

## Lab - 20

**Student Name: Navneet P**

**Student ID: AF0411619**

### **Topic – Numpy Mathematical Functions**

Functions/Concept Used in the Assignment:

1. **Numpy Array Addition:** in python, using numpy you can simply add 2 different arrays using the + operator or the Numpy.add() function.
2. **Numpy Array Subtraction:** in python, using numpy you can simply subtrat 2 different arrays using the - operator or the Numpy.subtract() function.
3. **Numpy.multiply():** in python, using numpy you can simply multiply 2 different arrays using this function.
4. **Numpy.where():** This function is used to conditionally evaluate an array and store the result in a separate array. The result involves the indices of the elements which satisfy the condition.

Q1. Calculate the total revenue generated by two product categories in a store

Input: category1\_revenue = np.array([500, 600, 700, 550])

category2\_revenue = np.array([450, 700, 800, 600])

Solution:

```
Question 1

import numpy as np # Import the NumPy library for numerical operations

# Define revenue data for two categories as NumPy arrays
category1_revenue = np.array([500, 600, 700, 550]) # Revenue data for category 1
category2_revenue = np.array([450, 700, 800, 600]) # Revenue data for category 2

# Print the revenue data for both categories
print("Revenue in category 1: ", category1_revenue) # Display revenue for category 1
print("Revenue in category 2: ", category2_revenue) # Display revenue for category 2

# Calculate total revenue by adding the revenue from both categories
total_revenue = category1_revenue + category2_revenue # Element-wise addition of the two arrays

# Print the total revenue
print("\nTotal Revenue: \n", total_revenue) # Display the total revenue
```

Output:

```
Revenue in category 1: [500 600 700 550]
Revenue in category 2: [450 700 800 600]

Total Revenue:
[ 950 1300 1500 1150]
```

Q2. Calculate the profit made by a company

Input: revenue = np.array([10000, 12000, 11000, 10500])  
expenses = np.array([4000, 5000, 4500, 4800])

Solution:

```
Question 2

import numpy as np # Import the NumPy library for numerical operations

# Define revenue and expenses data as NumPy arrays
revenue = np.array([10000, 12000, 11000, 10500]) # Revenue data for different time periods

expenses = np.array([4000, 5000, 4500, 4800]) # Expense data for the same time periods

# Print the revenue and expenses data
print("Calculated Revenue: ", revenue) # Display calculated revenue for each period
print("Total Expenses: ", expenses) # Display total expenses for each period

# Calculate profit by subtracting expenses from revenue
profit = revenue - expenses # Element-wise subtraction to find profit for each period

# Print the calculated profit
print("\nProfit Earned: \n", profit) # Display the profit earned for each period
```

Output:

```
Calculated Revenue: [10000 12000 11000 10500]
Total Expenses: [4000 5000 4500 4800]

Profit Earned:
[6000 7000 6500 5700]
```

Q3. Determine which products in a store are out of stock (quantity is 0).

Input: inventory = np.array([10, 0, 5, 0, 20, 0])

Solution:

```
Question 3

import numpy as np # Import the NumPy library for numerical operations

# Define inventory data as a NumPy array
inventory = np.array([10, 0, 5, 0, 20, 0]) # Inventory levels for different products
print("Given Inventory: \n", inventory) # Print the initial inventory levels
print() # Print a newline for better formatting

# Identify the indices of products that are out of stock (inventory equals zero)
ind_no_stock = np.where(inventory == 0) # Get indices where inventory is zero
no_stock = inventory[ind_no_stock] # Extract the inventory levels for out-of-stock products

# Loop through the indices of out-of-stock products and print a message for each
for i in ind_no_stock[0]: # Iterate over the indices of out-of-stock products
    print(f"Product {i + 1} out of stock..") # Print which product is out of stock (1-based index)

# Print the final inventory levels of out-of-stock products
print("\nFinal Out of stock inventory:\n", no_stock) # Display the inventory levels of products that are out of stock
```

Output:

```
Given Inventory:
[10  0  5  0 20  0]

Product 2 out of stock..
Product 4 out of stock..
Product 6 out of stock..

Final Out of stock inventory:
[0 0 0]
```

Q4. Calculate the total cost of items in a shopping cart, considering the quantity and price per item.

Input: quantity = np.array([2, 3, 4, 1])

price\_per\_item = np.array([10.0, 5.0, 8.0, 12.0])

Solution:

```
Question 4

import numpy as np # Import the NumPy library for numerical operations

# Define quantity and price per item as NumPy arrays
quantity = np.array([2, 3, 4, 1]) # Quantities of each item
price_per_item = np.array([10.0, 5.0, 8.0, 12.0]) # Price per item for each corresponding quantity

# Print the quantities and prices per item
print("Quantities: ", quantity) # Display the quantity of each item
print("Price / item: ", price_per_item) # Display the price of each item

# Calculate total price for each item by multiplying quantity and price per item
total = np.multiply(quantity, price_per_item) # Element-wise multiplication to get total prices

# Iterate over the total prices and print the total for each item
for i, tot in enumerate(total): # Loop through the total prices using enumerate for indexing
    print(f"Item {i + 1}") # Print the item number (1-based index)
    print(f"Total price: Rs. {tot}\n") # Print the total price for the current item
```

Output:

```
Quantities: [2 3 4 1]
Price / item: [10.  5.  8. 12.]
Item 1
Total price: Rs. 20.0

Item 2
Total price: Rs. 15.0

Item 3
Total price: Rs. 32.0

Item 4
Total price: Rs. 12.0
```