**Exercise 7: Financial Forecasting**

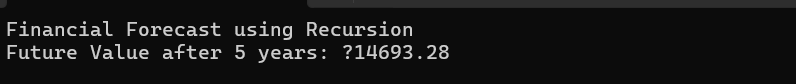
**Scenario:**

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.

**Output-**

****

**Code-**

using System;

class FinancialForecast

{

static void Main(string[] args)

{

double presentValue = 10000;

double growthRate = 0.08;

int years = 5;

Console.WriteLine("Financial Forecast using Recursion");

double futureValue = CalculateFutureValue(presentValue, growthRate, years);

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

}

static double CalculateFutureValue(double pv, double rate, int n)

{

if (n == 0)

return pv;

return (1 + rate) \* CalculateFutureValue(pv, rate, n - 1);

}

}

**Time complexity :** O(n)

**Space complexity :** O(n)

1. Explain how to optimize the recursive solution to avoid excessive computation.

**->** We can use an **iterative approach** to reduce stack usage and improve performance.