**Exercise 7: Financial Forecasting**

**Understanding Recursive Algorithms:**

Recursion is a programming technique where a method calls itself to solve a smaller version of the original problem. It simplifies problems by breaking them into base cases and recursive cases. In forecasting, recursion can model future values based on repeating patterns, such as annual growth.

**Program:**

public class Financial {

public static double calcRec(int n, double val, double rate) {

if (n == 0) {

return val;

}

return calcRec(n - 1, val, rate) \* (1 + rate);

}

public static double calcIter(int n, double val, double rate) {

double res = val;

for (int i = 1; i <= n; i++) {

res \*= (1 + rate);

}

return res;

}

public static void main(String[] args) {

double val = 1000.0;

double rate = 0.10;

int n = 5;

double ans1 = calcRec(n, val, rate);

System.out.printf("Recursive - Value after %d years: %.2f%n", n, ans1);

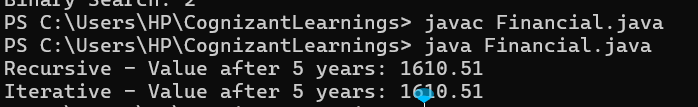
double ans2 = calcIter(n, val, rate);

System.out.printf("Iterative - Value after %d years: %.2f%n", n, ans2);

}

}

**Output:**

****

**Analysis**

The time complexity of this recursive approach is O(n), where n is the number of years. This is because the method calls itself once for each year until it reaches the base case.