Do not use var here. Try to get by with const. If necessary, use let.
Use existing JavaScript methods (for strings, arrays etc.) to complete tasks.
https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Number
https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/String
https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array
https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Object
https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Promise
https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/async_function

1. **Write a JavaScript function that returns the Fibonacci series up to a certain number.**
Input: 8
Output: [0, 1, 1, 2, 3, 5]

Input: 610
Output: [0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377]

2. **Write a JavaScript function to find the unique elements from two arrays.**
Example :
console.log(difference([1, 2, 3], [100, 2, 1, 10]));
["1", "2", "3", "10", "100"]
console.log(difference([1, 2, 3, 4, 5], [1, [2], [3, [[4]]],[5,6]]));
["1", "2", "3", "4", "5", "6"]
console.log(difference([1, 2, 3], [100, 2, 1, 10]));
["1", "2", "3", "10", "100"]

3. **Write a JavaScript function to create a case-insensitive search**
Example :
console.log(caseInsensitiveSearch('JavaScript Exercises', 'exercises'));
"Matched"
console.log(caseInsensitiveSearch('JavaScript Exercises', 'Exercises'));
"Matched"
console.log(caseInsensitiveSearch('JavaScript Exercises', 'Exercisess'));
"Not Matched"

4. **Write a JavaScript function to get a copy of the object where the keys have become the values and the values the keys.**

5. **Write a JavaScript function to convert an object into a list of `[key, value]` pairs.**

6. **Write a JavaScript function to uncamelize a string**
Example :
console.log(uncamelize('helloWorld'));
console.log(uncamelize('helloWorld','-'));
console.log(uncamelize('helloWorld','_'));
"hello world"
"hello-world"
"Hello_world"

7. **Write a JavaScript function to count the occurrence of a substring in a string.**

8. **Flat an array (use reduce here) and sort it (by ascending).**
Input:
[1, 2, 1000, 300, [400, [3, 10, [11, 12]], [1, 2, [3, 4]], 5, 6]]
Output:
[1, 1, 2, 2, 3, 3, 4, 5, 6, 10, 11, 12, 300, 400, 1000]

9. **Write a function that delete null and undefined values from the array.**
The function takes two parameters: array, callback, runs for 5 seconds and then calls a callback function-parameter that displays the result of the execution or an error.

10. **Write a function that returns Promise, which is resolved after 6 seconds.**
11. **Write a JavaScript program to run a given array of promises in series.**

12. Working with **Promises**
To complete this task you will need
Sing up at https://geocode.xyz/ and get the API key.
Read about **fetch()**. Use it to send requests in your browser.
Request example:
http://geocode.xyz/Minsk?json=1&auth=<your_key_here>

- Send parallel requests for information about cities - Minsk, Madrid, Rome. Display the result in format: city - country.
- Using Promise.race get the country of these cities - Paris, Nice. Display result.
- Write a function that returns a promise that resolves any cities' names after 3 second. After you get names send parallel requests for information about cities. Display the country of the cities.
- Handle errors