✅ NumPy Mastery — 100% Practical Challenge Set (Data Analyst Version)

40 practical coding tasks — no theory, no MCQs

📌 SECTION A — Array Creation & Manipulation (Basics to Intermediate)

1. Create a NumPy array from 1 to 100 (inclusive).

2. Create a 5×5 matrix filled with random integers between 10 and 99.

3. Generate an array of 50 evenly spaced numbers between 0 and 5.

4. Convert the list below into a NumPy float array:

```
[10, 20, 30, 40, 50]
```

5. Create a 10-element array of zeroes and replace the 5th index with the value 99.

6. Create an array of shape (3,4) with random floats and round the values to 2 decimals.

7. Create a 4×4 identity matrix.

8. Build a 3×3 matrix where every value is the row index + column index.

9. Create a 4×3 array containing numbers from 1 to 12 reshaped using `.reshape()`.

10. Flatten a 2D array into a 1D array using `ravel()` and print shape before/after.

📌 SECTION B — Indexing, Slicing, Filtering

11. Extract the 2nd, 3rd, and 4th elements from this array:

```
a = np.array([5, 10, 15, 20, 25, 30])
```

12. From the matrix below, extract the last two columns:

```
m = np.array([[1,2,3],
              [4,5,6],
              [7,8,9]])
```

13. Filter only values > 50 from a random 1D array of 20 integers (0–100).

14. From the array below, extract only even numbers:

```
a = np.array([2,5,7,9,12,14])
```

15. Reverse the order of columns in a 2D array using slicing.

16. Replace all negative values in the array with 0.

(Create an array that contains both positive & negative random numbers.)

17. Check if the array contains any value equal to 25 (True/False).

18. Extract every alternate row from a 6×3 matrix.

📌 SECTION C — Numeric Computation & Statistics

19. Compute the mean, median, std deviation, min, max for:

```
arr = np.array([10, 30, 40, 20, 50])
```

20. Normalize an array between 0 and 1 (min-max scaling).

21. Standardize an array (z-score normalization).

22. Calculate the percent change between consecutive elements in:

```
sales = np.array([100, 120, 150, 130, 180])
```

23. Apply the NumPy exponential, log, and sqrt functions on a 1D array.

24. Using vectorization, calculate squareroot of numbers from 1–15.

25. Calculate dot product of two 1D arrays of equal length.

26. Calculate matrix multiplication (`@`) for two 3×3 matrices.

27. Compute the correlation matrix for:

```
data = np.array([[10,20,30],
                 [40,50,60],
                 [10,20,15]])
```

28. Get unique values and their counts from an array containing duplicates.

📌 SECTION D — Reshaping, Joining, Splitting

29. Reshape a 1D array of 24 values into (4,6), then reshape into (6,4).

30. Stack two arrays vertically:

```
x = [1,2,3]
y = [4,5,6]
```

31. Stack the same arrays horizontally.

32. Split an array of 12 numbers into 3 equal parts.

33. Concatenate three separate 1D arrays into one final array.

34. Repeat each element of an array 3 times using NumPy.

📌 SECTION E — Broadcasting, Vectorization & Random

35. Add a 1D array to a 2D array using broadcasting (3×3 + 1×3).

36. Multiply a column vector (3×1) with a row vector (1×4) → output must be 3×4.

37. Use broadcasting to subtract the column mean from each column of a 5×5 matrix.

38. Generate a 4×4 matrix of random floats (0–1) with `seed=42`.

39. Shuffle an array randomly using NumPy only (no Python shuffle).

40. Real Analyst Scenario:

You have **24 months** of monthly sales:

```
sales = np.arange(1,25)
```

Reshape it into **12 rows × 2 columns** (12 months × 2 years) for comparison.
