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

1. Product (prod)
2. Multiplication (multiply)
3. Dot Product (dot)

In [ ]:
```
import numpy as np
matrix1 = np.random.rand(3, 3)
matrix2 = np.random.rand(3, 3)
product_result = np.prod(matrix1)
multiplication_result = np.multiply(matrix1, matrix2)
dot_product_result = np.dot(matrix1, matrix2)
print("Matrix 1:")
print(matrix1)
print("\nMatrix 2:")
print(matrix2)
print("\nProduct (prod) Result:")
print(product_result)
print("\nMultiplication (multiply) Result:")
print(multiplication_result)
print("\nDot Product (dot) Result:")
print(dot_product_result)
```

```
Matrix 1:
[[0.30633456 0.63857423 0.67352137]
 [0.83435997 0.83426022 0.75314701]
 [0.69623673 0.71921052 0.89603982]]

Matrix 2:
[[0.72500788 0.01526175 0.0395176 ]
 [0.98844533 0.34861276 0.04023338]
 [0.01560041 0.69696603 0.3138538 ]]

Product (prod) Result:
0.03099087159571356

Multiplication (multiply) Result:
[[0.22209497 0.00974576 0.02661595]
 [0.82471921 0.29083376 0.03030165]
 [0.01086158 0.5012653  0.28122551]]

Dot Product (dot) Result:
[[0.86379789 0.69671184 0.24918485]
 [1.44128757 0.82848544 0.30291506]
 [1.22965598 0.88586107 0.33767538]]
```

Perform the following set operations using the Numpy functions.

1. Union
2. Intersection
3. Set difference
4. XOR

In [ ]:
```
import numpy as np
array1 = np.array([1, 2, 3, 4, 5])
```

```
array2 = np.array([3, 4, 5, 6, 7])
union_result = np.union1d(array1, array2)
intersection_result = np.intersect1d(array1, array2)
difference_result = np.setdiff1d(array1, array2)
xor_result = np.setxor1d(array1, array2)
print("Array 1:", array1)
print("Array 2:", array2)
print("Union:", union_result)
print("Intersection:", intersection_result)
print("Set Difference (array1 - array2):", difference_result)
print("XOR (Symmetric Difference):", xor_result)
```

```
Array 1: [1 2 3 4 5]
Array 2: [3 4 5 6 7]
Union: [1 2 3 4 5 6 7]
Intersection: [3 4 5]
Set Difference (array1 - array2): [1 2]
XOR (Symmetric Difference): [1 2 6 7]
```

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

1. Cumulative sum
2. Cumulative Product
3. Discrete difference (with n=3)
4. Find the unique elements from the array

In [ ]:
```
import numpy as np
random_array = np.random.rand(10)
cumulative_sum = np.cumsum(random_array)
cumulative_product = np.cumprod(random_array)
discrete_difference = np.diff(random_array, n=3)
unique_elements = np.unique(random_array)
print("Random Array:")
print(random_array)
print("\nCumulative Sum:")
print(cumulative_sum)
print("\nCumulative Product:")
print(cumulative_product)
print("\nDiscrete Difference (n=3):")
print(discrete_difference)
print("\nUnique Elements:")
print(unique_elements)
```

```
Random Array:
[0.53278047 0.32905038 0.51184504 0.39023444 0.9256599  0.80746956
 0.85880056 0.23266297 0.45279329 0.76558776]

Cumulative Sum:
[0.53278047 0.86183085 1.37367589 1.76391033 2.68957023 3.49703979
 4.35584035 4.58850332 5.04129661 5.80688437]

Cumulative Product:
[0.53278047 0.17531162 0.08973238 0.03501667 0.03241352 0.02617293
 0.02247733 0.00522964 0.00236795 0.00181287]

Discrete Difference (n=3):
[-0.69093002  0.96144133 -1.31065187  0.82313713 -0.84698991  1.52373647
 -0.75360374]

Unique Elements:
[0.23266297 0.32905038 0.39023444 0.45279329 0.51184504 0.53278047
 0.76558776 0.80746956 0.85880056 0.9256599 ]
```

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

In [ ]:
```
import numpy as np
array1 = np.array([1, 2, 3, 4, 5])
array2 = np.array([10, 20, 30, 40, 50])
result_zip = np.array([a + b for a, b in zip(array1, array2)])
result_add = np.add(array1, array2)
def custom_add(x, y):
    return x + y
ufunc = np.frompyfunc(custom_add, 2, 1)
result_custom = ufunc(array1, array2)
print("Array 1:", array1)
print("Array 2:", array2)
print("Result using zip():", result_zip)
print("Result using np.add():", result_add)
print("Result using custom function:", result_custom)
```

```
Array 1: [1 2 3 4 5]
Array 2: [10 20 30 40 50]
Result using zip(): [11 22 33 44 55]
Result using np.add(): [11 22 33 44 55]
Result using custom function: [11 22 33 44 55]
```

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

In [ ]:
```
import numpy as np
elements = np.array([12, 18, 24, 36])
lcm_result = np.lcm.reduce(elements)
gcd_result = np.gcd.reduce(elements)
print("Array of Elements:", elements)
print("LCM:", lcm_result)
print("GCD:", gcd_result)
```

```
Array of Elements: [12 18 24 36]
LCM: 72
GCD: 6
```