Recursion

Guidance for Solutions:

For each of these exercises, start by identifying the base case(s) for your recursive function — these are the conditions under which the function will stop calling itself. Then, determine how each function should break down its problem into smaller subproblems, eventually reaching a base case. Remember to test your functions with various inputs to ensure they work as expected.

1. Sum of Natural Numbers

Write a recursive function that takes a number `n` and returns the sum of all natural numbers up to `n`. For example, if the input is `5`, the function should return `15` (i.e., `5 + 4 + 3 + 2 + 1`).

2. Reverse a String

Implement a recursive function that takes a string and returns its reverse. For example, given the string `"hello"`, the function should return `"olleh"`.

Certainly! Here are three simpler exercises focused on recursion. These exercises are designed to introduce the concept of recursion with more straightforward problems than the sum of natural numbers and reversing a string.

Exercise 3: Count Down

Write a recursive function that prints numbers from `n` down to 1. For example, if `n` is `5`, the function should print `5, 4, 3, 2, 1`.

Exercise 4: Print an Array

Write a recursive function that prints all the elements of an array, one element per line. You can assume the array contains only numbers or strings.