***Week 1\_4: MANDATORY HANDSON EXERCISE (ALGORITHMS\_DATA\_STRUCTURE)***

**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.

***CODE:***

//FinancialForecaster.cs

using System;

class FinancialForecaster

{

    // Recursive approach

    public static double CalculateFutureValue(double currentValue, double growthRate, int periods)

    {

        if (periods <= 0)

            return currentValue;

        double nextValue = currentValue \* (1 + growthRate);

        return CalculateFutureValue(nextValue, growthRate, periods - 1);

    }

    // Optimized iterative approach

    public static double CalculateFutureValueOptimized(double currentValue, double growthRate, int periods)

    {

        return currentValue \* Math.Pow(1 + growthRate, periods);

    }

    public static void Main(string[] args)

    {

        double initialValue = 25000;

        double growthRate = 0.10;

        int forecastPeriods = 11;

        // Using recursive method

        double futureValueRecursive = CalculateFutureValue(initialValue, growthRate, forecastPeriods);

        // Using optimized iterative method

        double futureValueOptimized = CalculateFutureValueOptimized(initialValue, growthRate, forecastPeriods);

        Console.WriteLine($"Initial investment: ${initialValue}");

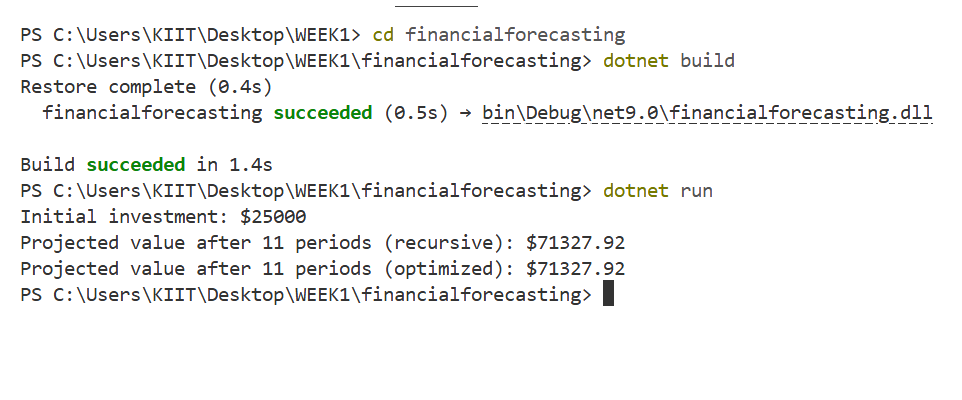
        Console.WriteLine($"Projected value after {forecastPeriods} periods (recursive): ${futureValueRecursive:F2}");

        Console.WriteLine($"Projected value after {forecastPeriods} periods (optimized): ${futureValueOptimized:F2}");

    }

}

***OUTPUT:***

******