NumPy Assignment Questions

Data Types & Attributes

1. Create a NumPy array `arr` of integers from 0 to 5 and print its data type.

2. Given a NumPy array `arr`, check if its data type is `float64`.

3. Create a NumPy array `arr` with a data type of `complex128` containing three complex numbers.

4. Convert an existing NumPy array `arr` of integers to `float32` data type.

5. Given a NumPy array `arr` with `float64` data type, convert it to `float32` to reduce decimal precision.

6. Write a function `array_attributes` that takes a NumPy array as input and returns its shape, size, and data type.

7. Create a function `array_dimension` that returns the dimensionality of a NumPy array.

8. Design a function `item_size_info` that returns the item size and total size in bytes of a NumPy array.

9. Create a function `array_strides` that returns the strides of the given NumPy array.

10. Design a function `shape_stride_relationship` that returns the shape and strides of the given array.

Array Creation

11. Create a function `create_zeros_array(n)` to return a NumPy array of zeros with `n` elements.

12. Write a function `create_ones_matrix(rows, cols)` to create a 2D array filled with ones.

13. Write a function `generate_range_array(start, stop, step)` to create a ranged NumPy array.

14. Design a function `generate_linear_space(start, stop, num)` for equally spaced values.
15. Create a function `create_identity_matrix(n)` using `numpy.eye`.
16. Write a function that converts a Python list into a NumPy array.
17. Create a NumPy array and use `numpy.view()` to create a new view with the same data.

Concatenation and Stacking

18. Write a function to concatenate two NumPy arrays along a specified axis.
19. Concatenate two arrays with different shapes horizontally using `numpy.concatenate`.
20. Vertically stack multiple arrays from a list using `numpy.vstack`.

Array Generation

21. Write a function to generate an array of integers within a specified range (inclusive) with a step.
22. Generate 10 equally spaced values between 0 and 1 using NumPy.
23. Create 5 logarithmically spaced values between 1 and 1000.

Pandas + NumPy

24. Create a Pandas DataFrame from a NumPy array with 5 rows and 3 columns of random integers (1-100).
25. Write a function to replace all negative values in a specific column with zeros using NumPy.

Indexing and Slicing

26. Access the 3rd element from the given NumPy array.
27. Retrieve the element at index (1, 2) from a 2D array.
28. Extract elements greater than 5 using boolean indexing.
29. Slice elements from index 2 to 5 (inclusive) from a NumPy array.
30. Slice the sub-array `[[2, 3], [5, 6]]` from a 2D array.

Advanced Indexing

31. Extract elements based on indices from a 2D array.
32. Filter elements greater than a threshold using boolean indexing.
33. Extract specific elements from a 3D array using separate index arrays.
34. Return elements satisfying two boolean conditions.
35. Extract elements from a 2D array using separate row and column index arrays.

Broadcasting

36. Add scalar 5 to every element of an array using broadcasting.
37. Multiply each row of a `(3, 4)` array by corresponding elements of a `(1, 3)` array.
38. Add a `(1, 4)` array to every row of a `(4, 3)` array using broadcasting.
39. Add two arrays of shapes `(3, 1)` and `(1, 3)` using broadcasting.
40. Handle shape incompatibility during multiplication between `(2, 3)` and `(2, 2)` arrays.

Aggregations and Statistics

41. Calculate column-wise mean of a 2D array.
42. Find maximum value in each row.
43. Find indices of maximum values in each column.
44. Apply a custom function to compute moving sum along rows.
45. Check if all elements in each column are even.

Reshaping and Flattening

46. Reshape a given array into dimensions `m x n`.
47. Return a flattened version of a given matrix.
48. Concatenate two arrays along a specified axis.
49. Split an array into sub-arrays along a specified axis.
50. Insert and delete elements at specified indices from an array.

Element-wise Operations

51. Perform element-wise addition between two arrays.
52. Perform element-wise subtraction: subtract `arr2` from `arr1`.
53. Perform element-wise multiplication.
54. Divide elements of `arr1` by `arr2` element-wise.
55. Perform element-wise exponentiation: `arr1 ** arr2`.

String Operations

56. Count occurrences of a substring in a string array.
57. Extract uppercase characters from a string array.

58. Replace substring occurrences with another string in a string array.
59. Concatenate strings in a NumPy array element-wise.
60. Find the length of the longest string in a string array.

Descriptive Statistics

61. Generate 100 random integers (1–1000) and compute mean, median, variance, and std deviation.
62. Generate 50 random numbers (1–100) and compute the 25th and 75th percentiles.
63. Compute correlation coefficient between two arrays using `np.corrcoef`.
64. Perform matrix multiplication using `np.dot`.
65. Compute the 10th, 50th, and 90th percentiles and quartiles for an array of 50 integers.

Search and Sort

66. Find index of a specific element in an array.
67. Sort a random array in ascending order.
68. Filter elements greater than 20.
69. Filter elements divisible by 3.
70. Filter elements between 20 and 40 (inclusive).

Byte Order & Swapping

71. Check byte order of a NumPy array using `dtype.byteorder`.
72. Perform in-place byte swapping using `byteswap()`.