NumPy 50 Tasks – Beginner, Intermediate, and Advanced

# Beginner Level

1. Create a NumPy array of 10 zeros.

2. Create a NumPy array of 10 ones.

3. Create a NumPy array of 10 fives.

4. Create an array of integers from 10 to 50.

5. Create an array of all even integers from 10 to 50.

6. Create a 3x3 matrix with values ranging from 0 to 8.

7. Create a 3x3 identity matrix.

8. Use NumPy to generate a random number between 0 and 1.

9. Use NumPy to generate an array of 25 random numbers from a standard normal distribution.

10. Create a 10x10 array with random values and find the minimum and maximum values.

# Intermediate Level

1. Create a 5x5 matrix with values 1,2,3,...,25 and reshape it.

2. Create an array of 10 linearly spaced points between 0 and 1.

3. Create a 5x5 matrix with row values ranging from 0 to 4.

4. Use broadcasting to add a vector to each row of a matrix.

5. Reverse an array (first element becomes last).

6. Create a checkerboard 8x8 matrix using slicing.

7. Create a matrix with random values and replace values higher than 0.5 with 1 and the rest with 0.

8. Find the mean, standard deviation, and variance of a given NumPy array.

9. Sort a NumPy array along the second axis.

10. Multiply a 5x3 matrix by a 3x2 matrix (matrix product).

# Advanced Level

1. Create a custom dtype for a structured NumPy array with 'name' and 'age'.

2. Extract all numbers from a NumPy array that are divisible by both 3 and 5.

3. Compute the correlation coefficient matrix for a 2D array.

4. Use a boolean mask to modify a NumPy array.

5. Implement a sliding window view of a 1D array.

6. Normalize a NumPy matrix (zero mean and unit variance).

7. Convert a NumPy array to a PIL image and back.

8. Find the row with the highest sum in a 2D array.

9. Replace NaNs with column averages in a NumPy array.

10. Use np.where to implement a conditional logic on arrays.

11. Implement one-hot encoding manually using NumPy.

12. Calculate the moving average of an array.

13. Detect and count outliers in an array (values beyond 3 std deviations).

14. Write a function to flatten a multi-dimensional array without using .flatten() or .ravel().

15. Simulate 1000 coin flips and count how many times 3 heads appear consecutively.

16. Generate a 3D array of shape (3, 3, 3) with random values and sort along the last axis.

17. Extract unique rows from a 2D NumPy array.

18. Compute the histogram of an image using NumPy.

19. Build a distance matrix for a list of 2D coordinates.

20. Find the most frequent value in a NumPy array.