Recursive Basic Problem

🌱 Stage 1: Baby Steps (get comfortable with flow)

1. Print numbers from `n` down to `1` and `1` up to `n`.
2. Factorial of `n`.
3. Sum of first `n` natural numbers.
4. Power of a number (`x^n`).
5. Sum of digits of a number.
6. Product of digits of a number.

🌿 Stage 2: String & Array Manipulation (trust the clone)

7. Reverse a string.
8. Check if a string is palindrome.
9. Count occurrences of a character in a string.
10. Remove all occurrences of a character from a string.
11. Replace all `"pi"` with `"3.14"` in a string (classic).
12. Find index of first/last occurrence of an element in an array.
13. Print all indices of an element in an array.

🌳 Stage 3: Classic Recursive Puzzles (pattern spotting)

14. Fibonacci numbers.
15. Climb stairs problem (ways to reach n steps if 1 or 2 steps allowed).
16. Generate all subsequences of a string.
17. Generate all subsets of an array.
18. Print all permutations of a string.
19. Print keypad combinations (like old Nokia keypad, e.g. 2→abc, 3→def).
20. Tower of Hanoi.

🌋 Stage 4: Thinking Like Divide & Conquer (recursion as strategy)

Now recursion starts solving bigger problems:

21. Binary search (recursive version).
22. Merge sort.
23. Quick sort.
24. Count inversions in an array (still recursion, no trees).

🎯 Outcome

If you nail this list, you'll have:

- Control over base cases.
- Intuition for shrinking problems.
- Confidence in string/array recursion.
- Comfort with divide & conquer recursion.