

Q1: const numbers = [1, 5, 18, 2, 77, 108]; print the odd numbers. You're not allowed to use for, while, do...while, for..of, forEach loop.

```
console.log(numbers.filter(item => (item%2) !== 0));
```

Q2: Create a function using function declaration named sum with one parameter of Array type, the returned result is the sum of all elements which are greater than 20.

```
sum([10, 20, 50, 30, 8]);
```

```
function sum(arr){  
    let result = arr.reduce((sum, current) => sum + current, 0);  
    return result;  
};
```

Q3: Create a function using function expression named getNewArray with one parameter of String Array, return a new array which contains all string, length is greater than and equal to 5, and contains letter 'a'.

```
getNewArray(["Hello", "Wonderful", "Happy", "People", "Have a great day"]);
```

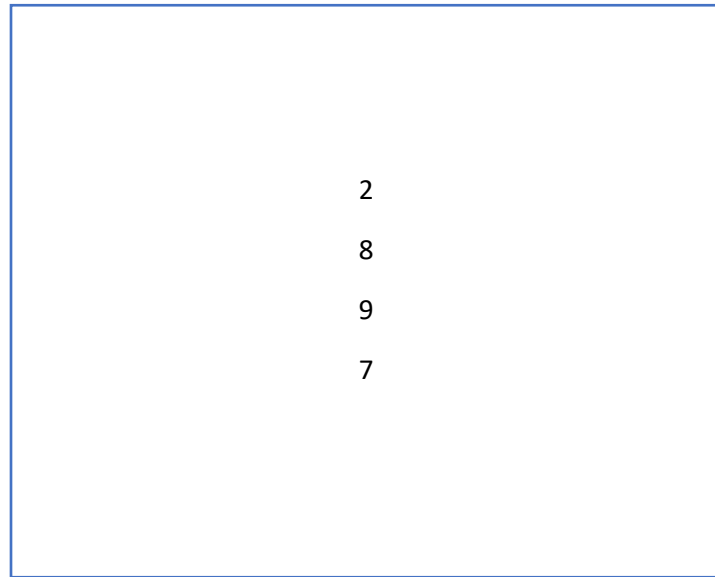
```
function getNewArray(arr){  
    let result = arr.filter(item => item.length >= 5 && item.includes('a'));  
    return result;  
}
```

Q4:

What will be the output?

```
var a = 2;
let b = 3;
function outer() {
  let c = 5;
  var d = 7;
  return function inner() {
    b = 8;
    let c = 9;

    console.log(a);
    console.log(b);
    console.log(c);
    console.log(d);
  }
}
outer();
```



Based on the code above,

1. What's the LE of global EC after creation phase finished before execution phase starts?
Global EC in creation LE:{outer: null, a: undefined, outer: fn}, TDZ:{b}
2. What's the LE of global EC after execution phase finished?
Global EC in execution LE:{outer: null, a:2, outer: fn, b: 3}
3. What's the LE of function outer EC after creation phase finished before execution phase starts?
outer EC in creation LE:{outer: global , arguments{length:0}, d: undefined}, TDZ:{c}
4. What's the LE of function outer EC after execution phase finished?
outer EC in execution LE:{outer: global , arguments{length:0}, d: 7, inner: fn, c: 5}
5. What's the LE of function inner EC after creation phase finished before execution phase starts?
inner EC in execution LE:{outer: outer , arguments{length:0}}, TDZ{c}
6. What's the LE of function inner EC after execution phase finished?
inner EC in execution LE:{outer: outer , arguments{length:0} , c: 9}
global scope => LE: { outer: null, a: 2, b: 8 }

Q5: When the HTML is

```
<body>
  <p style="color:red">First</p>
  <div class="central">
    <p class="item">Second</p>
    <ul>
      <li id="item">Third</li>
    </ul>
  </div>
</body>
```

And the CSS is:

```
body {
  background-color: yellow;
  color: blue;
}
p { color: orange; }
.central, .item {
  color: green;
}
#item {
  background-color: white;
}
ul{
  color: purple;
  background-color: beige;
}
```

Specify what colors will show on the screen for the:

	Background-color	color
First	yellow	red
Second	yellow	green
Third	white	purple

Q6: Write a regular expression that matches a string containing a date in the format mm/dd/yyyy.

```
^(0[1-9]|1[0-2])\V(0[1-9]|1\d|2\d|3[01])\V\d{4}$
```

Q7: Write a JavaScript function that takes an object with a firstName and lastName property, and returns a new object with a fullName property that combines the two names. Use an regular function to define the fullName property and the this keyword to refer to the firstName and lastName properties.

```
function foo(obj) {  
  
    return {  
  
        firstname: obj.firstname,  
  
        lastname: obj.lastname,  
  
        fullname: function () { return this.firstname + " " + this.lastname }  
  
    }  
  
}  
  
let newObj = foo({ firstname: "john", lastname: "smith" })  
  
console.log(newObj.fullname());
```

Q8: Write a JavaScript function that takes an object with a firstName and lastName property, and a callback function that logs a message using the this keyword. Bind the callback function to the object using the bind method, and call the bound function with no arguments.

```
function foo(obj, callback) {  
  
    callback.bind(obj)();  
  
}  
  
function callback() {  
  
    console.log("Hello " + this.firstname);  
  
}  
  
foo({ firstname: "john", lastname: "smith" }, callback);
```