Exercise 7: Financial Forecasting – Recursive Algorithm

1. Understanding Recursive Algorithms

Definition : Recursion is a method where a function calls itself to solve smaller instances of the same problem.

Example :
A classic example is factorial : factorial(n) = n * factorial(n - 1)
with base case : factorial(0) = 1.

This helps break down problems into simpler subproblems, making logic cleaner.

2. Setup

In this forecasting tool:

- The user provides:
  - Initial investment amount
  - Annual growth rate (in %)
  - Target year
  - CurrentYear (Taken using LocalDateTime.now().getYear())
- The method RecurssivePrediction() is created to predict the future value.

3. Implementation

This is the **base case**. When the target year equals the current year, the recursion ends

```
if (targetYear == currentYear)
    return amount;
```

For other years, it reduces the target year by 1 and applies the growth:

```
return RecurssivePrediction(targetYear - 1, currentYear, amount, rate) * (1 + rate / 100);
```

This continues until the base year is reached.

4. Analysis

- **Time Complexity**: O(n)
- **Space Complexity**: O(n)
- **Optimization (optional)** : This recursive solution is clean but not optimal for very large year gaps. **Memoization** can be used to store intermediate results and avoid recalculating
- **Map<Integer, Double> memo = new HashMap<>()**.(We use this to store values

Conclusion : This approach demonstrates how recursion simplifies forecasting logic by breaking the problem down year by year.