Task 1: Create a new promise that resolves after a set number of seconds and returns

a greeting.

Task 2: Fetch data from an API using promises, and then chain another promise to

process this data.

Task 3: Create a promise that either resolves or rejects based on a random number.

Task 4: Use Promise.all to fetch multiple resources in parallel from an API.

Task 5: Chain multiple promises to perform a series of asynchronous actions in

sequence.

5. Async/await:

Task 1: Rewrite a promise-based function using async/await.

Task 2: Create an async function that fetches data from an API and processes it.

Task 3: Implement error handling in an async function using try/catch.

Task 4: Use async/await in combination with Promise.all.

Task 5: Create an async function that waits for multiple asynchronous operations to

complete before proceeding.

6. Modules introduction, Export and Import:

Task 1: Create a module that exports a function, a class, and a variable.

Task 2: Import the module in another JavaScript file and use the exported entities.

Task 3: Use named exports to export multiple functions from a module.

Task 4: Use named imports to import specific functions from a module.

Task 5: Use default export and import for a primary function of a module.

7. Browser: DOM Basics:

Task 1: Select an HTML element by its ID and change its content using JavaScript.

Task 2: Attach an event listener to a button, making it perform an action when clicked.

Task 3: Create a new HTML element and append it to the DOM.

Task 4: Implement a function to toggle the visibility of an element.

Task 5: Use the DOM API to retrieve and modify the attributes of an element.

//1.Recursion and stack

//task1:

/\*let num=prompt("Enter a value");

let num1=Number(num);

function calculate(num1){

    if(num1==0){

        return 1;

    }else{

        return num1\*calculate(num1-1);

    }

}

console.log(calculate(num1));

\*/

//task2:

/\*let num=prompt("Enter a value");

let num1=Number(num);

function fibonacci(num1){

    if(num1==1){

        return 0;

    }else if(num1==2){

        return 1

    }

    else{

        return (fibonacci(num1-1)) + (fibonacci(num1-2));

    }

}

 console.log(fibonacci(num1));

\*/

//task3:

/\*let num=prompt("Enter a value");

let num1=Number(num);

function total(num1){

    if(num1==0){

        return 1;

    }else if(num1<0){

        return 0;

    }else{

        return total(num1-3) + total(num1-2) + total (num1-1);

    }

}

console.log("Combinations="+total(num1));

task4

const arr =[1,2,3,[5,6,7]];

const arr1=arr.flat(Infinity);

console.log(arr1);

task 5

function towerOfHanoi(n, from\_rod, to\_rod, aux\_rod)

{

        if (n == 0)

        {

            return;

        }

        towerOfHanoi(n - 1, from\_rod, aux\_rod, to\_rod);

        console.log("Move disk " + n + " from rod " + from\_rod +

        " to rod " + to\_rod);

        towerOfHanoi(n - 1, aux\_rod, to\_rod, from\_rod);

    }

    var N = 3;

    towerOfHanoi(N, 'A', 'C', 'B');

//2.JSON and variable length arguments/spread syntax

//task1

function sum(...args){

    let total =0;

    for(let arg of args){

        total=total+arg;

    }

   return total;

}

console.log(sum(1,2,3));

console.log(sum(9,1));

console.log(sum());

task 2

function sum(x,y,z){

    return x+y+z;

}

let number=[1,2,3];

console.log(sum(...number));

task 3

var s1={

    name:"dhanush",

    id:115

}

var s2=JSON.parse(JSON.stringify(s1));

s1.name="kumar";

console.log(s1);

task 4

function merge(a,b){

var c={

    ...a,

    ...b

}

return c;

}

let a={

    name:"dhanush",

    id:"717821E"

}

let b={

    surname:"kumar",

    surid:"115"

}

console.log(merge(a,b));

task 5

let obj={

    fruit:"apple",

    count:5

}

let obj1=JSON.parse(JSON.stringify(obj))

console.log(obj1);

//3.Closure

//task 1

function one(){

    let a=10

    function two(b){

        return a\*b;

    }

    return (two(5));

}

console.log(one());

//task 2

let count=3;

function counter(){

    function incre(){

        return count++;

    }

    return incre();

}

console.log(counter());

console.log(counter());

console.log(counter());

//task 3

 //4.promises,promlses chalnlng

 //task 1

 function delayGreeting(seconds) {

    return new Promise((resolve, reject) => {

      if (typeof seconds !== 'number' || seconds < 0) {

        reject('Invalid input');

      } else {

        setTimeout(() => {

          resolve('Hello!');

        }, seconds \* 1000);

      }

    });

  }

  delayGreeting(2)

    .then(greeting => {

      console.log(greeting);

    })

    .catch(error => {

      console.error(error);

    });

  \*/