**Exercise 4: Financial Forecasting**

**Sol:**

**File Name:** Program.cs

**Code:**

*using* System;

*class* Program

{

*static* *double* CalculateFutureValue(*double* *presentValue*, *double* *rate*, *int* *years*)

    {

*if* (*years* == 0)

*return* *presentValue*;

*return* CalculateFutureValue(*presentValue*, *rate*, *years* - 1) \* (1 + *rate*);

    }

*static* *double* CalculateFutureValueMemo(*double* *presentValue*, *double* *rate*, *int* *years*, *double*[] *memo*)

    {

*if* (*years* == 0)

*return* *presentValue*;

*if* (*memo*[*years*] != 0)

*return* *memo*[*years*];

*memo*[*years*] = CalculateFutureValueMemo(*presentValue*, *rate*, *years* - 1, *memo*) \* (1 + *rate*);

*return* *memo*[*years*];

    }

*static* *void* Main(*string*[] *args*)

    {

*double* presentValue = 2700;

*double* growthRate = 0.30;

*int* years = 15;

        Console.WriteLine("📈 Future Value Using Recursion:");

*double* result = CalculateFutureValue(presentValue, growthRate, years);

        Console.WriteLine($"After {years} years: {result:C}");

        Console.WriteLine("\n🚀 Future Value Using Memoized Recursion:");

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

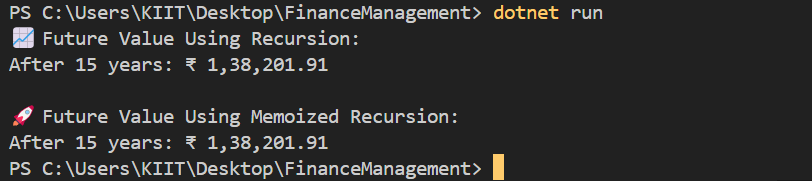
*double* resultMemo = CalculateFutureValueMemo(presentValue, growthRate, years, memo);

        Console.WriteLine($"After {years} years: {resultMemo:C}");

    }

}

**OUTPUT**

****