**DSA Exercise 7: Financial Forecasting**

4. Analysis:

**\* Basic Recursive:**

- Time Complexity: O(n)

- Space Complexity: O(n) (due to call stack)

**\* Memoized Recursive:**

- Time Complexity: O(n)

- Space Complexity: O(n) (due to memo array)

**\* Optimization Reason:**

- Prevents repeated computation of the same years' future values.

- Greatly improves performance for large `years` value.

**\* My Take:**

I first implemented the future value calculation using a simple recursive function. It calls itself once for each year, reducing the year count until it reaches zero. So, for 5 years, it makes 5 recursive calls. This was straightforward, but I realized that if the number of years is large, it could become inefficient due to the growing call stack.

To improve it, I added memoization by storing already computed values in an array. This way, if a value is already calculated for a particular year, the function doesn't compute it again—it just looks it up. This made the program faster and avoided unnecessary repeated calculations. Both versions technically have O(n) time complexity, but the memoized one is more efficient in practice, especially for larger inputs.