

1. Take a 3X3 matrix randomly and another matrix by assigning elements. Now add the 2 matrix and store it in a separate matrix C.

```
In [1]: import numpy as np
a = np.random.randint(1,10,(3,3))
print('a matrix = \n',a)
b_list = [[1,2,3],[4,5,6],[7,8,9]]
b = np.array(b_list)
print('b matrix = \n',b)
c = a + b
print('c matrix = \n',c)
```

```
a matrix =
[[9 3 2]
 [9 5 9]
 [3 2 2]]
b matrix =
[[1 2 3]
 [4 5 6]
 [7 8 9]]
c matrix =
[[10 5 5]
 [13 10 15]
 [10 10 11]]
```

2. Take 3X3 matrixes add it with another 3x3 of all 1 matrix and convert that to a complex datatype matrix.

```
In [2]: import numpy as np
a_list = [[1,2,3],[4,5,6],[7,8,9]]
a = np.array(a_list)
print('a matrix = \n',a)
b = np.ones((3,3),dtype=int)
print('b matrix = \n',b)
c = np.add(a,b)
c = np.array(c,dtype=complex)
print('c matrix = \n',c)
```

```
a matrix =
[[1 2 3]
 [4 5 6]
 [7 8 9]]
b matrix =
[[1 1 1]
 [1 1 1]
 [1 1 1]]
c matrix =
[[ 2.+0.j  3.+0.j  4.+0.j]
 [ 5.+0.j  6.+0.j  7.+0.j]
 [ 8.+0.j  9.+0.j 10.+0.j]]
```

3. Take a 3X3 matrix and multiply element wise with 3X 3 another matrix.

```
In [3]: import numpy as np
a_list = [[1,2,3],[4,5,6],[7,8,9]]
a = np.array(a_list)
print('a matrix = \n',a)
b = np.eye(3,dtype=int)
print('b matrix = \n',b)
c = a*b
print('c matrix = \n',c)
```

```

a matrix =
[[1 2 3]
 [4 5 6]
 [7 8 9]]
b matrix =
[[1 0 0]
 [0 1 0]
 [0 0 1]]
c matrix =
[[1 0 0]
 [0 5 0]
 [0 0 9]]

```

#### 4. Multiply 2 3X3 matrixes.

```

In [4]: import numpy as np
a_list = [[1,2,3],[4,5,6],[7,8,9]]
a = np.array(a_list)
print('a matrix = \n',a)
b = np.eye(3,dtype=int)
print('b matrix = \n',b)
c = a@b
print('c matrix = \n',c)

```

```

a matrix =
[[1 2 3]
 [4 5 6]
 [7 8 9]]
b matrix =
[[1 0 0]
 [0 1 0]
 [0 0 1]]
c matrix =
[[1 2 3]
 [4 5 6]
 [7 8 9]]

```

5. Randomly generate 10 numbers in between 1 to 10 and convert that to a 5X2 matrix. Take a floating-point matrix and perform the addition. a= floating point matrix and b is an integer matrix of samedimension. Why a+=b is not equal to b+=a.Explain briefly. How can we resolve this problem?

```

In [5]: import numpy as np
a = np.random.randint(1,10,(5,2))
print('a matrix = \n',a)
b = np.random.rand(5,2)
print('b matrix = \n',b)
c = a + b
print('c matrix = \n',c)

```

```

a matrix =
[[7 3]
 [8 6]
 [7 2]
 [7 7]
 [8 4]]
b matrix =
[[0.20158343 0.18771561]
 [0.36202891 0.80469033]
 [0.16266698 0.49985948]
 [0.74402508 0.29152472]
 [0.80967768 0.53434798]]
c matrix =
[[7.20158343 3.18771561]
 [8.36202891 6.80469033]
 [7.16266698 2.49985948]
 [7.74402508 7.29152472]
 [8.80967768 4.53434798]]

```

```

In [6]: a.dtype

```

```
Out[6]: dtype('int32')
```

```
In [7]: b.dtype
```

```
Out[7]: dtype('float64')
```

```
In [8]: b += a  
b
```

```
Out[8]: array([[7.20158343, 3.18771561],  
              [8.36202891, 6.80469033],  
              [7.16266698, 2.49985948],  
              [7.74402508, 7.29152472],  
              [8.80967768, 4.53434798]])
```

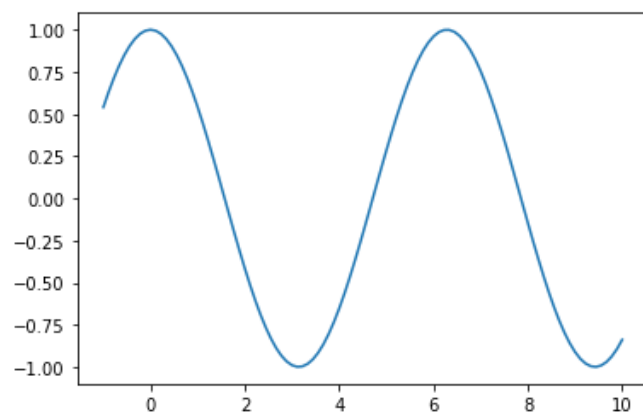
```
In [9]: a += b  
a
```

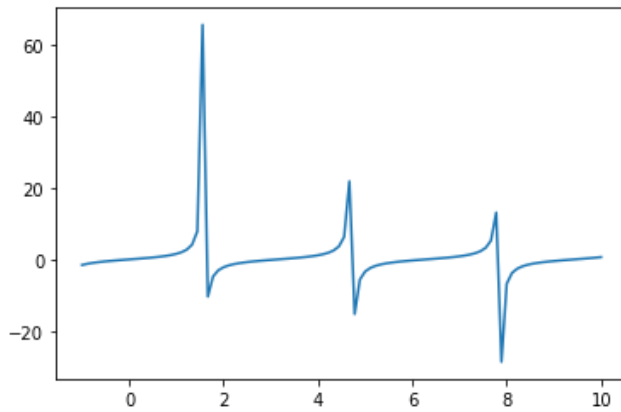
```
-----  
UFuncTypeError                                Traceback (most recent call last)  
Input In [9], in <cell line: 1>()  
----> 1 a += b  
      2 a  
  
UFuncTypeError: Cannot cast ufunc 'add' output from dtype('float64') to dtype('int32') with casting  
rule 'same_kind'
```

$a += b$  adds  $a$  and  $b$  and stores the result in  $a$  and original matrix of  $a$  is overwritten.  $b += a$  adds  $a$  and  $b$  and stores the result in  $b$  and original matrix of  $b$  is overwritten. To resolve it, we can store the result in another matrix  $c$ .

## 6. Take $x$ values and plot the $\cos(x)$ and $\tan(x)$ in a graph using matplotlib.

```
In [10]: import matplotlib.pyplot as plt  
import numpy as np  
x = np.linspace(-1,10,100)  
y = np.cos(x)  
z = np.tan(x)  
plt.plot(x,y)  
plt.show()  
plt.plot(x,z)  
plt.show()
```





7. Take a multidimensional array (3, 3, 3) and print the last column.

```
In [11]: import numpy as np
a = np.random.randint(1,10,(3,3,3))
print('a matrix = \n',a)
print('last column = \n',a[2])
```

```
a matrix =
[[[1 8 4]
  [1 5 9]
  [4 7 2]]

 [[4 4 1]
  [1 5 3]
  [9 4 9]]

 [[9 7 5]
  [8 3 8]
  [4 5 3]]]
last column =
[[9 7 5]
 [8 3 8]
 [4 5 3]]
```

8. Take a function  $f(x) = x^3 + 5y + 4z$  and determine the values and store them in a (3,3,3) matrix.

```
In [12]: import numpy as np
def fun(x,y,z):
    return x**3 + 5*y + 4*z
a = np.fromfunction(fun,(3,3,3),dtype=int)
print('Stored matrix=\n',a)
```

```
Stored matrix=
[[[ 0  4  8]
  [ 5  9 13]
  [10 14 18]]

 [[ 1  5  9]
  [ 6 10 14]
  [11 15 19]]

 [[ 8 12 16]
  [13 17 21]
  [18 22 26]]]
```

9. Using axis add the column values of a 3X3 matrix and then add it with the maximum values of rows taken in a 3X3 matrix

```
In [13]: import numpy as np
a=np.random.randint(1,10,(3,3))
print('a matrix=\n',a)
b=a.sum(axis=0)
print('column_sum matrix=\n',b)
c=a.max(axis=1)
print('row_max matrix=\n',c)
```

```

d=b+c
print('ans matrix=\n',d)

a matrix=
[[3 2 5]
 [3 1 5]
 [9 9 3]]
column_sum matrix=
[15 12 13]
row_max matrix=
[5 5 9]
ans matrix=
[20 17 22]

```

10.  $f(x) = x^3 + 5y$  and store it in (3,3) matrix.

```

In [14]: import numpy as np
def f(x,y):
    return x**3 + 5*y
a=np.fromfunction(f,(3,3),dtype=int)
print('Stored matrix=\n',a)

Stored matrix=
[[ 0  5 10]
 [ 1  6 11]
 [ 8 13 18]]

```

11. Take a function array 'a' cube with a range of 20 and find out what will be the value of a [[7,8], [9,11]]. If we take values a [[7, 8], [9, 21]] will it take if not why?

```

In [15]: import numpy as np
a = np.arange(20)**3
print('cube matrix=\n',a)
b=np.array([[7,8],[9,11]])
print('ans matrix=\n',a[b])

cube matrix=
[ 0  1  8  27  64 125 216 343 512 729 1000 1331 1728 2197
2744 3375 4096 4913 5832 6859]
ans matrix=
[[ 343 512]
 [ 729 1331]]

```

```

In [16]: c=np.array([[7,8],[9,21]])
print('ans matrix=\n',a[c])

```

```

-----
IndexError                                Traceback (most recent call last)
Input In [16], in <cell line: 2>()
      1 c=np.array([[7,8],[9,21]])
----> 2 print('ans matrix=\n',a[c])

IndexError: index 21 is out of bounds for axis 0 with size 20

```

If we take values a [[7,8],[9,21]] it will not work as index 21 is out of bounds for axis 0 of matrix a consists of only 20 elements.

12. Take a random number from 0 to 19 and make a 4X5 matrix then find the values of (i) 3rd row and 4th column only and (ii) only 4th column values.

```

In [17]: import numpy as np
a=np.random.randint(0,19,(4,5))
print('a matrix=\n',a)
b=a[2]
print('3rd row=\t',b)
c=a[2,3]
print('element at 3rd row and 4th column=\t',c)
d=a[:,3]
print('4th column=\t',d)

```

```

a matrix=
[[ 1  0  1  5  3]
 [ 3 16  0 15  7]
 [ 9  9  8 18  0]
 [ 9  7 13 15 13]]
3rd row=      [ 9  9  8 18  0]
element at 3rd row and 4th column=      18
4th column=    [ 5 15 18 15]

```

13. Take a matrix of 5X4 randomly and create 2 3x3 values i and j respectively and take a tuple named m with i and j as an argument and generate the values of the matrix for the specified tuple.  $i \leq 4$  and  $j \leq 3$ .

```

In [18]: import numpy as np
i=np.random.randint(0,3,(3,3))
j=np.random.randint(0,3,(3,3))
c=np.random.randint(10,20,(5,4))
m=(i,j)
print('ans matrix=\n',c[m])

```

```

ans matrix=
[[12 13 14]
 [10 16 16]
 [18 10 12]]

```

14. Take a matrix of 3x3 and find out the Eigen vector and Eigen values of that matrix.

```

In [19]: import numpy as np
a=np.random.randint(0,10,(3,3))
print('a matrix=\n',a)
b=np.linalg.eig(a)
print('eigen values and eigen vectors=\n',b)

```

```

a matrix=
[[0 7 5]
 [5 1 7]
 [4 8 9]]
eigen values and eigen vectors=
(array([16.43293737, -0.6408884 , -5.79204898]), array([[ -0.43709339, -0.71697217,  0.68175761],
 [ -0.48513125, -0.3605248 , -0.70463076],
 [ -0.7573619 ,  0.59663454,  0.19672837]]))

```

15. Take a matrix and by taking the i, j or x, y values implement the hstack and vstack methods.

```

In [20]: import numpy as np
x=np.arange(0,10,2)
y=np.arange(5)
print('x matrix=\n',x,'\ny matrix=\n',y)
m=np.hstack([x,y])
n=np.vstack([x,y])
print('hstack=\n',m,'\nvstack=\n',n)

```

```

x matrix=
[0 2 4 6 8]
y matrix=
[0 1 2 3 4]
hstack=
[0 2 4 6 8 0 1 2 3 4]
vstack=
[[0 2 4 6 8]
 [0 1 2 3 4]]

```

In [ ]: