**Question 01**

Change isPrime() that takes in a single number parameter and returns a new promise.  
Using setTimeout and after 500 milliseconds, the promise will either resolove or reject.  
if the input is prime number, the promise resolves with {prime: true}.  
if the input is not a prime number, it rejects with {prime: false}.

Write code and call your promisified function and test it for both scenarios (resolve and reject)

You may use the following function to determine if the number is prime or not

const isPrime = num => {

for(let i = 2, s = Math.sqrt(num); i <= s; i++)

if(num % i === 0) return false;

return num > 1;

}

After making change to isPrime function, if you make a call as below:

console.log('start');

isPrime(7)

.then(res => console.log(res))

.catch(err => console.error(err));

console.log('end');

In the console, you should expect:

start

end

{ prime: true } //this is printed after 500 millionseconds

 let isPrime = new Promise((num) => {

    for (let i = 2, s = Math.sqrt(num); i <= s; i++)

        if (num % i === 0) {

            setTimeout(() => "false", 1000);

        }

    setTimeout(() => num > 1, 1000);

});

**Question 02**

Create a method named removeDuplicates() method which will remove duplicates in an Array. Make sure this method is available for only Array objects which means can call like code below. In the code below, this removeDuplicates()  method runs synchronously.

console.log(`start`);

[4, 1, 5, 7, 2, 3, 1, 4, 6, 5, 2].removeDuplicates();

console.log(`end`);

// Output:

// start

// [4, 1, 5, 7, 2, 3, 6]

// end

Array.prototype.removeDuplicates = function () {

    var temp = [];

    var j = 0;

    for (let i = 0; i < this.length - 1; i++) {

        if(uniqe(this[i],i, this)){

            temp.push(this[i]);

        }

    }

     console.log(temp);

}

function uniqe(num, loc, arr){

    for (let i = loc+1; i < arr.length; i++) {

        if (arr[i] === num) {

            return false

        }

    }

    return true;

}

console.log(`start`);

[4, 1, 5, 7, 2, 3, 1, 4, 6, 5, 2].removeDuplicates();

console.log(`end`);

Rewrite an asynchronous version of it removeDuplicatedAsync() as following:

console.log(`start`);

[4, 1, 5, 7, 2, 3, 1, 4, 6, 5, 2].removeDuplicatesAsync();

console.log(`end`);

// Output:

// start

// end

// [4, 1, 5, 7, 2, 3, 6]

Array.prototype.removeDuplicatesAsync = setTimeout(function () {

    var temp = [];

    var j = 0;

    for (let i = 0; i < this.length - 1; i++) {

        if(uniqe(this[i],i, this)){

            temp.push(this[i]);

        }

    }

     console.log(temp);

}

,1000 ) ;

function uniqe(num, loc, arr){

    for (let i = loc+1; i < arr.length; i++) {

        if (arr[i] === num) {

            return false

        }

    }

    return true;

}

console.log(`start`);

[4, 1, 5, 7, 2, 3, 1, 4, 6, 5, 2].removeDuplicatesAsync();

console.log(`end`);