**Exercise 7: Financial Forecasting Scenario**

**Step 1: Understand Recursive Algorithms (Basics)**

* Recursion is when a method calls itself to solve smaller parts of a problem.
* It has:
  + Base case: when to stop recursion.
  + Recursive case: the method keeps calling itself.
* Useful for problems with repetitive or nested structure.
* In financial forecasting, each year's value depends on the previous year's value — making recursion a good fit.

**Step 4: Analysis**

In my recursive method, each function call calculates the future value by reducing the number of years by 1, so it runs once for each year. That means the time complexity is O(n), where n is the number of years.

But the issue is, every time the method calls itself, it holds that call in memory until it reaches the base case. So, the memory used also grows with the number of years — that’s O(n) space.

Now if the number of years is too high, it might even crash with a stack overflow error. That’s why the recursive method isn’t suitable for large inputs.

To avoid this, I’ve also written an iterative version. It does the same job but uses a simple loop, so it doesn’t use extra stack space and works faster in practice.