Experiment – 14

Aim: Implementing Indexing, Slicing, Reshaping of NumPy arrays.

Objective: Indexing in 1D, 2D and 3D NumPy arrays
Program:

```
import numpy as np

#Create a 1-D array
b = np.array([1, 2, 3, 4, 5])
print(b[1])

#Create a 2-D array
c = np.array([[1, 2, 3], [4, 5, 6]])
# Select the element at row 1 column 2
print(c[1,2])
#or
print(c[1][0])


#Create a 3-D array
d = np.array([[[1, 2, 3], [4, 5, 6]], [[1, 2, 3], [4, 5, 6]]])
# Select the element at row 1, column 2 and
print(d[1,1,2])
```

Output:

```
2

6

4

6
```

Objective: Slicing in 1D, 2D and 3D NumPy arrays
Program:

```
#Converting 1-D array with 12 elements into a 2-D array.
import numpy as np
S = np.array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])

print(S[2:5])
print(S[:4])
print(S[6:])
print(S[:])

A = np.array([
[11, 12, 13, 14, 15],
[21, 22, 23, 24, 25],
[31, 32, 33, 34, 35],
[41, 42, 43, 44, 45],
[51, 52, 53, 54, 55]])

print(A[:3, 2:])
print(A[3:, :])
print(A[:, 4:])
print(A[::2, ::3])
print(A[::, ::3])
```

Output:

```
[2 3 4]

[0 1 2 3]
```

```
[6 7 8 9]

[0 1 2 3 4 5 6 7 8 9]
[[13 14 15]
 [23 24 25]
 [33 34 35]]

[[41 42 43 44 45]
 [51 52 53 54 55]]

[[15]
 [25]
 [35]
 [45]
 [55]]

[[11 14]
 [31 34]
 [51 54]]

[[11 14]
 [21 24]
 [31 34]
 [41 44]
 [51 54]]
```

Objective: Reshaping in 1D, 2D and 3D NumPy arrays
Program:

```
#Converting 1-D array into a 2-D array.
import numpy as np
arr = np.array([1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12])
newarr = arr.reshape(4, 3)
print(newarr)

#Converting  1-D array into a 3-D array.
arr = np.array([1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12])
newarr = arr.reshape(2, 3, 2)
print(newarr)

#Converting to 2D
arr = np.array([1, 2, 3, 4, 5, 6, 7, 8, 9])
newarr = arr.reshape(3, 3)
print(newarr)

#Converting to layers, rows and columns
arr = np.array([1, 2, 3, 4, 5, 6, 7, 8])
newarr = arr.reshape(2, 2, 2)
print(newarr)

#Flatening the array, Converting multidimensional array to 1D array
arr = np.array([[1, 2, 3], [4, 5, 6]])
newarr = arr.reshape(-1)
print(newarr)
```

Output:

```
[[ 1  2  3]
 [ 4  5  6]
 [ 7  8  9]
```

```
 [10 11 12]]
```

```
[[[ 1  2]
  [ 3  4]
  [ 5  6]]

 [[ 7  8]
  [ 9 10]
  [11 12]]]
```

```
[[1 2 3]
 [4 5 6]
 [7 8 9]]
```

```
[[[1 2]
  [3 4]]

 [[5 6]
  [7 8]]]
```

```
[1 2 3 4 5 6]
```