**Exercise 4: Financial Forecasting**  
  
**1. Explain the concept of recursion and how it can simplify certain problems?**  
Recursion is a programming technique where a function calls itself to solve a smaller instance of the same problem.  
A recursive function must have:

1. Base Case – The stopping condition. Prevents infinite recursion.
2. Recursive Case – The function calls itself to approach the base case.

Example: Factorial of a Number (n!)  
def factorial(n):

if n == 0:

return 1 # Base case

else:

return n \* factorial(n - 1) # Recursive case  
  
Recursion Simplifies Problems:  
  
1. Divides Big Problems into Smaller Ones  
Problems like tree traversal, combinatorics, or maze solving naturally break into smaller, similar subproblems.  
2. Reduces Code Complexity  
Without recursion, tasks like computing factorial or traversing nested structures would require loops + stacks, making the code longer and harder to read.  
3. Elegant and Clean Logic  
Recursion mirrors mathematical definitions (like factorials, permutations, trees), making code easier to understand and debug.