### **Assignment 1: Callback Functions**

1. **Task**: Write a function fetchData that simulates fetching data from a server using a callback function.
   * The function should take a callback that processes the data after a delay of 2 seconds.
   * Use setTimeout to simulate the server delay.
   * The data should be an array of user names.
   * Implement error handling in the callback function to simulate a case where the server might fail.

### **Assignment 2: Working with Objects**

1. **Task**: Create an object bookLibrary to manage a collection of books.
   * The object should have the following properties and methods:
     + books: An array of book objects (each book has title, author, and yearPublished).
     + addBook(book): Adds a new book to the collection.
     + getBooksByAuthor(author): Returns all books by a given author.
     + removeBook(title): Removes a book by title.
     + Add a method getAllBooks to return a list of all book titles.

### **Assignment 3: Analyzing JavaScript Heap Memory**

1. **Task**: Create a program that continuously adds data to an array to simulate a memory leak.
   * Monitor heap memory usage using Chrome DevTools or performance.memory.
   * Implement a cleanup mechanism to prevent memory issues.
   * Use Chrome DevTools to capture a memory snapshot and analyze retained objects to observe the simulated memory leak.

### **\*\*Assignment 4: Working with map(), filter(), and \*\*reduce()**

#### **Task 1: Use map() to transform data**

* Create an array of product objects with properties name, price, and category.
* Use map() to create a new array with product names in uppercase.

#### **Task 2: Use filter() to extract specific data**

* Use filter() to create an array of products that belong to the 'Electronics' category.

#### **Task 3: Use reduce() to calculate a total**

* Use reduce() to calculate the total price of all products in the array.

#### **Task 4: Combine map(), filter(), and reduce()**

* Create a function that calculates the total price of products from a specific category using map(), filter(), and reduce().

### **\*\*Assignment 5: Callback Functions with map(), filter(), and \*\*reduce()**

1. **Task**: Create a processData function that accepts an array of numbers and a callback.
   * If the callback is filterOdd, filter out even numbers.
   * If the callback is doubleNumbers, double each number.
   * If the callback is calculateSum, return the sum of all numbers. **Bonus Task**: Implement a callback to find the maximum number in the array.