



JavaScript coding question

Reverse array

Problem (a)

```
const data = ["ambuj", "singh", "data"];
const reverseFunction = (data) => {
    let reverseData = [];
    for (let i = data.length - 1; i >= 0; i--) {
        reverseData.push(data[i]);
    }
    return reverseData;
};

console.log(reverseFunction(data));
```

Output:

```
[ 'data', 'singh', 'ambuj' ]
```

Problem (b)

```
const data = ["ambuj", "singh", "data"];
const reverseFunction =(data)=>{
    let reverseData=[];
    for(let j=0; j<=data.length-1; j++){
        let word = data[j];
```

```

        let reverseWord= ""
for(let i =word.length-1; i>=0; i--){
    reverseWord+=word[i]
}
reverseData.push(reverseWord)
}
return reverseData;
}
console.log(reverseFunction(data))

```

Output:

```
[ 'jubma', 'hgnis', 'atad' ]
```

Reverse last element

```

const data = ["ambuj", "singh", "data"];
const reverseFunction =(data)=>{
    let reverseData=[];
for(let j=0; j<=data.length-1; j++){
    if(data.length-1==j){
        let word=data[j];
        let reverseDataLast='';
for(let i=word.length-1; i>=0; i--){
    reverseDataLast += word[i]
}
reverseData.push(reverseDataLast)
}
else{
    reverseData.push(data[j])
}
}
return reverseData;
}
console.log(reverseFunction(data))

```

Output:

```
[ 'ambuj', 'singh', 'atad' ]
```

Currying Function

```
const curryingFuction = (a)=>(b)=>(c)=>a+b+c
```

```
console.log(curryingFuction(2)(3)(4))
```

OutPut - 9

Fizz Buzz

```
for (let i = 1; i <= 50; i++) {  
    if (i % 3 === 0 && i % 5 === 0) {  
        console.log('fizzbuzz')  
    }  
    else if (i % 5 === 0) {  
        console.log("fizz")  
    }  
    else if (i % 3 === 0) {  
        console.log("buzz")  
    }  
    else {  
        console.log(i)  
    }  
}
```

Output:

```
1  
2  
buzz  
4  
fizz  
buzz  
7  
8
```

```
buzz  
fizz  
11  
buzz  
13  
14
```

Check the element present in array

```
const funct =(array,element)=>{  
for(let i=0; i<=array.length-1; i++){  
if(array[i]==element){  
return true  
}  
}  
return false  
}  
  
console.log(funct(["ambuj", "singh", "data"], "data"))
```

Output -
True

Max min value in array

```
const findMaxAndMinValue =(data)=>{  
let minValue = data[0];  
  
for(let i=1; i<=data.length-1; i++){  
if(data[i]<=minValue){  
minValue=data[i];  
}  
}  
return minValue;  
}
```

```
console.log(findMaxAndMinValue([1, 2, 5, 6, 7]));
```

Unique array

```
const uniqueArray =(data)=>{
  let uniqueData=[];
  data.forEach(v=>{
    if(!uniqueData.includes(v)){
      uniqueData.push(v)
    }
  })
  return uniqueData;
}
console.log(uniqueArray([64, 423, 3267, 4238, 4238, 423, 4211, 23, 44]))
```

Sum of numbers

```
const sumOfNumbers=(data)=>{
  let totalNumber=0
  return data.map(v=>{
    totalNumber +=v;
    return totalNumber;
  })
}

console.log(sumOfNumbers([1, 24, 5, 52, 252, 52]))

//reverse of array

const reverseArray =(data)=>{
  let reverseData=[];
  for(let i=data.length-1; i>0;i--){
    reverseData.push(data[i])
  }
  return reverseData;
```

```
}
```

```
console.log(reverseArray([1, 3, 42, 4, 5]))
```

Check every element is greater than 50

```
let number =[111, 222, 54, 421, 223];
```

```
console.log(number.every(v=>v>50))
```

```
//check Index (find index)
```

```
let number =[10, 15, 20, 25, 15];
```

```
let target =25;
```

```
let findIndexData = number.indexOf(target);
```

```
console.log(findIndexData)
```

```
//remove the data from the array
```

```
const removeData= (data)=>{
  return data.filter (value=>value<20)
}
console.log(removeData([12,34,2,242,23]))
```

Reverse string

```
let str = "ambuj singh";
const reverseString =(str)=>{
  let reverseData='';
  for(let i=str.length-1; i>0; i--){
    reverseData+=str[i];
  }
  return reverseData;
}
console.log(reverseString(str));
```

Missing numbers

```
const numbers =[1, 2, 3, 4, 5, 7];

const checkMissingNumber= (numbers)=>{
let missingNumbers=[];
for(let i=0; i<=numbers.length-1;i++){
    if(!numbers.includes(i+1)){
        missingNumbers.push(i+1)
    }
}
return missingNumbers;
}

console.log(checkMissingNumber(numbers))
```

Call apply bind

1. call⇒ it used for the invoke the function

```
var person = {
name:"john"
};

function sayHello(){
return `hello` + [this.name](http://this.name/)
}

console.log(sayHello.call(person))
```

2. apply⇒ it is similar to call, it take argument in array

```
const person = {name:"john"}

function sayHello (greeting){
return greeting+ " "+ [this.name](http://this.name/);

}
```

```
console.log(sayHello.apply(person, ["hello"]))
```

3. bind ⇒ bind create a new function when called

```
const person = {name : "john"}  
function sayHello (greeting){  
    console.log(greeting + ' , '+ this.name);  
}  
const greetingJohn = sayHello.bind(person);  
greetingJohn("hello")
```

Reverse string

```
const str = 'a quick brown fox jumps over the lazy dog';  
  
const reverseString=(str)=>{  
    const word = str.split(" ");  
    let reverseStirng ='';  
    for(let i=0; i<=word.length-1; i++){  
        let words = word[i];  
        for(let j=words.length-1; j>=0; j--){  
            reverseStirng +=words[j];  
        }  
        reverseStirng+='\n'  
    }  
    return reverseStirng;  
}  
console.log(reverseString(str))
```

Output -

a kciuq nworb xof spmuj revo eht yzal god

Expected output ⇒ [1,3,6,10,15];

```

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

const myFunction =(arr)=>{
let newArr=[];
let sum=0;
arr.forEach(v=>{
    sum+=v;
    newArr.push(sum)
})
return newArr;
}

console.log(myFunction(arr));

```

Generics ⇒ it is used for reusable type

```

function createCode <T>(data: T): T {
  return data;
}
console.log(createCode<number>(60));
console.log(createCode<string>("Hello"));
console.log(createCode<boolean>(true));

```

Check the element present in array

```

function contain (arr, elements){
arr.map(v=>{
if(v==elements){
return true;
}
})
return false;
}
console.log(contain([1, 2, 3, 4, 5], 7))

```

Repeated array

```
const checkRepeatedArray =(arr)=>{
  let nonRepeateArr=[];
  let repeatedArr=[];
  arr.map(v=>{
    if(!repeatedArr.includes(v)){
      repeatedArr.push(v);
    }
    else{
      nonRepeateArr.push(v);
    }
  })
  return nonRepeateArr;
}

console.log(checkRepeatedArray([1,3,44,5,3,22, 22]))
```