NAME - AKSHAT SHRIVASTAVA

ROLL - 2305591

SECTION - CSE-35

AD - LAB - 04

NUMPY

1. Write a program to create a null vector of size 25 but the values from fifth to tenth elements are all

```
import numpy as np
v = np.zeros(25)
v[4:10] = 1
print(v)
```

```
[0. 0. 0. 0. 1. 1. 1. 1. 1. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
0.
 0.]
```

2. Write a program to create a vector with values ranging from 5 to 20 & reverse it

```
import numpy as np
v = np.arange(5, 21)
v_rev = v[::-1]
print(v_rev)
```

```
[20 19 18 17 16 15 14 13 12 11 10  9  8  7  6  5]
```

3. Write a program to create a vector with 10 random values & sort first half ascending&secondhalfdescending.

```
import numpy as np
v = np.random.rand(10)

first_half = np.sort(v[:5])
second_half = np.sort(v[5:])[::-1]

result = np.concatenate((first_half, second_half))
print(result)
```

```
[0.03099229 0.08326739 0.40421488 0.63686179 0.69205834 0.9812629
 0.59191616 0.32791435 0.23424283 0.11742402]
```

4. Write a program to create a vector of size 10 with values ranging from 0 to 1, both excluded.

```
import numpy as np
v = np.random.rand(10)
print(v)
```

```
[0.80342207 0.95943523 0.55748974 0.21977651 0.85345431 0.20705055
 0.90133528 0.83665451 0.53114074 0.41754664]
```

5. Write a program to concatenate element-wise two arrays of string.

```
import numpy as np
a = np.array(['akshat', 'shrivastava'])
b = np.array([' 2305591', 'cse-35'])

c = np.char.add(a, b)
print(c)
```

```
['akshat 2305591' 'shrivastavacse-35']
```

6. Write a program to create a 4x3 array with random values & find out the minimum&maximumvalues.

```
import numpy as  np
arr = np.random.rand(4, 3)

print("Min:", arr.min())
print("Max:", arr.max())


Min: 0.05229517944578144
Max: 0.9729940049389464
```

7. Write a program to create a 2D array with 1 on the border and 0 inside.

```
import numpy as  np
arr = np.ones((5, 5))
arr[1:-1, 1:-1] = 0
print(arr)

[[1. 1. 1. 1. 1.]
 [1. 0. 0. 0. 1.]
 [1. 0. 0. 0. 1.]
 [1. 0. 0. 0. 1.]
 [1. 1. 1. 1. 1.]]
```

8. Write a program to create a 4x4 matrix with values 1,2,3,4 just below & above the diagonal, rest zeros

```
import numpy as  np
arr = np.zeros((4, 4))

arr[np.arange(3), np.arange(1,4)] = [1,2,3]
arr[np.arange(1,4), np.arange(3)] = [1,2,3]

print(arr)
```

```
[[0. 1. 0. 0.]
 [1. 0. 2. 0.]
 [0. 2. 0. 3.]
 [0. 0. 3. 0.]]
```

9. Write a program to extract all the contiguous 3x3 blocks from a random 10x10 matrix

```
import numpy as  np
arr = np.random.rand(10, 10)

blocks = []
for i in range(8):
    for j in range(8):
        blocks.append(arr[i:i+3, j:j+3])

blocks = np.array(blocks)
print(blocks.shape)  # (64, 3, 3)
```

```
(64, 3, 3)
```

10. A magic square is a matrix all of whose row sums, column sums and the sums of the twodiagonalsare

the same. (One diagonal of a matrix goes from the top left to the bottom right, the other diagonalgoesfrom top right to bottom left.) Show by direct computation that if the matrix Ais given byA=np.array([[17, 24, 1, 8, 15],

[23, 5, 7, 14, 16], [4, 6, 13, 20, 22], [10, 12, 19, 21, 3], [11, 18, 25, 2, 9]])

The matrix A has 5 row sums (one for each row), 5 column sums (one for each column) andtwodiagonal sums. These 12 sums should all be exactly the same, and you could verify that theyarethesame by printing them and “seeing” that they are the same. It is easy to miss small differencesamongsomany numbers, though. Instead, verify that A is a magic square by constructing the 5 columnsumsandcomputing the maximum and minimum values of the column sums. Do the same for the5rowsums,and compute the two diagonal sums. Check that these six values are the same. If the maximumandminimum values are the same, the flyswatter principle says that all values are the same. Hints: The function np.diag extracts the diagonal of a matrix, and the function np.fliplr extractstheotherdiagonal.

```
import numpy as  np
A = np.array([
    [17,24,1,8,15],
    [23,5,7,14,16],
    [4,6,13,20,22],
    [10,12,19,21,3],
    [11,18,25,2,9]
])

row_sums = A.sum(axis=1)
col_sums = A.sum(axis=0)
```

```
diag1 = np.diag(A).sum()
diag2 = np.diag(np.fliplr(A)).sum()

values = np.concatenate((row_sums, col_sums, [diag1, diag2]))

print("Min:", values.min())
print("Max:", values.max())
```

```
Min: 65
Max: 65
```