1. A value x decreases by squaring itself and dividing by 2 each day for n days. Write a recursive function to calculate the total accumulated value.  
   **Calculation:**

Day 1: 8

Day 2: (8²)/2 = 64/2 = 32

Day 3: (32²)/2 = 1024/2 = 512

Total = 8 + 32 + 512 = 552

1. A city has p population. Each year, it declines by 7% and 100 people emigrate. Find total population over n years.
2. A file of size s MB downloads s/2 MB in first second, then s/4, then s/8... until n seconds. Find total data downloaded.
3. A tree grows by g% each year, but also loses 2 cm due to weather. Find height after n years.
4. A tank contains w liters of water. Each hour, 10% leaks and a fixed 3 liters are drained for usage. Calculate total water lost over n hours.
5. A patient takes a dose of m mg. Every hour, 20% is absorbed, and 10 mg is lost through metabolism. Total absorbed in n hours?
6. Given a number, recursively add its digits until a single digit remains.
7. Check if a string is a palindrome using recursion.
8. Print moves for solving Tower of Hanoi with n disks.
9. Print binary of a number using recursion.
10. Find sum of first n numbers **excluding** multiples of 3.
11. Use recursion to find the max in an array.
12. Given an array, is there a subset that sums to target?
13. You are at (0,0) in an m x n grid. You can only move **right** or **down**. Count how many paths to reach (m-1, n-1).
14. Count the number of zero digits in a number (e.g., 102030 has 3 zeroes).
15. Given an array of integers, write a function using only pointer arithmetic to reverse the array in-place (without using array indexing).
16. Write a program that demonstrates a dangling pointer scenario and then modifies the code to safely avoid the problem using smart pointers or best practices.
17. Create an array of function pointers that each point to arithmetic operations: add, subtract, multiply, divide. Allow the user to select an operation using an index and perform it on two numbers.
18. Write a program that intentionally causes a memory leak using new and delete, then fix it using proper memory management practices. Include comments showing the incorrect and corrected parts.
19. Dynamically allocate a 2D matrix of integers using pointers (not vector or array), populate it, print it, and then free all memory properly.
20. Find the maximum element in a 2D array recursively using pointer arithmetic.
21. Create a C++ program to generate a magic square of odd order (n x n) using pointers. A magic square is a square matrix in which the sum of every row, column, and both main diagonals is the same. The program should:
    1. Dynamically allocate a 2D array using pointers.
    2. Fill it using the Siamese method (only works for odd n).
    3. Display the square and deallocate memory properly.

**Constraints**:

* n must be an odd number ≥ 3.
* Use dynamic memory allocation (new and delete).
* Use pointer arithmetic wherever possible.