**NumPy Array Creation and Properties**

1. **Array Creation:** Create a NumPy array named array\_a containing the integers 1 through 9.
2. **Zeros and Shape:** Create a 3×4 NumPy array named array\_zeros filled entirely with the value 0.0.
3. **Ones and Data Type:** Create a 2×5 NumPy array named array\_ones filled entirely with the value 1, ensuring its data type is explicitly **integer** (int32).
4. **Array of Even Numbers:** Use the np.arange() function to create an array of all even numbers from 10 up to, but not including, 30.
5. **Linearly Spaced Array:** Use the np.linspace() function to create an array of 10 equally spaced points between 0 and 1 (inclusive).
6. **Identity Matrix:** Create a 4×4 identity matrix named identity\_matrix.
7. **Random Array:** Create a 3×3 array named random\_array filled with random floating-point numbers between 0 and 1.
8. **Reshaping:** Reshape the array created in Question 1 (array\_a) into a 3×3 matrix. Print the new shape.

**NumPy Indexing, Slicing, and Manipulation**

1. **Basic Indexing:** Given the 3×3 array from Question 8, retrieve and print the element in the **second row** and **third column**.
2. **Row Slicing:** From the same 3×3 array, retrieve and print the entire **first row**.
3. **Column Slicing:** From the same 3×3 array, retrieve and print the entire **third column**.
4. **Boolean Indexing:** Given an array data = np.array([10, 25, 3, 40, 15]), use **boolean indexing** to print only the elements that are **greater than 20**.
5. **Array Flattening:** Given a 2×2 array multi\_dim = np.array([[1, 2], [3, 4]]), use the flatten() method to convert it into a 1D array.
6. **Vertical Stacking:** Given two 1D arrays, arr1 = np.array([1, 2, 3]) and arr2 = np.array([4, 5, 6]), use np.vstack() to stack them vertically into a 2×3 array.

**NumPy Operations and Calculations**

1. **Element-wise Addition:** Given x = np.array([1, 2, 3]) and y = np.array([4, 5, 6]), calculate and print the result of **x+y**.
2. **Scalar Multiplication:** Given arr = np.array([2, 4, 6]), multiply every element in the array by 5 and print the result.
3. **Mean Calculation:** Given the array scores = np.array([85, 90, 78, 92, 88]), use a NumPy function to calculate and print the **mean (average)** score.
4. **Sum and Product:** For the same scores array, calculate and print both the **sum** of all elements and the **product** of all elements using two separate NumPy functions.
5. **Dot Product:** Given mat1 = np.array([[1, 2], [3, 4]]) and mat2 = np.array([[5, 6], [7, 8]]), calculate the **matrix dot product** of the two matrices.
6. **Universal Functions (ufunc):** Given angles = np.array([0, np.pi/2, np.pi]), apply the **sine function** (np.sin()) element-wise to the array and print the result.