akshay-206-lab7

September 4, 2023

[32]:

```
'''
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)
'''
import numpy as np
import random
matrix1 = np.random.randint(1,10,(3,3))
matrix2 = np.random.randint(1, 10,(3,3))
print(matrix1)
print(matrix2)
result1 = np.dot(matrix1, matrix2)
result2 = np.multiply(matrix1, matrix2)
result3 = np.product(matrix1)
print(result1)
print(result2)
print(result3)
```

```
[[4 3 4]
 [4 3 4]
 [1 5 9]]
[[8 4 1]
 [9 6 3]
 [7 5 7]]
[[ 87  54  41]
 [ 87  54  41]
 [116  79  79]]
[[32 12  4]
 [36 18 12]
 [ 7 25 63]]
103680
```

[45]:

```
'''
Perform the following set operations using the Numpy functions.
è Union
è Intersection
```

```
è Set difference
è XOR'''
set1 = np.array([1,2,3,4,5,6])
set2 = np.array([3,4,5,6,7,8])
union_res = np.union1d(set1, set2)
intersection_res = np.intersect1d(set1, set2)
diff_res = np.setdiff1d(set1, set2)
xor_res = np.setxor1d(set1, set2)
print("Union is ",union_res)
print("Instersection is ",intersection_res)
print("Difference is ",diff_res)
print("XOR is ",xor_res)
```

```
Union is  [1 2 3 4 5 6 7 8]
Instersection is  [3 4 5 6]
Difference is  [1 2]
XOR is  [1 2 7 8]
```

[62]:
```
'''
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'''
random_arr = np.random.randint(1,10,(1,10))
print(random_arr)
print("Cumulative Sum: ",np.cumsum(random_arr))
print("Cumulative Product: ",np.cumprod(random_arr))
print("The Unique: ",np.unique(random_arr))
print("Discrete: ",np.diff(random_arr, n=3))
```

```
[[2 3 4 2 4 9 4 1 7 8]]
Cumulative Sum:  [ 2  5  9 11 15 24 28 29 36 44]
Cumulative Product:  [     2      6     24     48    192   1728   6912   6912
 48384 387072]
The Unique:  [1 2 3 4 7 8 9]
Discrete:  [[ -3   7  -1 -13  12   7 -14]]
```

[64]:
```
'''Create two 1D array and perform the Addition using zip(), add() and user␣
 ↪defined function (frompyfunc())
'''
array1 = np.array([1, 2, 3, 4, 5])
array2 = np.array([6, 7, 8, 9, 10])

result_zip = np.array([a + b for a, b in zip(array1, array2)]) # Addition using␣
 ↪zip()
```

```
result_np_add = np.add(array1, array2)# Addition using np.add()


def custom_add(x, y): # Define a user-defined addition function
    return x + y

ufunc_custom_add = np.frompyfunc(custom_add, 2, 1) #user defined function

# Addition using the user-defined function
result_custom_add = ufunc_custom_add(array1, array2)

print("Array 1:", array1)
print("Array 2:", array2)
print("Addition using zip():", result_zip)
print("Addition using np.add():", result_np_add)
print("Addition using user-defined function:", result_custom_add)
```

```
Array 1: [1 2 3 4 5]
Array 2: [ 6  7  8  9 10]
Addition using zip(): [ 7  9 11 13 15]
Addition using np.add(): [ 7  9 11 13 15]
Addition using user-defined function: [7 9 11 13 15]
```

[63]:

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

from functools import reduce
from math import gcd, lcm

# Create an array of elements
elements = [12, 18, 24, 36, 48]


lcm_result = reduce(lambda x, y: lcm(x, y), elements) #LCM using reduce()


gcd_result = reduce(lambda x, y: gcd(x, y), elements) # Calculate the GCD␣
 ↪(Greatest Common Divisor) using reduce()
print("LCM (Least Common Multiple):", lcm_result)
print("GCD (Greatest Common Divisor):", gcd_result)
```

```
LCM (Least Common Multiple): 144
GCD (Greatest Common Divisor): 6
```