1. Реалізуйте функцію getPromise(message, delay), яка приймає текстове повідомлення message і цілочисельне значення затримки delay (в мс) і повертає Promise, який чекає задану кількість часу (використовуючи аргумент delay) і завершується повідомленням message.

Приклад застосування функції:

getPromise("test promise", 2000).then(function(data) {

console.log(data);

});

Результат: через 2 сек в консолі виводиться "test promise"

2. Реалізуйте функцію calcArrProduct(arr), яка приймає масив чисел. Функція повинна повернути Promise, робота якого завершується поверненням добутку елементів масиву, якщо вони є типу Numbers, або повідомленням " Error! Incorrect array! " у випадку, якщо хоча б 1 елемент масиву нечисловий.

Приклад застосування функції:

calcArrProduct([3,4,5]).then(result => console.log(result)); // 60

calcArrProduct([5,"user2", 7, 12]).then(result => console.log(result));

// "Error! Incorrect array!"

3. Заданий цикл for, який виводить послідовність чисел від 0 до 10 з випадковим інтервалом (від 0 до N мілісекунд). Використовуючи проміси потрібно змінити цикл так, щоб числа виводилися в строгій послідовності від 0 до 10. Наприклад, якщо виведення нуля займає 4 секунди, а одиниці 2 секунди, то одиниця повинна дочекатися виведення нуля і тільки потім почати свій відлік (щоб дотримуватися послідовності).

Для розв’язку задачі необзідно застосувати задану функцію delay(i, time), яка повертає проміс, який резолвиться поточним значенням числа-лічильника циклу і, яке виводиться через час time мілісекунд.

Приклад роботи:

const delay = (i, time) => new Promise(resolve => setTimeout(() => resolve(i), time));

function showNumbers() {

// your code with using delay(i, time)

}

showNumbers();

Результат:

з різним рендомним інтервалом (наприклад від 0 до 3 сек) в консоль послідовно виводяться числа 0,1,2,3,4,5,6,7,8,9

4. Задачу номер 3 потрібно реалізувати без застосування промісів. Переписати функцію showNumbers, використовуючи конструкцію async/await.