Q1. Create two 3×3 matrices using the random function in Numpy and perform the following operations.

è Product (prod)

è Multiplication (multiply)

è Dot Product (dot)

In [ ]:

```
import numpy as np
A = np.random.randint(0,9,size=(3,3),dtype=int)
B = np.random.randint(0,9,size=(3,3),dtype=int)
print("A:")
print(A)
print("B:")
print(B)
print("product *")
# print(A*B)
res = np.prod([A, B])
print(res)
print("multiplication")
print(np.multiply(A,B))
print("dot product")
print(np.dot(A,B))
```

```
A:
[[6 3 3]
 [2 0 4]
 [2 1 5]]
B:
[[2 7 7]
 [1 7 5]
 [7 3 8]]
product *
0
multiplication
[[12 21 21]
 [ 2  0 20]
 [14  3 40]]
dot product
[[36 72 81]
 [32 26 46]
 [40 36 59]]
```

Q2. Perform the following set operations using the Numpy functions.

è Union

è Intersection

è Set difference

è XOR

In [ ]:

```
import numpy as np
arr1 = np.array([1, 2, 3, 4])
arr2 = np.array([3, 4, 5, 6])
print(arr1,arr2)
unionArr = np.union1d(arr1, arr2)
print("Union:")
print(unionArr)
intersectionArr = np.intersect1d(arr1, arr2, assume_unique=True)
print("Intersection:")
print(intersectionArr)
differenceArr = np.setdiff1d(arr1, arr2, assume_unique=True)
```

```
print("Set Difference:")
print(differenceArr)
xorArr = np.setxor1d(arr1, arr2, assume_unique=True)
print("XOR:")
print(xorArr)
```

```
[1 2 3 4] [3 4 5 6]
Union:
[1 2 3 4 5 6]
Intersection:
[3 4]
Set Difference:
[1 2]
XOR:
[1 2 5 6]
```

Q3. Create a 1D array using Random function and perform the following operations.

è Cumulative sum

è Cumulative Product

è Discrete difference (with n=3)

è Find the unique elements from the array

In [ ]:

```
import numpy as np
arr1 = np.random.randint(0,9,size=(5),dtype=int)
print(arr1)


print("Cumulative Sum")
cumsumArr = np.cumsum(arr1)
print(cumsumArr)

print("Cumulative Product")
cumpdtArr = np.cumprod(arr1)
print(cumpdtArr)

print("Discrete difference")
difArr = np.diff(arr1)
print(difArr)

print("Unique elements from the array")
uniqueArr = np.unique(arr1)
print(uniqueArr)
```

```
[2 4 7 6 4]
Cumulative Sum
[ 2  6 13 19 23]
Cumulative Product
[   2    8   56  336 1344]
Discrete difference
[ 2  3 -1 -2]
Unique elements from the array
[2 4 6 7]
```

Q4. Create two 1D array and perform the Addition using zip(), add() and user defined function (frompyfunc())

In [ ]:

```
import numpy as np
arr1 = np.random.randint(0,9,size=(10),dtype=int)
print(arr1)
arr2 = np.random.randint(0,9,size=(10),dtype=int)
```

```
print(arr2)

print("Using zip()")
zipArr = [x+y for x,y in zip(arr1,arr2)]
print(zipArr)

print("Using add()")
addArr = np.add(arr1,arr2)
print(addArr)


print("Using frompyfunc()")
def addUsingFun(x, y):
  return x+y
addUsingFun = np.frompyfunc(addUsingFun, 2, 1)
print(addUsingFun(arr1,arr2))
```

```
[0 6 4 5 1 0 4 5 1 0]
[1 3 6 4 8 3 6 5 5 8]
Using zip()
[1, 9, 10, 9, 9, 3, 10, 10, 6, 8]
Using add()
[ 1  9 10  9  9  3 10 10  6  8]
Using frompyfunc()
[1 9 10 9 9 3 10 10 6 8]
```

Q5. Find the LCM (Least Common Multiple) and GCD (Greatest Common Divisor) of an array of elements using reduce().

In [ ]:

```
import numpy as np
arr = np.random.randint(0,9,size=(5),dtype=int)
print(arr1)

gcdArr = np.gcd.reduce(arr)
print("GCD: ",gcdArr)

lcmArr = np.lcm.reduce(arr)
print("LCM: ",lcmArr)
```

```
[4 2 1 7 3]
GCD:  1
LCM:  12
```