**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.
2. **Setup:**
   * Create a method to calculate the future value using a recursive approach.
3. **Implementation:**
   * Implement a recursive algorithm to predict future values based on past growth rates.
4. **Analysis:**
   * Discuss the time complexity of your recursive algorithm.
   * Explain how to optimize the recursive solution to avoid excessive computation.

using System;

public class FinancialForecast

{

public static double CalculateFutureValue(double initialAmount, double growthRate, int years)

{

if (years == 0)

return initialAmount;

return CalculateFutureValue(initialAmount, growthRate, years - 1) \* (1 + growthRate);

}

public static double CalculateFutureValueMemo(double initialAmount, double growthRate, int years, double[] memo)

{

if (years == 0)

return initialAmount;

if (memo[years] != 0)

return memo[years];

memo[years] = CalculateFutureValueMemo(initialAmount, growthRate, years - 1, memo) \* (1 + growthRate);

return memo[years];

}

public static void Main()

{

double initialInvestment = 10000;

double annualGrowthRate = 0.08;

int forecastYears = 5;

Console.WriteLine("--- Recursive Calculation ---");

double futureValue = CalculateFutureValue(initialInvestment, annualGrowthRate, forecastYears);

Console.WriteLine($"Future Value after {forecastYears} years: ₹{futureValue:F2}");

Console.WriteLine("\n--- Memoized Calculation ---");

double[] memo = new double[forecastYears + 1];

double futureValueMemo = CalculateFutureValueMemo(initialInvestment, annualGrowthRate, forecastYears, memo);

Console.WriteLine($"Future Value after {forecastYears} years: ₹{futureValueMemo:F2}");

}

}

