Department of Engineering Sciences and Technology,

Second Year Btech in Computer Science

Project Based Learning-Python

Assignment - 11

Name - Paritosh kolwadkar

SRN – 31231313

Roll no – 39

Batch – D2

Problem statement : **Write a program to demonstrate various indexing techniques (e.g., accessing specific elements, rows, columns) and slicing operations to extract subarrays. Include examples of Boolean and fancy indexing.**

Pre-requisites:Familiarity with NumPy arrays and their indexing capabilities.

Understanding of slicing syntax and Boolean/fancy indexing in NumPy.

Code:

```
# Import NumPy
import numpy as np


# Create a 2D NumPy array
array = np.array([
   [10, 20, 30, 40],
   [50, 60, 70, 80],
   [90, 100, 110, 120]
```

```
])


print("Original Array:")

print(array)


# 1. Accessing specific elements

print("\nAccess specific element [1,2] (row 1, column 2):", array[1, 2])  # 70


# 2. Accessing entire rows and columns

print("\nAccess row 0:", array[0])  # [10, 20, 30, 40]

print("Access column 2:", array[:, 2])  # [30, 70, 110]


# 3. Slicing operations

print("\nSlicing rows 0 to 1 and columns 1 to 3:")

print(array[0:2, 1:3])  # Subarray [[20, 30], [60, 70]]


# 4. Boolean indexing

print("\nBoolean indexing (elements > 50):")

print(array[array > 50])  # [60, 70, 80, 90, 100, 110, 120]


# 5. Fancy indexing

print("\nFancy indexing (specific rows and columns):")

rows = [0, 2]  # First and last rows

columns = [1, 3]  # Second and last columns

print(array[np.ix_(rows, columns)])  # Subarray [[20, 40], [100, 120]]
```

Explanation :

Accessing Specific Elements:

- `array[1, 2]`: **Accesses the element at the second row and third column (70).**

Accessing Entire Rows/Columns:

- `array[0]`: **Extracts the first row.**
- `array[:, 2]`: **Extracts the third column.**

Slicing:

- `array[0:2, 1:3]`: **Extracts a subarray covering rows 0 to 1 (exclusive of row 2) and columns 1 to 2 (exclusive of column 3).**

Boolean Indexing:

- `array[array > 50]`: **Creates a mask for elements greater than 50 and extracts them.**

Fancy Indexing:

- `np.ix_`: **Used to specify multiple specific rows and columns to extract a subarray.**

Output:

```
Original Array:
[[ 10  20  30  40]
 [ 50  60  70  80]
 [ 90 100 110 120]]

Access specific element [1,2] (row 1, column 2): 70

Access row 0: [10 20 30 40]
Access column 2: [ 30  70 110]

Slicing rows 0 to 1 and columns 1 to 3:
[[20 30]
 [60 70]]
```

Boolean indexing (elements > 50):

[60 70 80 90 100 110 120]

Fancy indexing (specific rows and columns):

[[20 40]

 [100 120]]

Output Explained:

Specific Element:

- The program extracts a single value using row-column indexing.

Entire Rows and Columns:

- Extracts all values from a particular row or column.

Slicing:

- Extracts a subset of the array using a combination of row and column slices.

Boolean Indexing:

- Filters elements that satisfy a condition (e.g., `>50`).

Fancy Indexing:

- Extracts specific rows and columns based on given indices.