JavaScript Mock Questions

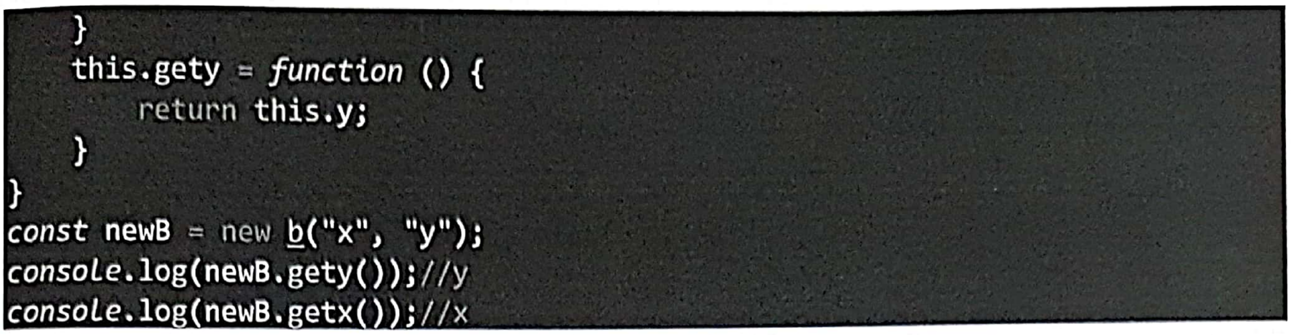
```
1. let x ={
    a:2,
    b:3
}
console.log(Object.entries(x));//[ [ 'a', 2 ], [ 'b', 3 ] ]
for(i in x){
    console.log(x[i])
}
```

```
2. let x = "hi";
//reverse string
let y= "ih";
console.log(x.split('').reverse().join(''));//ih
```

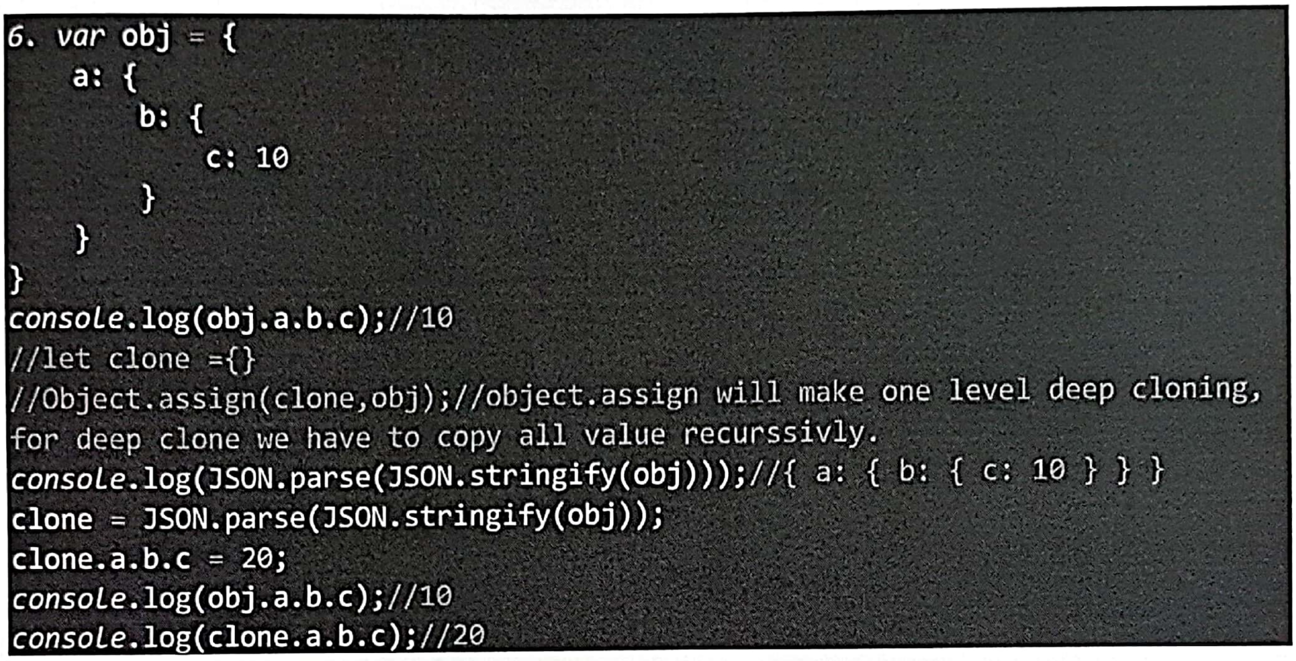
```
3. let obj = {
    a: 1,
    b: 2,
    getA() {
        console.log(this.a);
        return this;//with out this obj.getA().getB(); wont work
    },
    getB() {
        console.log(this.b)
    }
}
obj.getA().getB();
```

```
4. Array.prototype.print = (arr) => {
    console.log(String(arr));
}
[].print([2, 3]);//I want to get 1,2
```

```
5. const a = function (x) {
    this.x = x;
}
const b = function (x, y) {
    this.y = y;
    a.call(this, x);
    this.getx = function () {
        return this.x;
```

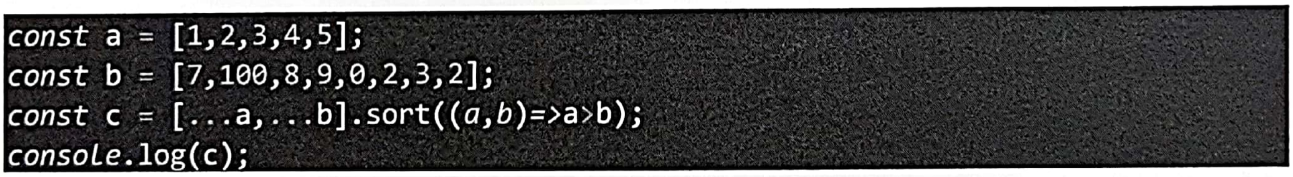


```
    }
    this.gety = function () {
        return this.y;
    }
}
const newB = new b("x", "y");
console.log(newB.gety());//y
console.log(newB.getx());//x
```



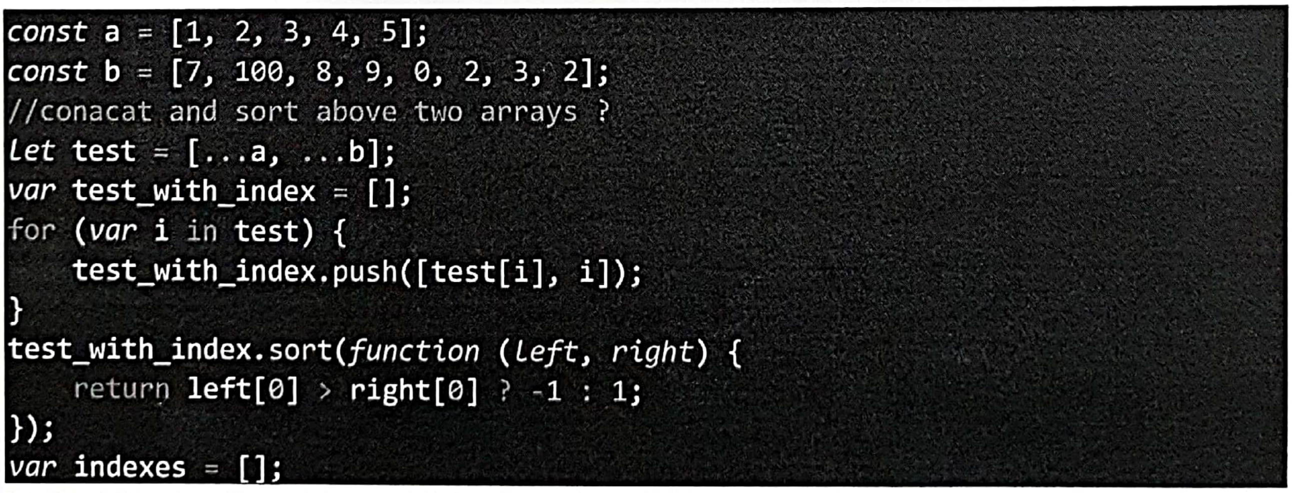
```
6. var obj = {
    a: {
        b: {
            c: 10
        }
    }
}
console.log(obj.a.b.c);//10
//let clone ={}
//Object.assign(clone,obj);//object.assign will make one level deep cloning,
for deep clone we have to copy all value recurssivly.
console.log(JSON.parse(JSON.stringify(obj)));//{ a: { b: { c: 10 } } }
clone = JSON.parse(JSON.stringify(obj));
clone.a.b.c = 20;
console.log(obj.a.b.c);//10
console.log(clone.a.b.c);//20
```

7.

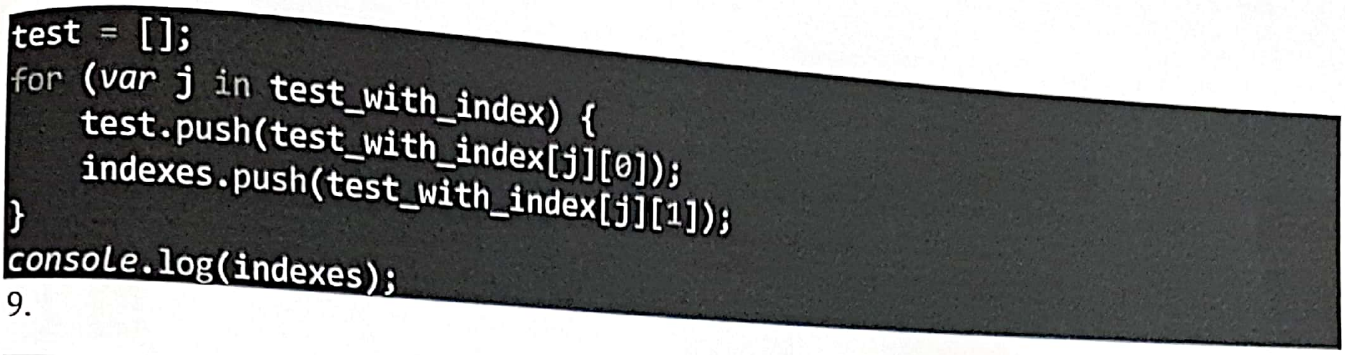


```
const a = [1,2,3,4,5];
const b = [7,100,8,9,0,2,3,2];
const c = [...a,...b].sort((a,b)=>a>b);
console.log(c);
```

8.

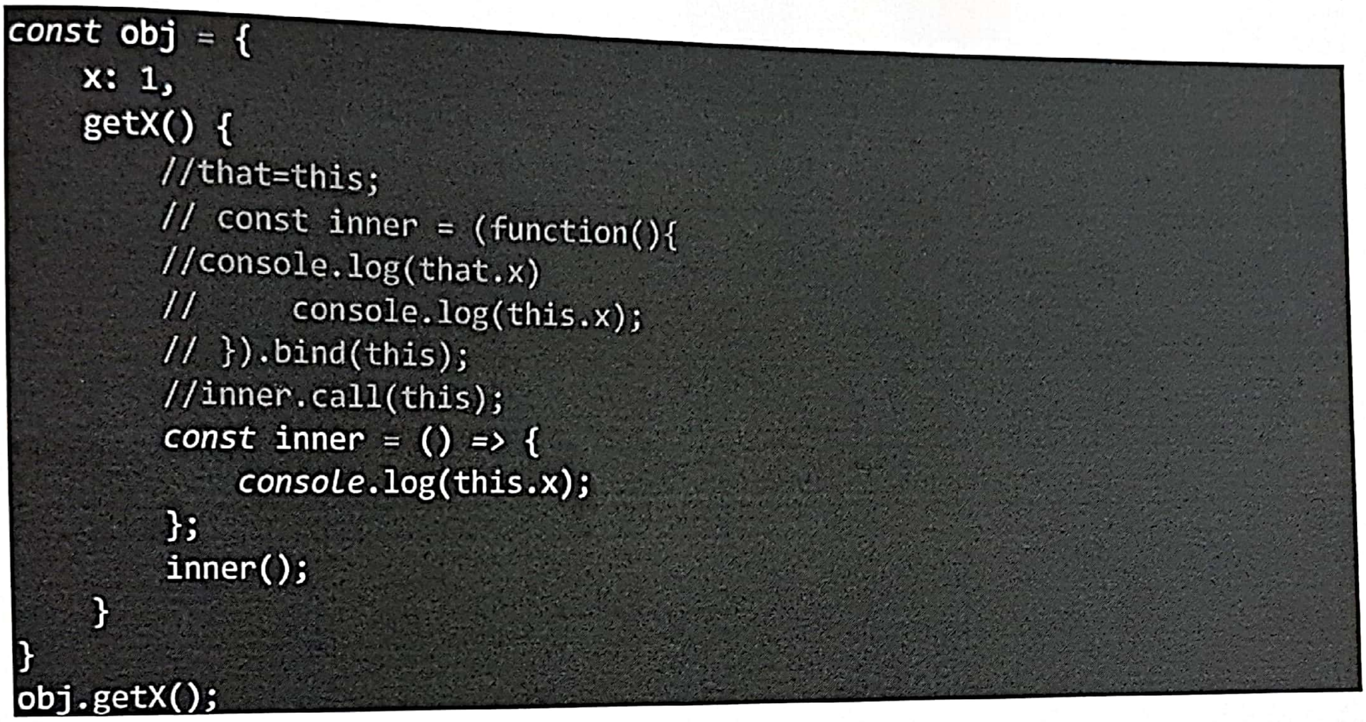


```
const a = [1, 2, 3, 4, 5];
const b = [7, 100, 8, 9, 0, 2, 3, 2];
//conacat and sort above two arrays ?
let test = [...a, ...b];
var test_with_index = [];
for (var i in test) {
    test_with_index.push([test[i], i]);
}
test_with_index.sort(function (left, right) {
    return left[0] > right[0] ? -1 : 1;
});
var indexes = [];
```



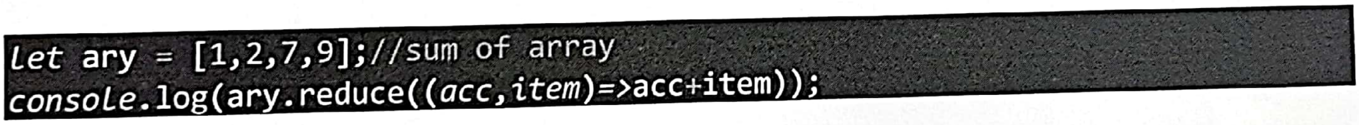
```
test = [];
for (var j in test_with_index) {
    test.push(test_with_index[j][0]);
    indexes.push(test_with_index[j][1]);
}
console.log(indexes);
```

9.



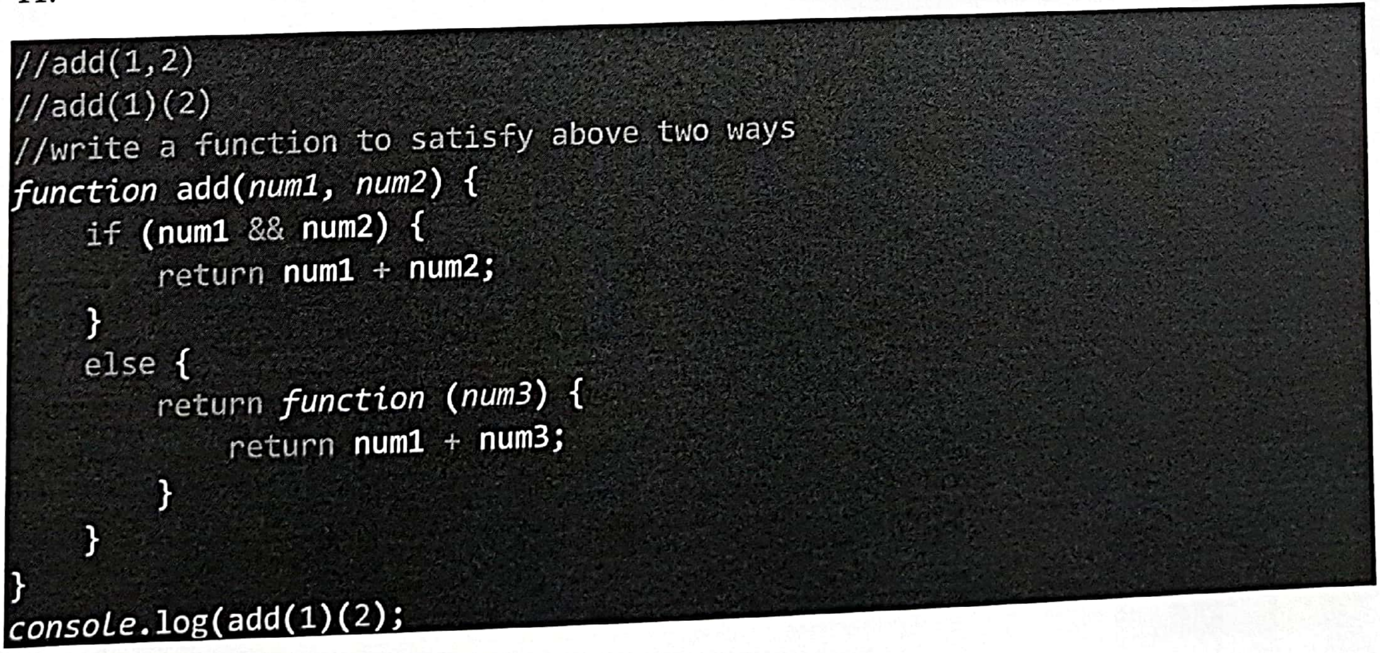
```
const obj = {
    x: 1,
    getX() {
        //that=this;
        // const inner = (function(){
        //console.log(that.x)
        //      console.log(this.x);
        // }).bind(this);
        //inner.call(this);
        const inner = () => {
            console.log(this.x);
        };
        inner();
    }
}
obj.getX();
```

10.



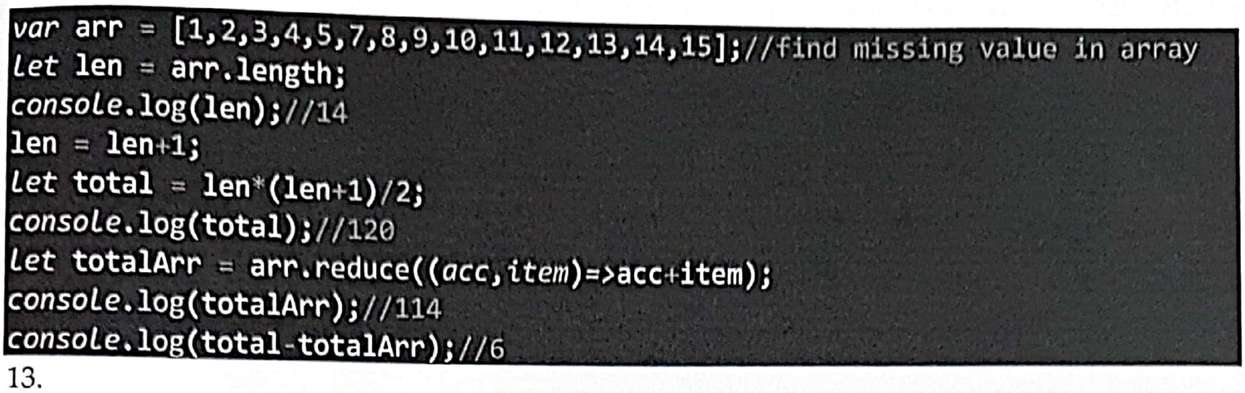
```
let ary = [1,2,7,9];//sum of array
console.log(ary.reduce((acc,item)=>acc+item));
```

11.



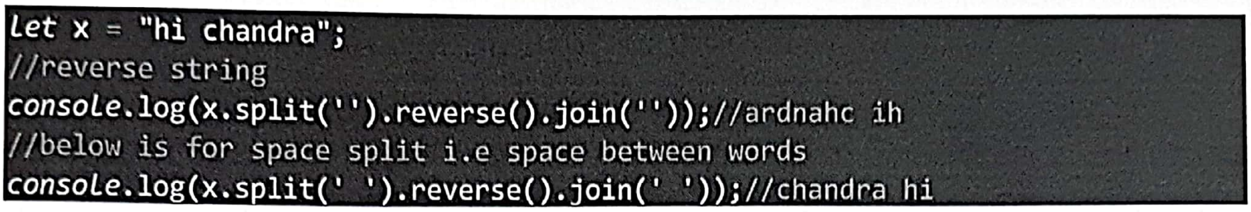
```
//add(1,2)
//add(1)(2)
//write a function to satisfy above two ways
function add(num1, num2) {
    if (num1 && num2) {
        return num1 + num2;
    }
    else {
        return function (num3) {
            return num1 + num3;
        }
    }
}
console.log(add(1)(2));
```

12.



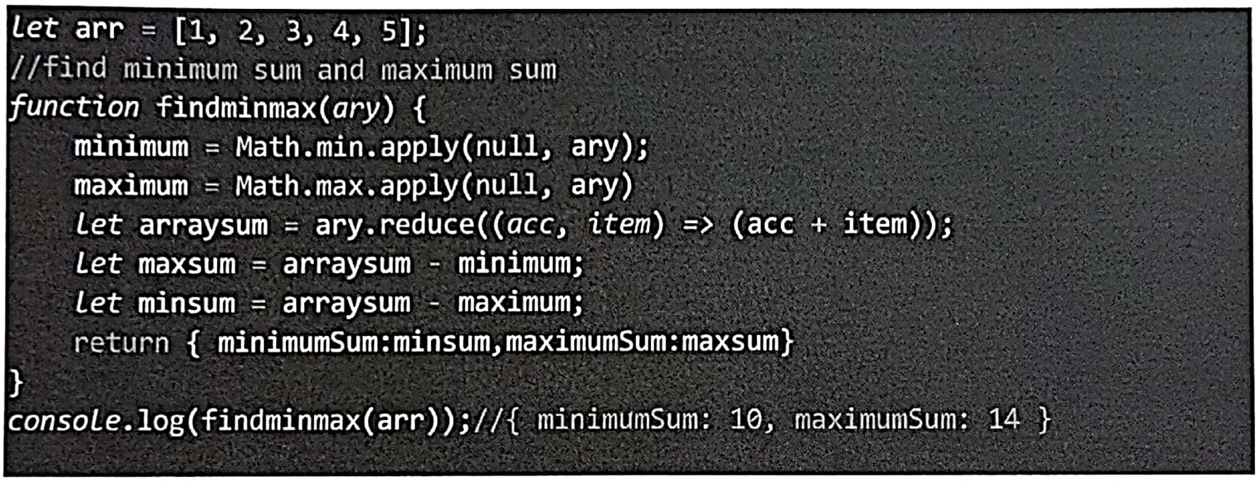
```
var arr = [1,2,3,4,5,7,8,9,10,11,12,13,14,15];//find missing value in array
let len = arr.length;
console.log(len);//14
len = len+1;
let total = len*(len+1)/2;
console.log(total);//120
let totalArr = arr.reduce((acc,item)=>acc+item);
console.log(totalArr);//114
console.log(total-totalArr);//6
```

13.



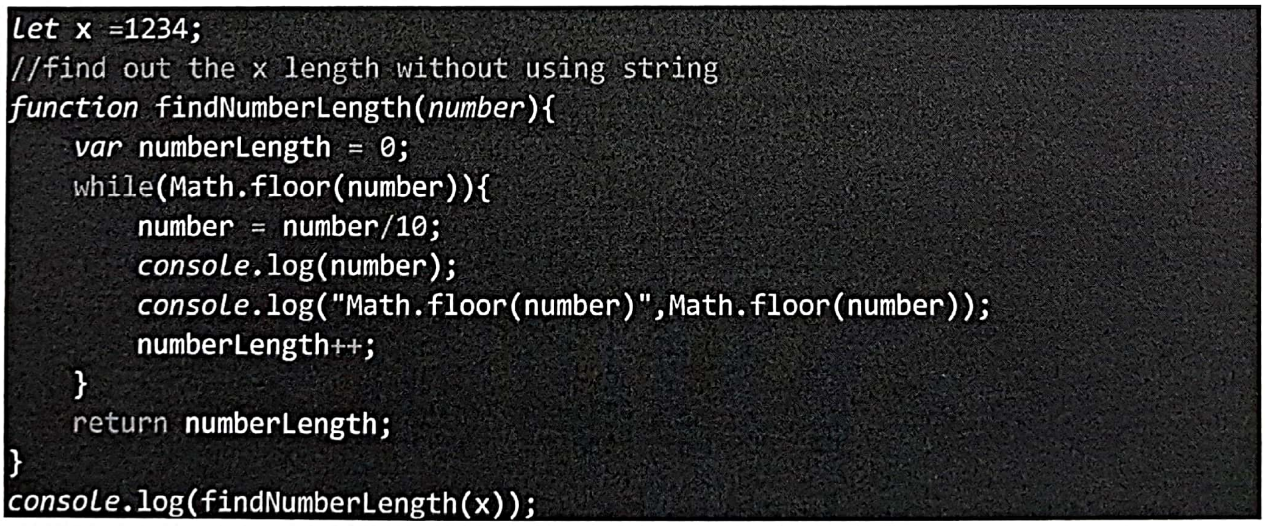
```
let x = "hi chandra";
//reverse string
console.log(x.split('').reverse().join(''));//ardnahc ih
//below is for space split i.e space between words
console.log(x.split(' ').reverse().join(' '));//chandra hi
```

14.



```
let arr = [1, 2, 3, 4, 5];
//find minimum sum and maximum sum
function findminmax(ary) {
    minimum = Math.min.apply(null, ary);
    maximum = Math.max.apply(null, ary)
    let arraysum = ary.reduce((acc, item) => (acc + item));
    let maxsum = arraysum - minimum;
    let minsum = arraysum - maximum;
    return { minimumSum:minsum,maximumSum:maxsum}
}
console.log(findminmax(arr));//{ minimumSum: 10, maximumSum: 14 }
```

15.



```
let x =1234;
//find out the x length without using string
function findNumberLength(number){
    var numberLength = 0;
    while(Math.floor(number)){
        number = number/10;
        console.log(number);
        console.log("Math.floor(number)",Math.floor(number));
        numberLength++;
    }
    return numberLength;
}
console.log(findNumberLength(x));
```