1. Print All Subsets of a Given Set (Power Set Problem)

Example:
Input: `{1, 2, 3}`
Output: `{}`, `{1}`, `{2}`, `{3}`, `{1,2}`, `{1,3}`, `{2,3}`, `{1,2,3}`

2. Generate All Permutations of a String

Example:
Input: **`"ABC"`**
Output: **`"ABC", "ACB", "BAC", "BCA", "CAB", "CBA"`**

Hint: Swap characters and recursively fix the rest of the string.

3. Subset Sum Problem

Determine if there exists a subset of numbers that sums to a given target.

Example:
Input: `{3, 34, 4, 12, 5, 2}`, sum = 9
Output: `true` (subset {4, 5})

4. Fibonacci Sequence

Print first `n` Fibonacci numbers.

Example:
Input: `n = 5` → Output: `0 1 1 2 3`

Recurrence: `f(n) = f(n-1) + f(n-2)`

5. Reverse a String

Example:
Input: "hello" → Output: "olleh"

6. Find the Power (x^n)

Example:
Input: x = 2, n = 5 → Output: 32

Logic:
If n even → power(x, n/2) * power(x, n/2)
If n odd → x * power(x, n-1)

6. Count Digits of a Number

Example:
Input: 12345 → Output: 5

7. Binary Search using Recursion

Example:
Input: [1, 2, 3, 4, 5, 6], search = 4 → Output: Found at index 3

Logic:
Check mid → if not found, recurse left or right half.

8. Find Maximum Element in Array

Example:
Input: [3, 9, 1, 7, 5] → **Output:** 9

Hint: max = max(arr[n-1], findMax(arr, n-1))

9. Count Occurrences of a Number in Array

Example:
Input: [1,2,3,2,4,2], **target=2** → **Output:** 3

Hint:
Increment count when match found, then recurse.

10. Reverse an Array using Recursion

Example:
Input: [1, 2, 3, 4, 5] → **Output:** [5, 4, 3, 2, 1]