SHUTIL PRACTICE TEST – DATA ANALYTICS AUTOMATION

1. Copy a single CSV file from raw_data/ to processed_data/.
2. Copy all files from daily_reports/ into monthly_backup/ while preserving metadata.
3. Copy only .csv files from input/ to csv_archive/.
4. Copy a file and rename it with today's date appended.
5. Copy sales_data/ directory to backup/sales_data_backup/.
6. Replace an existing backup folder completely with a fresh copy.
7. Copy a file only if it does not already exist in destination.
8. Create versioned backups like backup_v1, backup_v2.
9. Move all processed CSV files from staging/ to final/.
10. Move files larger than 5MB from incoming/ to large_files/.
11. Move files older than 7 days into archive/.
12. Move files starting with sales_ into a target folder.
13. After processing, move raw files into processed_backup/.
14. Separate Excel files into excel/ and CSV files into csv/.
15. Delete a single file safely.
16. Delete an entire directory containing reports.
17. Delete all folders starting with temp_.
18. Delete files older than 30 days from logs.
19. Remove a folder only if it exists.
20. Clean an output folder before regeneration.
21. Create project folder structure: raw, processed, reports, archive.
22. Clone directory structure without copying files.
23. Rename a directory using shutil.
24. Merge two folders into one destination.
25. Move an entire folder programmatically.
26. Check total, used, and free disk space.
27. Print disk usage in GB.
28. Stop processing if free disk < 2GB.
29. Log disk usage before and after copy.
30. Handle errors when copying existing files.
31. Gracefully handle permission errors.
32. Log failed copy operations.
33. Retry file copy once if it fails.
34. Avoid crash if source folder is missing.
35. Build pipeline: copy raw → staging → final → archive.
36. Create automated daily backup of reports.
37. Clean yesterday's temp files.
38. Move reports into date-wise folders.
39. Archive monthly reports into year folders.
40. Verify copied file size matches source.
41. Confirm folder copy success.
42. Compare file count before and after copy.
43. Detect incomplete copy operations.
44. Copy Excel reports generated via openpyxl.
45. Move CSV files downloaded via requests.
46. Clean old JSON files before new API writes.
47. Archive log files generated using logging.
48. Rotate backups – keep only last 5.
49. Create disaster recovery script.
50. Safely replace production data folder.
51. Create reusable copy-with-timestamp function.

52. Create reusable cleanup function.
53. Efficiently copy folder with thousands of files.
54. Measure time taken for copy.
55. Optimize large dataset movement.
56. Build a complete automated data analytics file system pipeline.