Step 1: Understand the Recursive Algorithm

## Explain the concept of recursion and how it can simplify certain problems.

Recursion is a programming technique where a function calls itself to solve a smaller part of the same problem. It continues until it reaches a base case, which stops further recursive calls.

Each recursive call is stored in the call stack, and the function works like a tree — breaking the problem into smaller subproblems.

Step 4: Analysis

## Discuss the time complexity of your recursive algorithm.

The function calls itself n number of times for n years, so the time complexity will be O(n).

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

Instead of using recursion, we can rewrite the problem using a bottom-up approach. This means we start from the base case and build the solution step by step using a loop and an array. This method completely avoids the overhead of recursive calls and is very efficient for problems that have a clear base case and build up in one direction — like calculating future values over time.