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.

Ans:

```
const numbers = [1, 5, 18, 2, 77, 108];
let i = 0;
function printOdd() {
  if (i >= numbers.length) {
    return;
  }
  if (numbers[i] % 2 !== 0) {
    console.log(numbers[i]);
  }
  i++;
  printOdd();
}
printOdd();
```

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]);
```

Ans:

```
function sum(arr) {
  let sum = 0;
  for (let i = 0; i < arr.length; i++) {
    if (arr[i] > 20) {
      sum += arr[i];
    }
  }
  return sum;
}
```

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"]);

Ans:

```
const getNewArray = function(strArr) {
  const newArr = [];
  for (let i = 0; i < strArr.length; i++) {
    if (strArr[i].length >= 5 && strArr[i].includes('a')) {
      newArr.push(strArr[i]);
    }
  }
  return newArr;
}
```

Q4:

```
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()();
```

What will be the output?

```
2

8

9

7
```

Based on the code above,

1. What's the LE of global EC after creation phase finished before execution phase starts?

- The var keyword was used to declare a variable with the value of 2. The entire code will have access to this variable because it is kept in the global scope.
- a let-declared variable named b with the value of 3. This variable will be accessible only within the block scope in which it was declared (in this case, the global scope), even though it will be kept in the global scope as well.

- outer(), which is a function declared using the function keyword and saved in the global scope.
- The global this keyword designates the global object, often the window object in a browser environment.

2. What's the LE of global EC after execution phase finished?

3. What's the LE of function outer EC after creation phase finished before execution phase starts?

4. What's the LE of function outer EC after execution phase finished?

5. What's the LE of function inner EC after creation phase finished before execution phase starts?

6. What's the LE of function inner EC after execution phase finished?

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	yellow
Second	transparent	green
Third	white	green

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

let regex = /^\d{2}\/\d{2}\/\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 arrow function to define the fullName property and the this keyword to refer to the firstName and lastName properties.

Ans:

```
function createFullName(obj) {

 return {

   ...obj,

   fullName: () => `${this.firstName} ${this.lastName}`

 };

}
```

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.

Ans:

```
function logMessage(obj, callback) {

  const boundCallback = callback.bind(obj);

  boundCallback();

}
```

Student ID: 615881
Name: Osama Ahmed Abdelhamid