ASSIGNMENT-2

1. **SOURCE CODE:**

```
import numpy as np
a = np.array([[1,2,3],[4,6,5],[7,8,9]])
Print("Array:", a)
Print("Shape =", a.shape)
```

OUTPUT →

```
Array:- [[1,2,3]
         [4,6,5]
         [7,8,9]]
Shape = (3,3)
```

2. **SOURCE CODE →**

a)
```
my_arr = np.ones((3,4))
Print("Array filled with ones:", my_arr)
```

b)
```
my_arr = 5
Print("Array after adding 5:", my_arr)
```

c)
```
my_arr_log = np.log(my_arr)
Print("Element wise logarithm of the array:.", my_arr_log)
```

d)
```
arr_mean = np.mean(my_arr)
Print("Mean of the array:", arr_mean)
```

e)
```
arr_std = np.std(my_array)
Print("Standard deviation of the array:", arr_std)
```

f)
```
arr_sum = np.sum(my_arr)
Print("Sum of the array:", arr_sum)
```

g)
```
arr_max = np.max(my_arr)
Print("Maximum value of the array:", arr_max)
```

h)
```
arr_max_idx = np.argmax(my_arr)
Print("Index of the max value in the array:", arr_max_idx)
```

i)
```
arr_mean_rows = np.mean(my_arr, axis = 1)
```

Name: Debshuti Banerjee Roll No.: 62

Section: A Year: 3rd (2024)

Print ("Mean along the rows axis : ", arr_mean_rows)

■ OUTPUT → a) Array filled with ones : [[1,1,1]
[1,1,1]
[1,1,1]]

b) Array after adding 5 : [[6,6,6]
[6,6,6]
[6,6,6]]

c) Element-wise logarithm of the array : [[1.791, 1.791 1.791]
[1.791 1.791 1.791]
[1.791 1.791 1.791]]

d) Mean of the array : 6.0

e) Standard Deviation of the array : 0.0

f) Sum of the array : 72.0

g) Max value in the array : 60

h) Index of the max value in the array : 0

i) Mean along the rows axis : [6, 6, 6]

3. ■ SOURCE CODE → my_arr = np.ones ((5,5))

print ("5×5 array filled with ones : ", my_arr)

a) value = my_arr [3,1]

Print ("Value at fourth row and second column : ", value)

b) Slice_arr = my_arr [0:3, 1:3]

Print ("Slice array : (rows 0-2, columns 1-2):", Slice_arr)

c) Slice_arr = my_arr [-3:, :]

Print ("Slice with all columns of the last 3 rows : ", Slice_arr)

d) Slice_arr = my_arr [-3:, :]

Slice_arr [:] = -1

Print ("Slice array after assigning -1 to the last 3 rows : "
Slice_arr)

Name : Debabuti Banerjee Roll No. : 62

Section : A Year : 3rd (2024)

■ OUTPUT → 5×6 array filled with ones : [[1, 1, 1, 1, 1]
[1, 1, 1, 1, 1]
[1, 1, 1, 1, 1]
[1, 1, 1, 1, 1]
[1, 1, 1, 1, 1]]

a) value at fourth row and second column : 1.0

b) Slice array (rows 0-2, column 1-2) : [[1, 1]
[1, 1]
[1, 1]]

c) Slice array with all columns of the last 3 rows :
[[1, 1, 1, 1, 1]
[1, 1, 1, 1, 1]
[1, 1, 1, 1, 1]]

d) Slice array after assigning -1 to the last 3 rows :
[[-1, -1, -1, -1, -1]
[-1, -1, -1, -1, -1]
[-1, -1, -1, -1, -1]]

4. ■ SOURCE CODE →
```
arr = np.ones([4, 6])
Print("4×6 Array filled with ones : ", arr)
a) non_zero_count = np.count_nonzero(zero)
Print("Number of non-zero values : ", non_zero_count)
b) element_count = arr.shape[0]
Print("No of element with row axis : ", element_count)
c) one_d_arr = np.array([0,0,1,2,3,0,0]) trimmed_arr = np.trim
Print("Trimmed one D array : " trimmed_arr) zeros(one_d_arr)
d) rev_arr = arr[::-1, ::-1]
Print("Reversed array : ", rev_arr)
e) dia_sum = np.trace(arr)
```

Name : Debabuti Banerjee Roll No. : 62

Section : A Year : 3rd (2024)

```
Print("Sum of Diagonal elements: ", dia_sum)
```

f)
```
m1 = np.array([[1,2],[3,4]])
m2 = np.array([[5,6],[7,8]])
add_m = m1+m2
sub_m = m1-m2
Print("Added Matrix: ", add_m)
Print("Subtracted Matrix: ", sub_m)
```

g)
```
new_row = np.array([1,1,1,1,1,1])
arr = np.vstack([arr, new_row])
Print("Array after adding a row: ", arr)
new_col = np.array([1],[1],[1],[1],[1],[1])
arr = np.hstack([arr, new_col])
Print("Array adding a column: ", arr)
```

h)
```
m1 = np.array([[1,2]],[3,4]])
m2 = np.array([[5,6]],[7,8]])
mul_m = np.dot(m1, m2)
Print("Multiplication Result: ", mul_m)
```

OUTPUT → a) No of non-zero value : 24

b) No of element along with row axis : 4

c) Trimmed one-D array : [1. 2 3]

d) Reversed array :
```
[[1. 1  1. 1 1]
 [1  1  1  1 1]
 [1  1  1  1 1]
 [1  1  1. 1 1]]
```

e) Sum of diagonal elements : 4.0

f) Added matrix :
```
[[ 6. 8]
 [10. 12]]
```
Subtracted matrix :
```
[[-4. -4]
 [-4. -4]]
```

g) Array after adding a row :
```
[[1 1 1 1 1 1]
 [1. 1 1 1 1 1]
 [1 1 1 1 1 1]]
```

h) Multiplication Result :
```
[[19 22]
 [43 50]]
```

Name: Debahuti Banerjee Roll No.: 62

Section: A Year: 3rd (2024)

5. Source Code → a) sorted_ind = np.argsort(arr, axis = none)
Print("Indices of the sorted array: ", sorted_ind)

b) k = 5
k_smallest_value = np.partition(arr.flatten() k)[:k]
Print("The k smallest values in array: ", k_smallest_value)

c) n = 5
n_lar_val = np.partition(arr.flatten(), -n)[-n:]
Print("The n largest values in array: ", n_lar_val)

d) sorted_m = np.sort(arr, axis = none)
Print("Sorted matrix values: ", sorted_m)

Output → a) Indices of the sorted array: [0 21 20 19 18
17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 22 23]

b) The k smallest values in the array: [1 1 1 1 1]

c) The n largest values in array: [1 1 1 1 1]

d) Sorted matrix values: [1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]