* Discuss the time complexity of your recursive algorithm.

This function calls itself once per month, from months down to 0.So, for n months, it makes n recursive calls.

Each call performs constant work (just a multiplication), and makes one recursive call:

Time Complexity: O(n)

Because it makes one call per month

Each call does a small amount of work (O(1))

* Explain how to optimize the recursive solution to avoid excessive computation.

**Use Iteration**

If recursion isn't essential, replace it with a loop to improve space efficiency**.**

Time Complexity: O(n)

Space Complexity: O(1**)**

**Memoization**

If the recursive function recomputes the same values (like in Fibonacci), store previous results in a data structure like a HashMap or array**.**

Reduces exponential time (O(2ⁿ)) to linear time O(n)