IARE

INSTITUTE OF AERONAUTICAL ENGINEERING

(An Autonomous Institute affiliated to JNTUH, Hyderabad)

Dundigal, Hyderabad - 500 043

LABORATORY WORK BOOK

Name of the Student : HIMAKAR C

Roll Number: 2 1 9 5 1 A 0 5 6 5

Class : CSE - B Semester : VI

Course Code : ACIC08 Course Name : ~~Data~~ DMKD Laboratory

Name of the Course Faculty : Dr. ~~Achutha Suresh Babu~~ D Durga Bhavani Faculty ID : IARE10991

Exercise Number : Week Number : 01 Date : 19/03/24

S. No.	Exercise Number	EXERCISE NAME	MARKS AWARDED						
			Aim/ Preparation	Algorithm / Procedure	Performance in the Lab	Source Code Calculations and Graphs	Program Execution Results and Error Analysis	Viva - Voce	Total
			4	4		4	4	4	20
1	1-1	Implement multi-dine arrays using numpy	4	2	2	4	4	4	20
2	1-2	Implement matrix full of zeros and ones							
3	1-3	Implement functions Reshape and flatten							
4	1-4	Implement functio append data							
5		vertically and horizontally							
6	1-5	Apply Indexing and slicing on							
7		Arrays using Numpy							
8	1-6	Implement statistical functions							
9		on array min, max, mean, median							
10		and SD							
11									
12									

Signature of the Student

Signature of the Faculty

3.1 Implement Multidimensional 2-D & 3-D arrays using Numpy

```
import numpy as np
l = np.array(eval(input('Enter the list: ')))
Print('List is: ')
Print(l)
Print('The shape is: ')
Print(np.shape(l))
Print('The dimension is: ')
Print(np.ndim(l)).
```

OUTPUT/INPUT:

```
Enter the list:
[[1,2,3], [3, 4, 5]]
List is:
[[1  2  3]
 [3  4  5]]
The shape is:
(2, 3)
The dimension is:
2
```

1.2 Implement a matrix full of zeroes & ones

```
import numpy as np
X = np.zeros(5)
Y = np.ones(6)
X = X.astype('int')
Y = Y.astype('int')
Print('Matrix full of zeros is:')
Print(X)
Print('Matrix full of ones is:')
Print(Y)
```

INPUT/OUTPUT:

```
Matrix full of zeros is:
[0 0 0 0 0]
Matrix full of ones is:
[1 1 1 1 1 1]
```

ROLL NUMBER :

1-3 Implement functions Reshape and flatten data in array

```
import numpy as np
l = np.array(list(map(int, input('Enter list: ').split())))
a, b = map(int, input('Enter dimensions to
                       reshape: '))
print('After Reshape: ')
print(np.reshape(l, (a, b))
Print('After flattening: ')
Print(l.flatten())
```

INPUT/OUTPUT:

```
Enter list: 1 2 3 4 5 6 7 8 9
Enter dimensions to reshape: 3 3
After Reshape:
[[1 2 3]
 [4 5 6]
 [7 8 9]]
After flattening:
[1 2 3 4 5 6 7 8 9]
```

ROLL NUMBER :

3.4 Implement functions Append data vertically & horizontally

```
import numpy as np
l1 = np.array(list(map(int, input('Enter list:').split())))
l2 = np.array(list(map(int, input('Enter list2:').split())))
l3 = np.hstack((l1, l2))
Print ('Appending data horizontally:')
print (l3)
l1 = np.array(list(map(int, input('Enter list:').split())))
l2 = np.array(list(map(int, input('Enter list:').split())))
l1 = np.vstack((l1, l2))
Print ('Appending data vertically:')
Print(l1)
```

INPUT/OUTPUT:

```
Enter list: 1 2 3 4 5 6
Enter list2: 7 8 9 0 0 0
Appending data horizontally:
[1 2 3 4 5 6 7 8 9 0 0 0]
Enter list: 1 2 3 4 5
Enter list: 2 2 121 11 1
Appending data vertically:
[[ 1  2  3  4  5]
 [ 2  2 121 11  1]]
```

ROLL NUMBER

1.5 Applying indexing & slicing on array using numpy

```
import numpy as np
l1 = np.arange(10, 1, -2)
print('List is: ', l1)
l = l1[np.array([3, 1, 2])]
print('Elements at indices 3, 1, 2 are: ')
print(l)
l2 = np.arange(10)
print('List is: ', l2)
print('l[5] = ', l2[5])
print('Slicing:')
print('l[-8:-1:1] = ', l2[-8:-1:1])
```

INPUT/OUTPUT:

```
List is: [10 8 6 4 2]
Elements at indices 3, 1, 2 are
[4 8 6]
List is: [0 1 2 3 4 5 6 7 8 9]
l[5] = 5
Slicing:
l[-8:-1:1] = [2 3 4 5 6 7 8]
```

ROLL NUMBER :

1.6 Implement statistical functions on array, Min, Max, Mean, Median and standard Deviation

```
import numpy as np
l = np.array(list(map(int, input('Enter list:').split())))
print("Median is : ", np.median(l))
print('Mean is: ', np.mean(l))
print('Standard Deviation is: ', np.std(l))
print('Maximum is: ', np.max(l))
print('Minimum is: ', np.min(a))
```

INPUT/OUTPUT:

```
Enter list: 1 2 3 4 5
Median is : 3.0
Mean is: 3.0
Standard Deviation is: 1.4142135623730951
Maximum is: 5
Minimum is: 1
```