# Exercise 7: Financial Forecasting (.NET)

## What I Learned

In this task, I built a financial forecasting algorithm using recursion in .NET. I understood how recursion simplifies problems that involve repeated calculations over time. I created a method to predict future financial values based on growth rate and number of years.

## Concepts Covered

- Recursion in C#

- Predictive modeling

- Time complexity of recursive algorithms

- Optimization using memoization or iteration

## C# Code

using System;  
  
class FinancialForecast  
{  
 public static double PredictFutureValue(double presentValue, double growthRate, int years)  
 {  
 if (years == 0)  
 return presentValue;  
 return (1 + growthRate) \* PredictFutureValue(presentValue, growthRate, years - 1);  
 }  
  
 static void Main(string[] args)  
 {  
 double presentValue = 1000;  
 double growthRate = 0.1;  
 int years = 5;  
  
 double futureValue = PredictFutureValue(presentValue, growthRate, years);  
 Console.WriteLine("Predicted Future Value after " + years + " years: " + futureValue);  
 }  
}

## output

## 

## Conclusion

I used recursion to calculate the future value based on compound growth. The recursive solution is clear and simple but can be inefficient for large inputs. To optimize it, I can use memoization or convert it to an iterative approach to avoid redundant calculations.