Programs to practice in the Lab

Topics Covered: Spread and Rest operators, Destructuring, Symbols, Function Generators

---

Question 1:

Write a function mergeArrays that accepts any number of arrays as arguments and merges them into a single array. Use the **spread operator** to accomplish this.

Example Input:

mergeArrays ([1, 2], [3, 4], [5, 6])

Example Output:

[1, 2, 3, 4, 5, 6]

Question 2:

Create a function calculateTotal that takes an arbitrary number of numeric values (using the **rest operator**) and returns their sum.

Example Input:

calculateTotal (10, 20, 30, 40)

Example Output:

100

Question 3:

Write a function updateEmployeeDetails that takes an object representing an employee's details and updates their role while maintaining the other properties using the **spread operator**.

Example Input:

const employee = {name: 'Afroz', role: 'Developer', age: 28, location: 'IN'}

updateEmployeeDetails (employee, 'Senior Developer')

Example Output:

{name: 'Afroz', role: 'Senior Developer', age: 28, location: 'IN'}

Question 4: You are given multiple objects representing user details and preferences. Write a function mergeUserData that merges all the objects into one using the **spread operator**.

Example Input:

const userDetails = {name: 'CVR', age: 25}

const userAddress = {address: 'Mangalpally', city: 'Hyderabad'}

const userPreferences = {theme: 'Engineering Edu', language: 'EN'}

Example Output:

{name: 'CVR', age: 25, address: 'Mangalpally', city: 'Hyderabad',
theme: 'Engineering Edu', language: 'EN'}

Question 5:

Write a function updateProductInfo that extracts the id and name properties from a product object and then adds new properties (discount and inStock) using the **spread operator**.

Example Input:

const product = {id: 101, name: ‘Laptop’, price: 1000, category: ‘Electronics’}

Example Output:

{id: 101, name: ‘Laptop’, discount: 10, inStock: true}

Question 6:

Given the following object:

const user = {name: ‘Afroz’, age: 30, city: ‘Hyderabad’, country: ‘INDIA’ };

Use **object destructuring** to extract the properties name and city into individual variables.

Question 7:

Given the following object with nested objects:

const person = {name: ‘Alice’, address: {street: ‘123 Main St’, city: ‘Los Angeles’, zip: ‘90001’}, age: 25};

Use **object destructuring** to extract the street and city from the nested address object.

Question 8:

Given the following array:

const colors = [‘red’, ‘green’, ‘blue’, ‘yellow’]

Use **array destructuring** to extract the first and second elements of the array into individual variables.

Use **array destructuring** to extract the second and fourth elements of the array into individual variables.

Question 9:

You have an array numbers:

const numbers = [1, 2, 3, 4, 5]

Use **array destructuring** to extract the last element of the array and store it in a variable called lastElement.

Question 10:

Given the following array of arrays:

`const matrix = [[1, 2], [3, 4], [5, 6]]`

Use **array destructuring** to extract the values 3 and 5 from the inner arrays.

Question 11:

How can you swap two variables using **array destructuring**? Write an example that swaps the values of two variables, a and b, without using a temporary variable.

Question 12:

Write a function createProfile that accepts an object with name, age, email, and address. Use object destructuring to extract the name and email and return a new object containing only the name and email.

Question 13:

Given the following code, use **array and object destructuring** to extract the first element of the array and the name and age from the object:

`const data = [{name: ‘John’, age: 30 }, { name: ‘Jane’, age: 25 }];`

Question 14:

You have the following array of 5 numbers:

`const nums = [10, 20, 30, 40, 50];`

Use **array destructuring** to swap the first and third elements of the array and the second and fourth elements. Then, log the modified array.

Question 15:

Write a function createProfile that accepts an object with name, age, and an array interests. Use **object destructuring** to extract name and age, and **array destructuring** to extract the first two interests. The function should return a new object with the following structure:

`{name: ‘John’, age: 25, primaryInterest: ‘Reading’, secondaryInterest: ‘Traveling’}`

Question 16:

Create an object where two properties are defined using unique Symbols. Demonstrate how you can access them and show that they do not conflict with each other even if they have the same description.

Question 17:

Create two global symbols with the same key using Symbol.for(). When retrieved from the global symbol registry, show that they are the same symbol and explain the difference between Symbol() and Symbol.for().

Question 18:

Create a custom iterable object that allows iteration through an array in reverse order. Ensure that it uses a proper iterator with the next() method and handles the done state correctly.

Example Input: const arr = [1, 2, 3, 4]

Expected Output: 4 3 2 1

Question 19:

Write an iterator that generates an infinite sequence of numbers starting from 1. Include a mechanism for stopping the iteration after a certain number of steps.

Question 20:

Implement an iterable object (e.g., a list of numbers) with a reset method. After the iteration, ensure the iterator can be reset to the initial position and used again.

Question 21:

Write a generator function that generates an infinite Fibonacci sequence. Use the generator to get the next Fibonacci number and explain how the generator pauses and resumes.

Question 22:

Create a generator that combines two separate sequences (e.g., an array of numbers and a Fibonacci sequence) and yields values from both sequences. Ensure that the generator can continue the sequence in a combined manner.

Question 23:

Create a generator function that computes the cumulative sum of an array. Return the final sum when the iteration is completed and explain how to access both the yielded values and the return value.

###@@@@###