NAME: DESAI HINDUJA

REG NO : 21BCB7070

VIT-AP University

Import numpy as np:

```
In [108]: import numpy as np
```

Create an array of 10 zeros:

```
In [109]: t=np.zeros(10)
          t
Out[109]: array([0., 0., 0., 0., 0., 0., 0., 0., 0., 0.])
```

Create an array of 10 ones:

```
In [110]: t=np.ones(10)
          t
Out[110]: array([1., 1., 1., 1., 1., 1., 1., 1., 1., 1.])
```

Create an array of 10 fives:

```
In [111]: t=np.full(10,5)
          t
Out[111]: array([5, 5, 5, 5, 5, 5, 5, 5, 5, 5])
```

Create an array of the integers from 10 to 50:

```
In [116]: arr=np.arange(10,51)
          print(arr)
[10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33
 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50]
```

Create an array of all the even numbers from 10 to 50:

```
In [118]: arr=np.arange(10,51,2)
          print(arr)
[10 12 14 16 18 20 22 24 26 28 30 32 34 36 38 40 42 44 46 48 50]
```

Create a 3x3 matrix with ranging 0 to 8:

```
In [119]: a=np.array([[0,1,2],[3,4,5],[6,7,8]])
          a

Out[119]: array([[0, 1, 2],
                 [3, 4, 5],
                 [6, 7, 8]])
```

Create a 3x3 matrix identity matrix:

```
In [120]: l=np.eye(3)
          l

Out[120]: array([[1., 0., 0.],
                 [0., 1., 0.],
                 [0., 0., 1.]])
```

Use numpy to generate a random between 0 and 1:

```
In [121]: h=np.random.rand()
          print(h)

          0.6555534001009518
```

Use numpy to generate an array of 25 random numbers sampled from a standard normal distribution:

```
In [122]: p=np.random.rand(25)
          print(p)

          [0.97151986 0.74253904 0.8078721  0.42520974 0.25316778 0.2051707
           0.98661339 0.76946844 0.54800328 0.93424898 0.8386575  0.36973817
           0.5592058  0.61364531 0.6644895  0.90321405 0.28410268 0.8766117
           0.41821505 0.14888859 0.34574874 0.42995722 0.87000753 0.34838145
           0.64242323]
```

Create the following matrix:

```
In [123]: u=np.arange(0.01,1.01,0.01).reshape(10,10)
          print(u)

          [[0.01 0.02 0.03 0.04 0.05 0.06 0.07 0.08 0.09 0.1 ]
           [0.11 0.12 0.13 0.14 0.15 0.16 0.17 0.18 0.19 0.2 ]
           [0.21 0.22 0.23 0.24 0.25 0.26 0.27 0.28 0.29 0.3 ]
           [0.31 0.32 0.33 0.34 0.35 0.36 0.37 0.38 0.39 0.4 ]
           [0.41 0.42 0.43 0.44 0.45 0.46 0.47 0.48 0.49 0.5 ]
           [0.51 0.52 0.53 0.54 0.55 0.56 0.57 0.58 0.59 0.6 ]
           [0.61 0.62 0.63 0.64 0.65 0.66 0.67 0.68 0.69 0.7 ]
           [0.71 0.72 0.73 0.74 0.75 0.76 0.77 0.78 0.79 0.8 ]
           [0.81 0.82 0.83 0.84 0.85 0.86 0.87 0.88 0.89 0.9 ]
           [0.91 0.92 0.93 0.94 0.95 0.96 0.97 0.98 0.99 1.  ]]
```

Create an array of 20 linearly spaced points between 0 and 1:

```
In [124]: r=np.linspace(0,1,29)
          print(r)
```

```
[0.         0.03571429 0.07142857 0.10714286 0.14285714 0.17857143
 0.21428571 0.25       0.28571429 0.32142857 0.35714286 0.39285714
 0.42857143 0.46428571 0.5        0.53571429 0.57142857 0.60714286
 0.64285714 0.67857143 0.71428571 0.75       0.78571429 0.82142857
 0.85714286 0.89285714 0.92857143 0.96428571 1.        ]
```

numpy indexing and seletion:

```
In [125]: mat=np.arange(1,26).reshape(5,5)
          mat
```

```
Out[125]: array([[ 1,  2,  3,  4,  5],
                 [ 6,  7,  8,  9, 10],
                 [11, 12, 13, 14, 15],
                 [16, 17, 18, 19, 20],
                 [21, 22, 23, 24, 25]])
```

```
In [126]: mat[2:,1:5]
```

```
Out[126]: array([[12, 13, 14, 15],
                 [17, 18, 19, 20],
                 [22, 23, 24, 25]])
```

```
In [127]: mat[3,4]
```

```
Out[127]: 20
```

```
In [128]: mat[0:3,1:2]
```

```
Out[128]: array([[ 2],
                 [ 7],
                 [12]])
```

```
In [129]: mat[-1:]
```

```
Out[129]: array([[21, 22, 23, 24, 25]])
```

```
In [130]: mat[-2:]
```

```
Out[130]: array([[16, 17, 18, 19, 20],
                 [21, 22, 23, 24, 25]])
```

Get the sum of all the values in mat:

```
In [131]: mat.sum()
Out[131]: 325
```

Get the dtandard deviation of the values in mat:

```
In [132]: s=np.std(mat)
          print(s)
          7.211102550927978
```

Get the sum of all the columns in mat:

```
In [133]: x=np.sum(mat,axis=0)
          print(x.tolist())
          [55, 60, 65, 70, 75]

In [134]: j=np.array([55,60,65,70,75])
          j
Out[134]: array([55, 60, 65, 70, 75])

In [135]: j.sum()
Out[135]: 325
```