Exercise 7: Financial Forecasting

# 1. Understanding Recursive Algorithms

Recursion is a method in programming where a function calls itself to solve a smaller instance of the same problem. It's especially useful for problems that can be broken down into smaller, similar problems. In financial forecasting, recursion helps in calculating compound growth over several time periods without writing repeated logic.

# 2. Setup

We define a function `predictFutureValue()` that takes the current amount, the annual growth rate, and the number of years. It recursively calculates the value for each year by applying the growth rate.

# 3. Recursive Algorithm Implementation

If the number of years is 0, we simply return the current value. Otherwise, we recursively multiply the value by (1 + growthRate) and reduce the number of years by 1.

This continues until we reach the base case (years == 0).

# 4. Time Complexity and Optimization

Time Complexity: O(n), where n is the number of years.  
Space Complexity: O(n) due to the recursive call stack.  
To optimize, we can convert the recursion into an iterative loop, especially if we are working with a large number of years.

# 5. Conclusion

This recursive approach is clean and easy to understand for small forecasting tasks. However, for large datasets or real-time applications, an iterative or memoized approach is preferred to avoid stack overflow and improve efficiency.