# Recursion

1. Write a recursive function to calculate the factorial of a given positive integer.

```
console.log(factorial(5)); // Output: 120
```

---

2. Write a recursive function to generate the nth number in the Fibonacci sequence.

```
console.log(fibonacci(7)); // Output: 13
```

---

3. Write a recursive function to calculate the value of a number raised to an exponent.

```
console.log(power(2, 3)); // Output: 8
```

---

4. Write a recursive function to calculate the sum of an array of numbers.

```
console.log(arraySum([1, 2, 3, 4, 5])); // Output: 15
```

---

5. Write a recursive function to check if a given string is a palindrome.

```
console.log(isPalindrome("racecar")); // Output: true
console.log(isPalindrome("hello")); // Output: false
```

---

6. Write a recursive function to perform a binary search on a sorted array.

```
const sortedArray = [1, 3, 5, 7, 9, 11, 13, 15, 17];
console.log(binarySearch(sortedArray, 9)); // Output: 4
console.log(binarySearch(sortedArray, 10)); // Output: -1
```

---

7. Write a recursive function to flatten an array of nested arrays.

```
const nestedArray = [1, [2, [3, 4], 5], 6];
console.log(flattenArray(nestedArray)); // Output: [1, 2, 3, 4, 5, 6]
```

8. Write a recursive function to flatten a nested object.

```
const nestedObject = {
  a: 1,
  b: {
    c: 2,
    d: {
      e: 3,
      f: 4,
    },
  },
  g: 5,
};

const flattenedObject = flattenObject(nestedObject);
console.log(flattenedObject); /* {
    "a": 1,
    "b.c": 2,
    "b.d.e": 3,
    "b.d.f": 4,
    "g": 5
} */
```

9. Write a recursive function to count the occurrences of a value in an array.

```
const numbers = [2, 3, 4, 2, 5, 2, 6, 2];
console.log(countOccurrences(numbers, 2)); // Output: 4
```

10. Write a recursive function to calculate the sum of digits of a positive integer.

```
console.log(sumOfDigits(12345)); // Output: 15
```

11. Write a recursive function to calculate the Greatest Common Divisor (GCD) of two positive integers.

```
console.log(gcd(48, 18)); // Output: 6
```

12. Write a recursive function to generate all permutations of a given string.

```
console.log(permutations("hello"));
["hello", "helol", "heoll", "hlelo", "hleol", "hlleo", "hlloe"];
```

---

13. Write a recursive function to generate all combinations of a given array.

```
const inputArray = [1, 2, 3, 4];
console.log(generateCombinations(inputArray, 2)); // Output: [[1, 2], [1, 3], [1, 4], [2, 3], [2, 4], [3, 4]]
```

---

14. Write a recursive function to find if there is a subset of a given array that adds up to a target sum.

```
const inputArray = [2, 4, 6, 8];
console.log(subsetSum(inputArray, 10)); // Output: true
```