/\*

Создайте функцию-конструктор Calculator, который создаёт объекты с тремя методами:

read() запрашивает два значения при помощи prompt и сохраняет их значение в свойствах объекта.

sum() возвращает сумму введённых свойств.

mul() возвращает произведение введённых свойств.

Например:

let calculator = new Calculator();

calculator.read();

alert( "Sum=" + calculator.sum() );

alert( "Mul=" + calculator.mul() );

\*/

function Calculator() {

    this.read = function () {

        this.a = + prompt('Enter a:', '');

        this.b = + prompt('Enter b:', '');

    };

    this.sum = () => this.a + this.b;

    this.mul = () => this.a \* this.b;

}

let calculator = new Calculator();

calculator.read();

console.log("Sum=" + calculator.sum());

console.log("Mul=" + calculator.mul());

/\*Напишите функцию-конструктор Accumulator(startingValue).

Объект, который она создаёт, должен уметь следующее:

Хранить «текущее значение» в свойстве value. Начальное значение устанавливается в аргументе конструктора startingValue.

Метод read() использует prompt для получения числа и прибавляет его к свойству value.

Таким образом, свойство value является текущей суммой всего, что ввёл пользователь при вызовах метода read(), с учётом начального значения startingValue.

Ниже вы можете посмотреть работу кода:

let accumulator = new Accumulator(1); // начальное значение 1

accumulator.read(); // прибавит ввод prompt к текущему значению

accumulator.read(); // прибавит ввод prompt к текущему значению

alert(accumulator.value); // выведет сумму этих значений

\*/

function Accumulator(startingValue) {

    this.value = startingValue;

    this.read = () => {

        this.value += + prompt('enter number:', '');

    };

}

let accumulator = new Accumulator(1);

accumulator.read();

accumulator.read()

console.log(accumulator.value);

// Напишите функцию sumTo(n), которая вычисляет сумму чисел 1 + 2 + ... + n.

function sumTo(n) {

    let res = n == 1 ? n : n + sumTo(n - 1);

    return res;

}

let averegeSum = sumTo(4);

// Задача – написать функцию factorial(n), которая возвращает n!, используя рекурсию.

function factorial(n) {

    let res = n == 1 ? n : n \* factorial(n - 1);

    return res;

}

let factorNum = factorial(5);