1. Write a function that declares variables using `var`, `let`, and `const`. Attempt to reassign each variable within the same function scope, and observe the outcome. Explain any differences in behavior.

2. Create a function named `greet` that takes two parameters: `name` and `greeting`, with `greeting` having a default value of `"Hello"`. If no greeting is provided, the function should output "Hello, [name]!".

3. Using template literals, write a function that takes three parameters: `firstName`, `lastName`, and `age`, and returns a string in the format: `"My name is [firstName] [lastName] and I am [age] years old."`.

4. Write an arrow function named `add` that takes two parameters and returns their sum. Rewrite it as a one-line function without using the `return` keyword.

5. Write a function that accepts any number of arguments and returns the maximum value among them using the spread operator. Test it with varying numbers of arguments.

6. Create a function `mergeArrays` that takes two arrays as arguments and returns a single array containing all elements of the two arrays, without modifying the original arrays, using the spread operator.

7. Given an object `person` with properties `name`, `age`, and `country`, write a function that extracts `name` and `country` using object destructuring and returns them in an array.

8. Write a function that takes an array of numbers as input, and uses array destructuring to extract the first, second, and third elements, returning them in an object with properties `first`, `second`, and `third`.

9. Create a function that takes an object representing a book with properties `title`, `author`, `year`, and `publisher`. Use destructuring to extract `title` and `author`, and return a formatted string: `"Title: [title], Author: [author]"`.

10. Given an array of numbers, write a function that uses the spread operator to copy the array, then adds a new number at the end without modifying the original array.