

Index	Question
01	Write a function sum that takes two numbers and a callback function as arguments. The function should add the two numbers together and pass the result to the callback function.
02	Write a function forEach that takes an array and a callback function as arguments. It should iterate over the array and call the callback function with each element of the array.
03	Write a function that returns a Promise. The Promise should resolve after a specified amount of time (e.g., 1 second) with a message "Promise resolved!".
04	Create two functions, asyncTask1 and asyncTask2, each returning a Promise that resolves after a random time between 1 to 3 seconds with a message. Chain these two promises so that asyncTask2 starts only after asyncTask1 resolves. Log the final messages when both promises are resolved.
05	Write a function that returns a Promise which either resolves with a given message after a certain time or rejects with an error message. Implement error handling to catch any rejections and log the error message.
06	Write a function that takes an array of promises as input and returns a new Promise. The new Promise should resolve with an array containing the resolved values of all input promises or reject if any of the promises in the array rejects.
07	Write an async function getData() that simulates fetching data from an API using fetch(). Inside the function, await the result of the fetch request and return the parsed JSON response.
08	Create an async function getDataErrorHandle() that fetches data from an API using fetch(). Handle any errors that occur during the fetch operation by catching them and logging an error message.
09	Implement an async function performTask() that performs three asynchronous tasks sequentially: Task 1 takes 2 seconds, Task 2 takes 3 seconds, and Task 3 takes 1 second. Ensure that Task 2 starts only after Task 1 completes and Task 3 starts only after Task 2 completes. Log the completion message after all tasks are done.
10	Implement an async function waitForTimeout() that takes a time in milliseconds as input. The function should resolve after waiting for the specified time using setTimeout.