**Exercise 7: Financial Forecasting**

**Program:**

**1. FinancialForecast.java**

public class FinancialForecast {

    public static double futureValue(double presentValue, double growthRate, int years) {

        if (years == 0) {

            return presentValue;

        }

        return futureValue(presentValue \* (1 + growthRate), growthRate, years - 1);

    }

    public static void main(String[] args) {

        double presentValue = 10000.0;

        double annualGrowthRate = 0.08;

        int years = 5;

        double predictedValue = futureValue(presentValue, annualGrowthRate, years);

        System.out.printf("Predicted value after %d years: %.2f\n", years, predictedValue);

    }

}

**Questions:**

**1) Explain the concept of recursion and how it can simplify certain problems.** Recursion is when a function calls itself to solve a smaller part of the problem. It makes solving problems easier by breaking them into simple steps. This helps in tasks that repeat, like calculating values over time or doing the same action many times.

**2) Discuss the time complexity of your recursive algorithm.** The time complexity of the recursive algorithm is O(n), where n is the number of years. This is because the function runs once for each year until it reaches zero, making one call per year.

**3) Explain how to optimize the recursive solution to avoid excessive computation.** We can optimize the recursive solution by using a simple loop instead of recursion. Loops use less memory and are faster, especially when the number of steps is large. Also, adding a proper base case helps stop the function at the right time and avoids unnecessary calls. This makes the program work more efficiently.

**Output:**

