ASSIGNMENT 05

SUBMITTED TO: MR. MUBASHIR IQBAL

SUBMITTED BY: AHMED SALEEM RANA

REG ID: 24-CYS-023

SECTION: B

HITEC UNIVERSITY TAXILA, CANTT

TASK 01

Write a Python program to find the maximum and minimum value of a given flattened array. Expected Output:

• Original flattened array:

[[0 1], [2 3]]

• Maximum value of the above flattened array: 3

• Minimum value of the above flattened array: 0

INPUT:

```python
import numpy as np

# create a 2D array
arr = np.array([[0, 1], [2, 3]])

print("Original flattened array:")
print(arr)

# flatten the array
flat_arr = arr.flatten()

# find max and min values
max_val = flat_arr.max()
min_val = flat_arr.min()

# display results
print("Maximum value of the above flattened array:", max_val)
print("Minimum value of the above flattened array:", min_val)
```

OUTPUT:

```
Original flattened array:
[[0 1]
 [2 3]]
Maximum value of the above flattened array: 3
Minimum value of the above flattened array: 0

=== Code Execution Successful ===
```

TASK 02

Create a 3x3 matrix with values ranging from 2 to 10.

INPUT:

```python
import numpy as np

mat_3x3 = np.arange(2, 11).reshape(3, 3)
print("3x3 matrix with values from 2 to 10:")
print(mat_3x3)

```

OUTPUT:

```
3x3 matrix with values from 2 to 10:
[[ 2  3  4]
 [ 5  6  7]
 [ 8  9 10]]

=== Code Execution Successful ===
```

TASK 03

Write a NumPy program to convert an array to a float type. Sample output:

• Original array [1, 2, 3, 4]

• Array converted to a float type: [1. 2. 3. 4.]

INPUT:

```python
import numpy as np

orig = np.array([1, 2, 3, 4])
converted = orig.astype(float)

print("Original array", orig)
print("Array converted to a float type:", converted)

```

OUTPUT:

```
Original array [1 2 3 4]
Array converted to a float type: [1. 2. 3. 4.]

=== Code Execution Successful ===
```

TASK 04

Multiply 2 x 2 matrix and take transpose of a resultant matrix.

INPUT:

```python
import numpy as np

M1 = np.array([[5, 5],
               [1, 2]])

M2 = np.array([[0, 1],
               [1, 0]])

result = np.dot(M1, M2)   # or M1 @ M2
transpose_result = result.T
print("Matrix M1:")
print(M1)
print("Matrix M2:")
print(M2)
print("Multiplication of M1 and M2:")
print(result)

print("Transpose of the resultant matrix:")
```

OUTPUT:

```
Matrix M1:
[[5 5]
 [1 2]]
Matrix M2:
[[0 1]
 [1 0]]
Multiplication of M1 and M2:
[[5 5]
 [2 1]]
Transpose of the resultant matrix:
[[5 2]
 [5 1]]

=== Code Execution Successful ===
```

TASK 05

Find inverse of matrix.

INPUT:

```python
import numpy as np
M = np.array([
    [6,  1, 1, 3],
    [4, -2, 5, 1],
    [2,  8, 7, 6],
    [3,  1, 9, 7]
], dtype=float)

print("M =")
print(M)

try:
    M_inv = np.linalg.inv(M)
    print("Inverse of M =")
    print(M_inv)
except np.linalg.LinAlgError:
    print("Matrix is singular (non-invertible).")

```

OUTPUT:

```
M =
[[ 6.  1.  1.  3.]
 [ 4. -2.  5.  1.]
 [ 2.  8.  7.  6.]
 [ 3.  1.  9.  7.]]
Inverse of M =
[[ 0.13368984  0.10695187  0.02139037 -0.09090909]
 [-0.00229183  0.02673797  0.14820474 -0.12987013]
 [-0.12987013  0.18181818  0.06493506 -0.02597403]
 [ 0.11000764 -0.28342246 -0.11382735  0.23376623]]

=== Code Execution Successful ===
```

CONCLUSION:

In this lab, we efficiently used NumPy to perform a variety of numerical and matrix-based operations. We explored how to flatten arrays, find maximum and minimum values, create matrices with specific ranges, convert data types, and perform matrix multiplication and transposition. Additionally, we calculated the inverse of a matrix using NumPy's linear algebra functions. These tasks demonstrate how NumPy simplifies complex mathematical operations and provides fast, reliable tools for data handling and computation. Overall, this lab enhanced our understanding of matrix operations and showed the importance of NumPy in scientific and engineering applications.