1.

```
try{
const obj=new Promise(function(a,b){
    b(100)
    a(20)
})
const res=await obj;
}catch(e){
  console.log("catch",e);
}
```

what is output in console ?

○ Catch, 100
○ Catch, undefined
○ Catch,20
○ Catch,null

Ans: A

2.

```
const arr=[22,11,33,55,555]

arr.slice(4,2)
```

○ []
○ [33,55]

Ans: A

3.

How many ways can get the data from promise object ?

☑ then
☑ await
☐ using dot notation
☐ using [] notation

Ans: AB

4.

(function(){

console.log("called")

})()

output?

◉ called
○ Runtime Error
○ Syntax Error
○ None of the above

Ans: A

5.

var a=10;

var b=20;

var obj={a,b}

is it valid syntax ?

◉ Yes
○ No

Ans: A

6.

```
function f1(a){
    return function(b){
        console.log(a+b)
    }
}
```
To get the the output 30 -

- ◉ f1(10)(20)
- ○ f1(30)
- ○ Syntax Error
- ○ Runtime Error

Ans: A

7.

```
var arr=[1,2,3];

var x=arr.includes(1,1);

x?
```

- ○ -1
- ○ 0
- ○ true
- ◉ false

Ans: D

8.
```
function f1(a){
    a()
}
function f2(){
    console.log("f2 called");
}
```

f1(f2)

output?

○ Error
● f2 called
○ Syntax error
○ None of the above

Ans: B

9.

Below which methods are inside sessionStorage ?

☑ getItem
☑ setItem
☑ removeItem
☑ clear

Ans: ABCD

10.

var name="Sachin"

name.includes("ach");

?

○ 1
● true
○ false
○ 2

Ans: B

11.

var arr=[22,33,44,55]

```
arr.find((val)=>{
   return val>35
})
```

?

- ○ 44
- ○ 44,55
- ○ [44,55]

Ans: A

12.

finally block will execute , when ever the exception raised or not raised in try block ?

- ○ Yes
- ○ No

Ans: A

13.

```
var obj={
   n1:10,
   n1:20,
   n1:30
}
```

obj ?

- ○ {n1:30}
- ○ {n1:10}
- ○ {n1:20}
- ○ {n1:10,n1:20,n1:30}

Ans: A

14.

var a=10;

var b=6

a%b ?

○ 4
○ 1

Ans: A

15.

a()

a member type should be ?

○ number
● function
○ boolean
○ array

Ans: B

16.

```
async function fn(){
    const res=await <promise object>
}
```

is it valid syntax

● Yes
○ No

Ans: A

17.

```
function fn(n1,n2,…data){

}
```

is it valid syntax ?

○ Yes
○ No

Ans: A

18.

```
var loc;
```

what is the loc value?

○ null
○ undefined
○ 0
○ ""

Ans: B

19.

```
function f1(){
    function f2(){
        return 10
    }
    return f2
}

var x=f1()
```

x type is ?

○ number
◉ function

Ans: B

20.

Below how many ways we can pass the asynchronous action results to caller ?

◉ callbacks
○ Observables
○ class
○ functions

Ans: AB

WRITE AGAIN