**Financial Forecasting – Recursive Algorithm Analysis**

**Objective**

To forecast future financial values using a recursive algorithm based on:

Initial value

Annual growth rate

Forecasting period (in years)

**Code Implementation**

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

{

if (years == 0)

return currentValue;

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

}

**Sample Output**

Input:

Current Value = 10,000

Growth Rate = 10% (0.1)

Years = 3

Output:

Forecasted value after 3 years: 13310.00

Calculation steps:

Year 1: 10000 × 1.1 = 11000

Year 2: 11000 × 1.1 = 12100

Year 3: 12100 × 1.1 = 13310

**Time Complexity Analysis**

Time Complexity: O(n) where n is the number of years

The function makes n recursive calls to calculate the final result

**Optimization Discussion**

While recursion simplifies the logic, it may become inefficient for large n. To optimize:

Convert to Iterative Approach (uses loop instead of function calls)

Memoization (if the same calculations are repeated)

**Iterative Alternative**

public static double PredictFutureValueIterative(double currentValue, double growthRate, int years)

{

for (int i = 0; i < years; i++)

currentValue \*= (1 + growthRate);

return currentValue;

}

**Conclusion**

Recursion offers a clean and elegant solution to forecast future financial values. However, for large datasets or performance-critical applications, using an iterative approach is more efficient.