# Financial Forecasting Report

## 1. Objective

To implement a recursive approach to predict future financial values based on past data and a constant annual growth rate.

## 2. Understanding Recursive Algorithms

Recursion is a programming technique where a function calls itself to solve smaller instances of a problem. It helps simplify problems like future value prediction, where each year’s value depends on the previous year.

## 3. Setup

Create a class named FinancialForecast with a method to calculate future value using recursion.

## 4. Implementation

public class FinancialForecast {  
  
 public static double predictFutureValue(double initialAmount, double growthRate, int years) {  
 if (years == 0) {  
 return initialAmount;  
 }  
 return predictFutureValue(initialAmount, growthRate, years - 1) \* (1 + growthRate);  
 }  
  
 public static void main(String[] args) {  
 double initialInvestment = 10000; // ₹10,000  
 double annualGrowthRate = 0.10; // 10%  
 int forecastYears = 5;  
  
 double futureValue = predictFutureValue(initialInvestment, annualGrowthRate, forecastYears);  
  
 System.out.printf("Future value after %d years: ₹%.2f\n", forecastYears, futureValue);  
 }  
}

## 5. Sample Output

Future value after 5 years: ₹16105.10

## 6. Analysis

• Time Complexity: O(n), where n is the number of years.  
• Space Complexity: O(n), due to recursive call stack.  
• Optimization: For simple growth rate predictions, recursion is fine. For more complex calculations or large 'n', use iteration or memoization to reduce call stack depth.

## 7. Iterative Alternative (Optimized)

public static double predictIterative(double amount, double rate, int years) {  
 for (int i = 0; i < years; i++) {  
 amount \*= (1 + rate);  
 }  
 return amount;  
}