**Exercise 7: Financial Forecasting**

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

**Recursion** is a technique where a function calls itself to solve smaller instances of the same problem.  
It is especially useful for problems that have a natural repetitive structure (for example: computing factorial, Fibonacci, traversing trees, or financial growth across years).

In this exercise, recursion simplifies the process of forecasting by naturally representing "future years" as smaller sub-problems.

* **Discuss the time complexity of your recursive algorithm.**

**Time Complexity:**  
For this recursive solution:

* At each level, the method makes **1 recursive call** and does **O(1)** work.
* The number of recursive calls is **equal to the number of years**.
* Therefore: **Time complexity = O(n)** where n = years.
* **Explain how to optimize the recursive solution to avoid excessive computation.**

**Optimization:**

* This problem is simple linear recursion — it does not suffer from exponential recomputation (like Fibonacci).
* Still, to avoid excessive stack usage for large years, we can:
  + Use **iteration** instead of recursion (iterative loop is safer for large input)
  + Or use **tail recursion** if supported by the compiler.