Q-1)
```
(function(x, f = () => x) {
  var x;
  var y = x;
  x = 2;
  return [x, y, f()];
})(1)
```

A. [2, 1, 1]
B. [2, undefined, 1]
C. [2, 1, 2]
D. [2, undefined, 2]

Q-2)
```
(function() {
  return [
    (() => this.x).bind({ x: 'inner' })(),
    (() => this.x)()
  ]
}).call({ x: 'outer' })
```

A. ['inner', 'outer']
B. ['outer', 'outer']
C. [undefined, undefined]
D. Error

Q-3)
```
(function() {
  let f = this ? class g { } : class h { };
  return [
    typeof f,
    typeof h
  ];
})();
```

A. ["function", "undefined"]
B. ["function", "function"]
C. ["undefined", "undefined"]
D. Error

Q-4)
```
let arr = [ ];
for (let { x = 2, y } of [{ x: 1 }, 2, { y }]) {
  arr.push(x, y);
}
arr;
```

A. [2, { x: 1 }, 2, 2, 2, { y }]
B. [{ x: 1 }, 2, { y }]
C. [1, undefined, 2, undefined, 2, undefined]
D. Error

Q-5)
```
(typeof (new (class { class () {} })))
```

A. "function"
B. "object"
C. "undefined"
D. Error

Q-6)
```
[...[...'...']].length
```

A. 1
B. 3
C. 6
D. Error

Q-7)
```
((...x, xs)=>x)(1,2,3)
```

A. 1
B. 3
C. [1,2,3]
D. Error

Q-8)
```
let x, { x: y = 1 } = { x }; y;
```

A. undefined
B. 1
C. { x: 1 }
D. Error

Q-9) Which is not Array methods in ES6

A. Array.of()
B. Array.from()
C. Array.prototype.find()
D. Array.copy()

Q-10) What is ES6?

A. Acronym of JSON And XML
B. Acronym of ECMAScript 6
C. Acronym of Ex Scripting 6
D. None Of the Above

Q-11)
```
function mys(...params) {
return params;
}
let x = mys(1,23,4)
```

A. "x" is undefined
B. "x" becomes [1,23,4]
C. "x" becomes "1 23 4"
D. "x" becomes 1 23 4

Q-12) What is a Promise() in ES6?

A. Tool for managing asynchronous control flow. A promise represents an operation expected to complete in the future.
B. The opposite of Amateurmise().
C. Something you say, when you want someone to believe you.
D. None of the above.

Q-13) What is/are the advantages of the arrow function?

A. It reduces code size.
B. The return statement is optional for a single line function.
C. Lexically bind the context.
D. All the above

Q-14) What is stored in the triangle array?

```
let point = [1,3], segment = [point,[5,5]], triangle = [...segment,[1,8]];
```

A. [[1,3], [5,5], [1,8]]
B. [1,3,5,5,1,8]
C. 23
D. None of the above

Q-15) Which one is correct using the spread operator?

```
let num1 = [40,50,60];
let num2 = [10,20,30,...num1,70,80,90,100];
console.log(num2);
```

A. [10, 20, 30, 40, 50, 60, 70, 80, 90, 100]
B. [40, 50, 60]
C. [10, 20, 30, 70, 80, 90, 100]
D. None of these

Q-16) ES6 gives an alternative way to assign variables. Can you guess what the below code does?

```
let a = 12, b = 3;
[a, b] = [b, a];
```

A. Swaps the values inside a and b, without using extra variables.
B. Makes both a and b equal 12.
C. Creates an array that contains a and b.
D. None of above

Q-17)

```
const obj = {
  outer: function() {
    const self1 = this
    const inner1 = () => {
      const self2 = this
      const inner2 = () => {
        const self3 = this
      }
    }
  }
}
```

A. self1 === self3
B. self2 === self3
C. self1 === self2
D. self2 !== self3

Q-18)

```
function show(...args) {
let sum = 0;
for (let i of args) {
sum += i;
}
console.log("Sum = "+sum);
}
show(10, 20, 30);
```

A. Sum = 60
B. Undefined
C. [10,20,30]
D. [60]

Q-19)

```
let x=150
if(x>100)
    let x =1
console.log(x);
```

A. 150
B. 1
C. 0
D. None of these

Q-20)
How do you empty an array?

A) Array. Length = 0;
B) Array()= 0;
C) Empty(Array) =0;
D) None of these

Q-21) Which of the following can be used to call a JavaScript Code Snippet?

Function/Method
Preprocessor
Triggering Event
RMI

Q-22) Which of the following scoping type does JavaScript use?

Sequential
Segmental
Lexical
Literal