# Module 8.1 Advanced Functions: Coding Questions

## Question 1: Develop a function that automates data validation for user inputs in a web form. It should check for empty fields, correct data types, and range values.

Hint: Utilize functions for each type of validation and loop through fields to apply these functions.

## Question 2: Create a high-order function that logs the execution time of any function passed to it, returning both the function's result and the execution time.

Hint: Use decorators to wrap the original function, utilizing time module to calculate execution time.

## Question 3: Write a function that dynamically generates SQL queries based on provided parameters such as table name, fields, and conditions.

Hint: Build the query string using conditional statements and loops based on the function's arguments.

## Question 4: Implement a caching mechanism for a function that performs resource-intensive calculations, to reuse previously computed results.

Hint: Use a dictionary to store previous results, checking the cache before performing the calculation.

## Question 5: Design a function that processes and transforms customer feedback text into structured data, identifying key themes and sentiment.

Hint: Apply string methods and possibly import libraries for sentiment analysis, organizing results in a dictionary.

## Question 6: Construct a function that simulates stock market transactions, calculating the portfolio's total value based on current market prices.

Hint: Loop through the portfolio items, fetching current prices (mocked or via API), and calculate the total value.

## Question 7: Develop a recursive function to solve the Tower of Hanoi problem for any number of disks.

Hint: Implement the classic recursive solution, carefully managing the base case and recursive calls.

## Question 8: Create a generator function that produces an infinite sequence of Fibonacci numbers.

Hint: Use a while True loop in the generator, yielding the next number in the sequence each time.

## Question 9: Write a function that accepts a list of tasks (functions) and a list of arguments, then executes each task with the corresponding arguments in parallel.

Hint: Consider using threading or multiprocessing modules to execute tasks in parallel.

## Question 10: Implement a function that automatically retries a failed network request with exponential backoff.

Hint: Use a loop with sleep intervals that increase exponentially upon each failure, until a success condition is met or a maximum number of retries is reached.