**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:**

using System;

class Forecast

{

public static decimal PredictValue(decimal initialValue, decimal growthRate, int years)

{

if (years == 0) return initialValue;

return PredictValue(initialValue, growthRate, years - 1) \* (1 + growthRate);

}

}

class Program7

{

static void Main()

{

decimal baseValue = 1000;

decimal growthRate = 0.05m; // 5%

int years = 5;

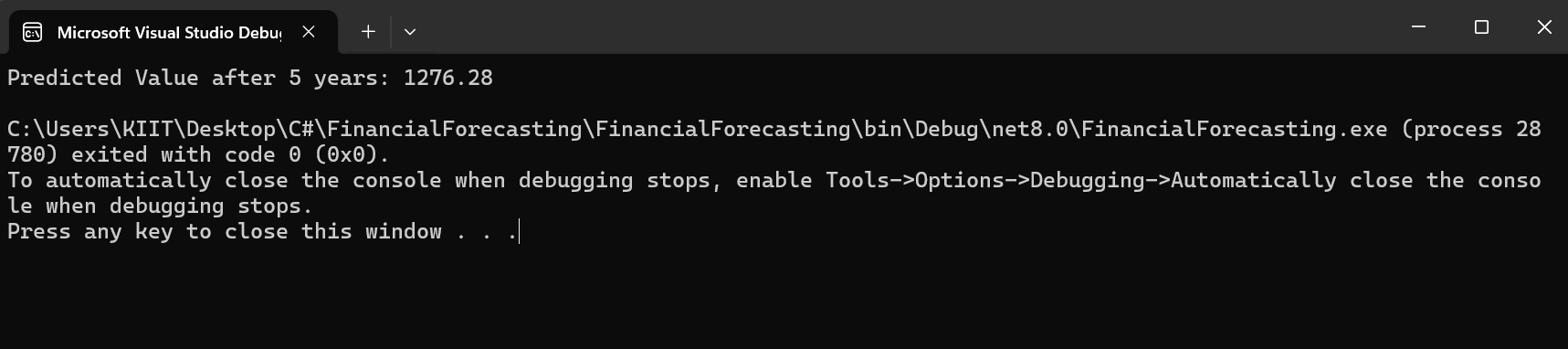
decimal futureValue = Forecast.PredictValue(baseValue, growthRate, years);

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

}

}

**OUTPUT:**



**ANALYSIS:**

**Time Complexity:**  
O(n), where n = number of years (recursive calls made).

**Optimization of recursion:**  
M**emoization** or convert to **iteration** to avoid redundant calls and stack overflow can be done.