# Practice Questions on Python Decorators, Logging, Authorization, and Lambda Functions (Easy to Intermediate Level)

## Decorators (10 Questions)

1. Write a decorator to print 'Function started' before a function runs and 'Function ended' after it runs.
2. Create a decorator that multiplies the return value of a function by 2.
3. Write a decorator that logs the name of the function being called.
4. Create a decorator to check if the function is called with exactly 2 arguments.
5. Write a decorator that counts and prints how many times the function has been called.
6. Write a decorator that restricts a function from running more than once.
7. Write a decorator to check if a user is authenticated (pass is\_authenticated=True as a keyword argument).
8. Create a decorator with arguments that repeats the function n times.
9. Write a decorator that measures the execution time of a function.
10. Write a decorator that modifies a function to return None if it raises any exception.

## Logging (10 Questions)

1. Write a simple logging function that logs to the console when a function is called.
2. Create a logger using the logging module that logs messages to a file.
3. Write a function that logs both arguments and return values.
4. Add logging to a calculator function that logs each operation.
5. Configure a logger to log only warnings and errors.
6. Write a decorator that logs the execution time of a function.
7. Use logging to log uncaught exceptions in a function.
8. Write a logger that logs the user IP address when a function is called.
9. Create a rotating file logger using logging.handlers.
10. Write a decorator that logs the start and end time of a data processing function.

## Authorization (10 Questions)

1. Write an authorization decorator that allows only users with the role 'admin'.
2. Create a function that checks if the user has permission 'view\_reports'.
3. Write a decorator that blocks a function call if the user’s status is 'inactive'.
4. Implement a decorator that checks if a user email is in the authorized list.
5. Write a decorator that checks if the user token is valid.
6. Write a decorator that allows only users with subscription 'premium' to access a function.
7. Simulate an API call where only authenticated users can access data using a decorator.
8. Write a decorator that denies access if the user tries to access outside working hours.
9. Write a decorator that logs unauthorized access attempts.
10. Create a decorator that restricts access to functions based on country code.

## Lambda Functions (10 Questions)

1. Write a lambda function to square a number.
2. Use filter() with lambda to get all even numbers from a list.
3. Use map() with lambda to get the cube of all numbers in a list.
4. Write a lambda function to check if a string is a palindrome.
5. Sort a list of tuples based on the second element using lambda.
6. Use reduce() with lambda to calculate the factorial of a number.
7. Write a lambda function to check if a number is divisible by both 3 and 5.
8. Use map() and lambda to convert a list of strings to uppercase.
9. Use lambda inside sorted() to sort a list of dictionaries by the 'age' key.
10. Write a lambda function that returns the maximum of two numbers.