Array Operations, Functions, and Matrix Manipulations

```
import numpy as np

print("\n--- TASK 1: PASSING LIST ---")

# Task 1.1: Convert list to array and display type
lst = [1, 2, 3, 4, 5, 6, 7, 8]
arr = np.array(lst)
print("Array:", arr)
print("Type:", type(arr))
# Output: [1 2 3 4 5 6 7 8], <class 'numpy.ndarray'>

# Task 1.2: Convert nested list to matrix and show shape & dtype
matrix = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]
matrix_arr = np.array(matrix)
print("Matrix:\n", matrix_arr)
print("Shape:", matrix_arr.shape)
print("Data type:", matrix_arr.dtype)
# Output: (3, 3), int64

print("\n--- TASK 2: OPERATORS ---")

# ARANGE
print("Arange 1 to 10:", np.arange(1, 11))  # 1 to 10
print("Arange with step 2:", np.arange(1, 11, 2))  # 1,3,5,7,9

# ZEROS AND ONES
print("Zeros array:", np.zeros(3))  # [0. 0. 0.]
print("10x10 Ones matrix:\n", np.ones((10, 10)))
```

```
# LINSPACE
print("Linspace 25 numbers from 1 to 11:", np.linspace(1, 11, 25))

# IDENTITY MATRIX
print("10x10 Identity Matrix:\n", np.eye(10))

# RANDOM PACKAGE
print("Random uniform (0,1) array:", np.random.rand(4))  # 1D array of 4 values

# RANDN
print("Random standard normal (1D):", np.random.randn(2))
print("Random standard normal (6x6):\n", np.random.randn(6, 6))

# RANDINT
print("Random int (1 to 5):", np.random.randint(1, 6))
print("Random 1D array of 10 ints (1 to 5):", np.random.randint(1, 6, 10))

# ARRAY AND ATTRIBUTES
print("Sequential 1D array (0-24):", np.arange(25))
print("Random 10 integers (0-49):", np.random.randint(0, 50, 10))

a = np.array([[1, 2, 3], [4, 5, 6]])
print("Shape of a:", a.shape)  # (2, 3)

# RESHAPE
a = np.array([1, 2, 3, 4, 5, 6])
reshaped = a.reshape(2, 3)
print("Reshaped 2x3:\n", reshaped)

# MINIMUM
arr_min = np.array([8, 3, 5, 1, 9])
```

```
print("Minimum value:", arr_min.min())  # Output: 1

print("\n--- TASK 3: ARGUMENT FUNCTIONS, SLICING, INDEXING ---")

# ARGMAX
r = np.array([10, 22, 5, 78, 3])
print("Index of max value:", np.argmax(r))  # Output: 3

# SLICING
a = np.array([1, 2, 3, 4, 5, 6, 7, 8, 9, 10])
a[:5] = 1000
print("After slicing:", a)  # [1000 1000 1000 1000 1000 6 7 8 9 10]

# INDEXING
mat = np.array([[1, 2, 3], [4, 5, 6], [7, 8, 9]])
print("Matrix:\n", mat)

print("\n--- TASK 4: ARITHMETIC OPERATORS ---")

# Arithmetic on array
c = np.arange(1, 11)
print("Original array:", c)
print("Addition:", c + c)
print("Subtraction:", c - c)
print("Multiplication:", c * c)
print("Division:", c / c)

print("\n--- TASK 5: UNIVERSAL ARRAY FUNCTIONS ---")

# ELEMENT-WISE ADDITION
arr1 = np.array([1, 2, 3])
```

```
arr2 = np.array([4, 5, 6])

print("Element-wise addition:", arr1 + arr2)


# TRIGONOMETRIC FUNCTIONS

angles = np.array([0, np.pi / 2, np.pi])

print("Sine:", np.sin(angles))

print("Cosine:", np.cos(angles))


# EXPONENTIATION

base_arr = np.array([2, 3, 4])

exp = 3

print("Exponentiation:", np.power(base_arr, exp))  # [8 27 64]


# SQUARE ROOT

sq_arr = np.array([4, 9, 16, 25])

print("Square roots:", np.sqrt(sq_arr))  # [2. 3. 4. 5.]


# MATRIX MULTIPLICATION (element-wise)

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

print("Element-wise matrix multiplication:\n", mat * mat)
```