Advanced Python – Practical Programs (NumPy)

Course: MCA Semester-III
Subject: Advanced Python (3CS2010308T)
Topic: NumPy

Program 1: Array Creation, Indexing, and Slicing

Objective:
Create a 2D array of shape (4x5) with values from 1 to 20. Perform the following:
- Extract the second row.
- Extract the third column.
- Extract a subarray from rows 2-3 and columns 2-4.

Program Code:

```
import numpy as np
data = np.arange(1, 21).reshape(4, 5)
print("Original Array:\n", data)
print("Second row:", data[1])
print("Third column:", data[:, 2])
print("Subarray [1:3, 1:4]:\n", data[1:3, 1:4])
```

Expected Output:

```
Original Array:
[[ 1  2  3  4  5]
 [ 6  7  8  9 10]
 [11 12 13 14 15]
 [16 17 18 19 20]]
Second row: [ 6  7  8  9 10]
Third column: [ 3  8 13 18]
Subarray [1:3, 1:4]:
[[ 7  8  9]
 [12 13 14]]
```

Program 2: Copy vs View with Data Type Conversion

Objective:
Create a 1D NumPy array. Generate a copy and a view. Modify elements and observe effects. Convert the array to float.

Program Code:

```
import numpy as np
data = np.array([10, 20, 30, 40, 50])
view = data.view()
copy = data.copy()
view[0] = 100
copy[1] = 200
print("Original:", data)
print("View:", view)
print("Copy:", copy)
float_array = data.astype(float)
print("Float array:", float_array)
```

Expected Output:

```
Original: [100  20  30  40  50]
View: [100  20  30  40  50]
Copy: [ 10 200  30  40  50]
Float array: [100.  20.  30.  40.  50.]
```

Program 3: Array Reshaping and Iteration

Objective:
Create a 1D array of values from 1 to 24. Reshape to (4x6), (2x12). Iterate using loops.

Program Code:

```
import numpy as np
data = np.arange(1, 25)
reshaped_1 = data.reshape(4, 6)
reshaped_2 = data.reshape(2, 12)
print("Reshaped (4x6):\n", reshaped_1)
print("Reshaped (2x12):\n", reshaped_2)
print("Iterating over array:")
for x in np.nditer(reshaped_1):
   print(x, end=' ')
```

Expected Output:

Reshaped (4x6):
[[1 2 3 4 5 6]
 [7 8 9 10 11 12]
 [13 14 15 16 17 18]
 [19 20 21 22 23 24]]
Reshaped (2x12):
[[1 2 3 4 5 6 7 8 9 10 11 12]
 [13 14 15 16 17 18 19 20 21 22 23 24]]
Iterating over array:
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24

Program 4: Join, Split, and Sort Arrays

Objective:
Create two 1D arrays. Join them, split them into two. Sort in descending order.

Program Code:

```
import numpy as np
a = np.array([3, 6, 9, 12])
b = np.array([15, 18, 21, 24])
joined = np.concatenate((a, b))
split = np.split(joined, 2)
sorted_desc = np.sort(joined)[::-1]
print("Joined array:", joined)
print("Split arrays:", split)
print("Sorted (desc):", sorted_desc)
```

Expected Output:

Joined array: [3 6 9 12 15 18 21 24]
Split arrays: [array([3, 6, 9, 12]), array([15, 18, 21, 24])]
Sorted (desc): [24 21 18 15 12 9 6 3]

Program 5: Array Filtering and Searching

Objective:
Create a NumPy array. Filter values > 10, find their indices, replace values > 20 with 0.

Program Code:

```
import numpy as np
data = np.array([5, 12, 7, 20, 35, 2, 18, 27])
filtered = data[data > 10]
indices = np.where(data > 10)
data[data > 20] = 0
print("Filtered values >10:", filtered)
print("Indices of values >10:", indices)
print("Array after replacement:", data)
```

Expected Output:

```
Filtered values >10: [12 20 35 18 27]
Indices of values >10: (array([1, 3, 4, 6, 7]),)
Array after replacement: [ 5 12  7 20  0  2 18  0]
```