PRACTICE TEST: Python Logging & Schedule for Data Analytics Automation

PART A: LOGGING

Q1. Create a logger writing INFO logs to pipeline.log with timestamp, level, message.
Q2. Log to file and console simultaneously.
Q3. Log start, success, and error for loading sales.csv.
Q4. Log DEBUG rows count, INFO transform success, WARNING missing values, ERROR division error.
Q5. Disable DEBUG logs globally.
Q6. Separate info.log and error.log using handlers.
Q7. Log API request lifecycle and timing.
Q8. Create reusable get_logger(name) avoiding duplicates.
Q9. Log ETL steps with stack trace on failure.
Q10. Log in JSON format.
Q11. Implement log rotation (1MB, 5 backups).
Q12. Daily log files with date in filename.
Q13. Log execution time using decorator.
Q14. Add correlation ID to every log.
Q15. Master log with runtime and exit status.

PART B: SCHEDULE

Q16. Schedule daily job at 08:00.
Q17. Two jobs: 10 min refresh, daily report.
Q18. Weekday-only job.
Q19. Hourly job between 9 AM and 6 PM.
Q20. Stop job after 5 runs.
Q21. Scheduled ETL with logging.
Q22. Random failure job with retries.
Q23. Skip Sunday execution.
Q24. Dynamic interval scheduling.
Q25. Pause and resume job.

PART C: FINAL CHALLENGE

Q31. Build full analytics automation using logging + schedule with rotation, timing, retries, correlation ID.