**WEEK 1   
  
 ALGORITHMS AND DATA STRUCTURES  
  
 EXERCISE 7 - FINANCIAL FORECASTING**  
  
  
**STEP 1 - Forecaster.cs :**

namespace FinancialForecastingExample

{

    public class Forecaster

    {

        public static double PredictFutureValue(double initialValue, double growthRate, int years)

        {

            if (years == 0)

                return initialValue;

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

        }

        public static double PredictFutureValueMemo(double initialValue, double growthRate, int years, double[] memo)

        {

            if (years == 0)

                return initialValue;

            if (memo[years] != 0)

                return memo[years];

            memo[years] = PredictFutureValueMemo(initialValue, growthRate, years - 1, memo) \* (1 + growthRate);

            return memo[years];

        }

    }

}

**STEP 2 - Program.cs :**

using System;

namespace FinancialForecastingExample

{

    class Program

    {

        static void Main(string[] args)

        {

            double initialValue = 1000;

            double growthRate = 0.05;

            int years = 5;

            double result = Forecaster.PredictFutureValue(initialValue, growthRate, years);

            Console.WriteLine($"Recursive: Value after {years} years = {result:F2}");

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

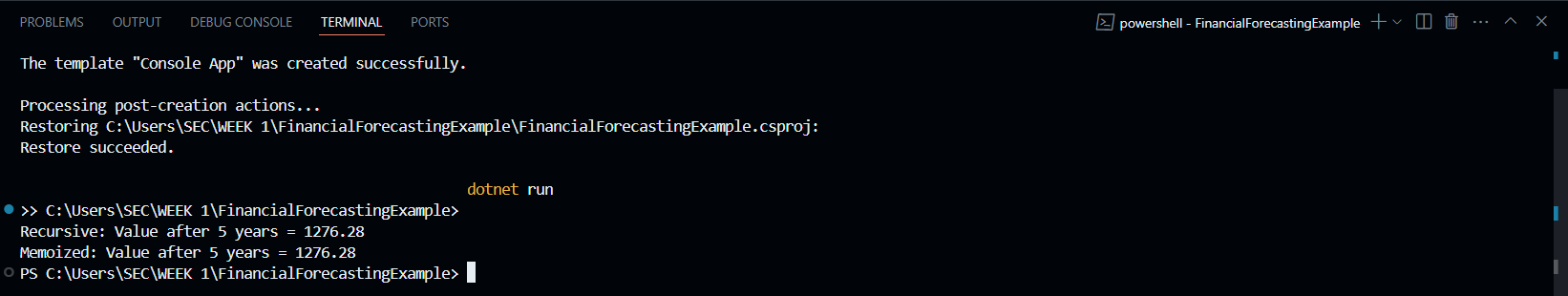
            double memoResult = Forecaster.PredictFutureValueMemo(initialValue, growthRate, years, memo);

            Console.WriteLine($"Memoized: Value after {years} years = {memoResult:F2}");

        }

    }

}

**OUTPUT :   
  
  
**