

Assignment 1:

Write a JavaScript function that takes a number as a parameter and prints whether it's positive, negative, or zero.

```
function checkNumber(number) {  
  if (number > 0) {  
    console.log("The number is positive.");  
  } else if (number < 0) {  
    console.log("The number is negative.");  
  } else {  
    console.log("The number is zero.");  
  }  
}  
  
// Example usage:  
checkNumber(5); // Output: The number is positive.  
checkNumber(-2); // Output: The number is negative.  
checkNumber(0); // Output: The number is zero.
```

Assignment 2:

Write a JavaScript function that takes a positive integer as a parameter and calculates its factorial using a for loop. The factorial of a number N is the product of all positive integers less than or equal to N.

```
function calculateFactorial(number) {  
  // Check if the input is a positive integer  
  if (number < 0 || !Number.isInteger(number)) {  
    console.log("Please provide a positive integer.");  
    return;  
  }  
  
  let factorial = 1;  
  
  // Calculate factorial using a for loop  
  for (let i = 1; i <= number; i++) {  
    factorial *= i;  
  }  
  
  console.log(`The factorial of ${number} is: ${factorial}`);  
}  
  
// Example usage:  
calculateFactorial(5); // Output: The factorial of 5 is: 120  
calculateFactorial(0); // Output: The factorial of 0 is: 1
```

Assignment 3:

Write a JavaScript function that takes two numbers as parameters and returns the larger one.

```
function findLargerNumber(num1, num2) {  
  if (num1 > num2) {  
    return num1;  
  } else {  
    return num2;  
  }  
}
```

// Example usage:

```
let result = findLargerNumber(8, 5);  
console.log(`The larger number is: ${result}`); // Output: The larger number is: 8
```

```
result = findLargerNumber(-2, 10);  
console.log(`The larger number is: ${result}`); // Output: The larger number is: 10
```

Assignment 4:

Write a JavaScript function that takes a string as a parameter and determines whether it's a palindrome or not. A palindrome is a word, phrase, number, or other sequence of characters that reads the same forward and backward (ignoring spaces, punctuation, and capitalization).

```
function isPalindrome(str) {  
  // Remove non-alphanumeric characters and convert to lowercase  
  const cleanStr = str.replace(/[^a-zA-Z0-9]/g, "").toLowerCase();  
  
  // Compare the original string with its reverse  
  const reversedStr = cleanStr.split("").reverse().join("");  
  return cleanStr === reversedStr;  
}
```

// Example usage:

```
console.log(isPalindrome("level")); // Output: true
```

Assignment 5:

Write a JavaScript function that takes a positive integer as a parameter and prints all the prime numbers less than or equal to that integer. A prime number is a natural number greater than 1 that is not a product of two smaller natural numbers.

```
function printPrimesUpToN(n) {  
  // Function to check if a number is prime  
  function isPrime(num) {  
    if (num <= 1) return false;  
    for (let i = 2; i <= Math.sqrt(num); i++) {  
      if (num % i == 0) {  
        return false;  
      }  
    }  
    return true;  
  }  
  
  // Print prime numbers up to n  
  for (let i = 2; i <= n; i++) {  
    if (isPrime(i)) {  
      console.log(i);  
    }  
  }  
}
```

```
// Example usage:  
printPrimesUpToN(20);
```

Assignment 6:

Write a JavaScript function that simulates a simple calculator. The function should take two numbers and an operator (+, -, *, or /) as parameters and perform the corresponding operation.

```
function simpleCalculator(num1, num2, operator) {  
  switch (operator) {  
    case '+':  
      return num1 + num2;  
    case '-':  
      return num1 - num2;  
    case '*':  
      return num1 * num2;  
    case '/':  
      if (num2 !== 0) {  
        return num1 / num2;  
      }  
  }  
}
```

```

    } else {
      return "Cannot divide by zero";
    }
    default:
      return "Invalid operator";
  }
}

```

```

// Example usage:
console.log(simpleCalculator(5, 3, '+')); // Output: 8
console.log(simpleCalculator(8, 2, '-')); // Output: 6
console.log(simpleCalculator(4, 6, '*')); // Output: 24
console.log(simpleCalculator(10, 2, '/')); // Output: 5

```

Assignment 7:

Write a JavaScript function that takes a string as a parameter and counts the number of vowels (a, e, i, o, u) in the string.

```

function countVowels(str) {
  // Convert the string to lowercase to make the count case-insensitive
  const lowercasedStr = str.toLowerCase();

  // Define an array of vowels
  const vowels = ['a', 'e', 'i', 'o', 'u'];

  // Initialize a counter for vowels
  let vowelCount = 0;

  // Iterate through each character in the string
  for (let char of lowercasedStr) {
    // Check if the character is a vowel
    if (vowels.includes(char)) {
      vowelCount++;
    }
  }

  console.log(`The number of vowels in the string is: ${vowelCount}`);
}

```

```

// Example usage:
countVowels("Hello, World!"); // Output: The number of vowels in the string is: 3
countVowels("This is a sample."); // Output: The number of vowels in the string is: 6
countVowels("JavaScript"); // Output: The number of vowels in the string is: 3

```

Assignment 8:

Write a JavaScript function that takes a positive integer as a parameter and checks if it's a perfect number. A perfect number is a positive integer that is equal to the sum of its proper divisors, excluding itself.

```
function isPerfectNumber(number) {
  if (number <= 0) {
    return false; // Perfect numbers are positive integers
  }

  let sum = 0;

  // Find divisors and sum them up
  for (let i = 1; i < number; i++) {
    if (number % i === 0) {
      sum += i;
    }
  }

  // Check if the sum of divisors is equal to the original number
  return sum === number;
}

// Example usage:
console.log(isPerfectNumber(28)); // Output: true (28 is a perfect number: 1 + 2 + 4 + 7 + 14 = 28)
console.log(isPerfectNumber(6)); // Output: true (6 is a perfect number: 1 + 2 + 3 = 6)
```

Assignment 9:

Write a JavaScript function that takes a number as a parameter and prints the Fibonacci series up to that number. The Fibonacci series is a sequence of numbers in which each number is the sum of the two preceding ones.

```
function printFibonacciSeries(n) {
  if (n <= 0) {
    console.log("Please provide a positive integer.");
    return;
  }

  let fibSeries = [0, 1];

  // Generate the Fibonacci series up to n
  while (fibSeries[fibSeries.length - 1] + fibSeries[fibSeries.length - 2] <= n) {
    fibSeries.push(fibSeries[fibSeries.length - 1] + fibSeries[fibSeries.length - 2]);
  }
}
```

```
}  
  
    console.log("Fibonacci series up to", n, ":", fibSeries.join(', '));  
}
```

// Example usage:

```
printFibonacciSeries(10);
```

// Output: Fibonacci series up to 10: 0, 1, 1, 2, 3, 5, 8

```
printFibonacciSeries(20);
```

// Output: Fibonacci series up to 20: 0, 1, 1, 2, 3, 5, 8, 13

Assignment 10:

Write a JavaScript function that takes a positive integer as a parameter and prints its multiplication table up to 10.

```
function printMultiplicationTable(number) {  
    if (number <= 0) {  
        console.log("Please provide a positive integer.");  
        return;  
    }  
  
    console.log(`Multiplication table for ${number}:`);  
    for (let i = 1; i <= 10; i++) {  
        console.log(`${number} x ${i} = ${number * i}`);  
    }  
}
```

// Example usage:

```
printMultiplicationTable(5);
```

// Output:

// Multiplication table for 5:

// 5 x 1 = 5

// 5 x 2 = 10

// 5 x 3 = 15

// 5 x 4 = 20

// 5 x 5 = 25

// 5 x 6 = 30

// 5 x 7 = 35

// 5 x 8 = 40

// 5 x 9 = 45

// 5 x 10 = 50

