**Exercise 7: Financial Forecasting**

**Scenario:**

You are developing a financial forecasting tool that predicts future values based on past data.

**Steps:**

1. **Understand Recursive Algorithms:**
   * Explain the concept of recursion and how it can simplify certain problems.

Recursion is a programming technique where a function calls itself to solve smaller instances of the same problem. This approach simplifies problems that can be broken down into similar subproblems. Each recursive function has a base case, which stops the recursion, and a recursive case, which continues it. For example, the factorial function n! can be defined recursively as n! = n (n-1)! with 0! = 1. Recursion is particularly useful in tasks like tree traversals, sorting algorithms like Merge Sort, and mathematical computations, as it can make the code more intuitive and easier to understand by directly reflecting the problem's structure.

1. **Analysis:**
   * Discuss the time complexity of your recursive algorithm.

The time complexity of the calValue method is O(n)O(n)O(n), where nnn is the number of periods. This is because the method contains a loop that iterates exactly periods times, performing a constant-time operation (multiplication and addition) in each iteration. Thus, the time complexity is linear with respect to the number of periods.

In the context of the entire program:

•Reading inputs from the user is O(1)O(1)O(1) for each input operation.

•Calculating the future value is O(n)O(n)O(n).

•Printing the result is O(1)O(1)O(1).

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

Use techniques like Memoization or Dynamic Programming to store intermediate results and avoid redundant calculations, reducing time complexity.