Find the largest number in an array

```
  function large(array){
  var large=array[0];
  for( let n of array){
  if(array[n]>large){
  large=arra[n]}
  }
  console.log(large)}
```

Find the reverse of the string

```
function reverseString(text) {

return text.split('').reverse().join('');

// Code goes here

 var rev=""

 for(let char of text){

 rev=char+rev

 }

 return rev

}
```

Find the number of vowels in a string

```
function vowelsCounter(text) {

// Code goes here

 const arr=['a','i','o','u','e']

 var counter=0;

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

 if(arr.includes(text[i].toLowerCase()))

 {

 counter++;
```

```
 }

 }

return counter;

}
```

Finding the Most Recurring Character:

```
function maxRecurringChar(text) {

// Code goes here

var charMap={};

let repeating=0;

let rChar=''

for(let char of text ){

if( charMap.hasOwnProperty(char)){

charMap[char]++;

}

else{

charMap[char]=1;}

}

// console.log(chararray)

for(let char in charMap){

if(charMap[char]>repeating){

repeating=charMap[char];

rChar=char;

}

}

return(rChar)

}
```

Creating a char map:

```
function createCharMap(text) {

 let charMap = {}
 for (let char of text) {
 if (charMap.hasOwnProperty(char)) {
 charMap[char]++
 } else {
 charMap[char] = 1
 }
 }

 return charMap
 }

 console.log(createCharMap('dogg'))
```

Anagrams

Burtt force

```
 if(stringA.length===stringB.length){
 const ArrofCharA=stringA.toLowerCase().split("").sort();
 const ArrofCharb=stringB.toLowerCase().split("").sort();
 for(let i=0; i<=stringB.length;i++){
 if(ArrofCharA[i]!==ArrofCharb[i]){
 return false
 }

 }
 return true


 }
 return false
```

Using chaMaps

```
function createCharMap(text) {
 let charMap = {}

 for (let char of text) {
 if (charMap.hasOwnProperty(char)) {
```

```
charMap[char]++
} else {
charMap[char] = 1
}
}

return charMap
}

if (stringA.length === stringB.length) {
let stringAMap = createCharMap(stringA)
let stringBMap = createCharMap(stringB)
for (let char in stringAMap) {
if (stringAMap[char] !== stringBMap[char]) {
return false
}
}
return true
} else {
return false
}
```

Repeating a text

```
const repeatText=(text, num)=>{
var rText=''
while(num>0){
rText=rText+text;
num--;

}
console.log(rText)}
```

Removing the duplicates elements in an array

```
Var array=[1, 2, 3, 5, 1, 5, 9, 1, 2, 8];
 array.filter((item, index)=>{return array.indexOf(item)===index})
```

With out filter

```
var array=[1, 2, 3, 5, 1, 5, 9, 1, 2, 8];

let dep=[]
for(let i=0;i<array.length; i++){
if(i===array.indexOf(array[i])){
dep.push(array[i])

}}
```

Second most repeating char in text

```
function secondMaxRecurringChar(text) {
var charMap = {};
let repeating;

for (let char of text) {
if (charMap.hasOwnProperty(char)) {
charMap[char]++;
} else {
charMap[char] = 1;
}
}

let mapArray = Object.values(charMap);
mapArray.sort((a, b) => a - b);

let idx = mapArray[mapArray.length - 2];

for (let char in charMap) {
if (charMap[char] === idx) {
repeating = char;
}
}

return repeating;
}
```

what the output of the code

```
var arr=[10,20, 30, 40];

for(var i=0;i<arr.length;i++){
  setTimeout(function(){
    console.log(i);
  },3000);
}

// output:4
```

```
var arr=[10,20, 30, 40];

for(let i=0;i<arr.length;i++){
  setTimeout(function(){
    console.log(i);
  },3000);
}

// output:0,1,2,3
```