0)]

```
const sevenRandomNumbers = () => {  
  let array = [];  
  for(var i = 0; i < 6; i++) {  
    let randomNumber = Math.floor(Math.random() * 10);  
    if(array.includes(randomNumber)) {  
      i--;  
      continue;  
    }  
    else  
      array.push(randomNumber);  
  }  
  return array;  
}  
console.log(sevenRandomNumbers());
```

20. Write a function called reverseCountries, it takes countries array and first it copy the array and returns the reverse of the original array

```
const reverseCountries = (array) => {  
  let index = array.length-1;  
  let newArray = array.slice();  
  for(let i = 0; i < array.length/2; i++) {  
    let temp = array[i];  
    array[i] = array[index];  
    array[index] = temp;  
    index--;  
  }  
  return array;  
}  
console.log(reverseCountries(["India", "USA", "China",  
"Japan"]));
```