**Practical – 2**

**Aim :**

1. **Create a 10x10 array with random values and find the minimum and maximum values**

**Program :**import numpy as np

import random

arr = np.random.rand(10,10)

print(f"Maximum of given array is {np.max(arr)}.")

print(f"Minimum of given array is {np.min(arr)}.")

**Output :**

****

1. **Write a NumPy program to create a 8x8 matrix and fill it with a checkerboard pattern.**

**Checkerboard pattern:**

**[[0 1 0 1 0 1 0 1]**

**[1 0 1 0 1 0 1 0]**

**[0 1 0 1 0 1 0 1]**

**[1 0 1 0 1 0 1 0]**

**[0 1 0 1 0 1 0 1]**

**[1 0 1 0 1 0 1 0]**

**[0 1 0 1 0 1 0 1]**

**[1 0 1 0 1 0 1 0]]**

**Program :**

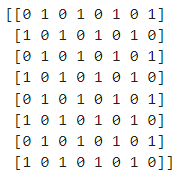
checker\_board = np.zeros((8,8),dtype=int)

checker\_board[1::2,::2] = 1

checker\_board[::2,1::2] = 1

print(checker\_board)

**Output :**

****

1. **Write a NumPy program to append values to the end of an array.**

**Expected Output: Original array: [10, 20, 30]**

**Program :**

arr1 = np.array([int(i) for i in input("Enter values for array - 1 (Space-Separated) : ").split(' ')])

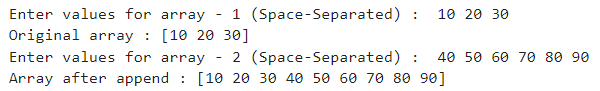
print(f"Original array : {arr1}")

arr2 = np.array([int(i) for i in input("Enter values for array - 2 (Space-Separated) : ").split(' ')])

arr = np.append(arr1,arr2)

print(f"Array after append : {arr}")

**Output :**

****

1. **Write a NumPy program to calculate the arithmetic means of corresponding elements of two given arrays of same size.**

**Program :**

array1 = np.array([float(i) for i in input("Enter values for array - 1 (Space-Separated) : ").split(' ')])

array2 = np.array([float(i) for i in input("Enter values for array - 2 (Space-Separated) : ").split(' ')])

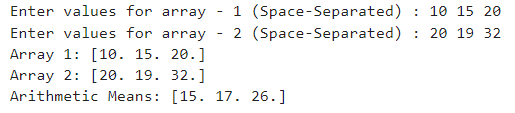
mean\_array = (array1 + array2) / 2

print("Array 1:", array1)

print("Array 2:", array2)

print("Arithmetic Means:", mean\_array)

**Output :**

****

1. **Write a Numpy program to test whether a numpy array is faster than a Python list or not.**

**Program :**

import time

my\_list = [np.random.randint(1000) for i in range(10\*\*5)]

my\_array = np.arange(10\*\*5)

start\_time = time.time()

np.sort(my\_array)

end\_time = time.time()

print(f"Time taken for sorting array is {end\_time - start\_time}.")

start\_time = time.time()

my\_list.sort()

end\_time = time.time()

print(f"Time taken for sorting list is {end\_time - start\_time}.")

**Output :**

****

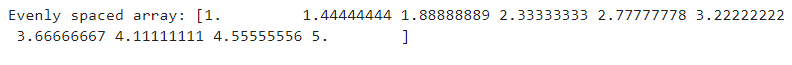
1. **Write a Python function to create an array of 10 values from 1 to 5 evenly spaced.**

**Program :**

array = np.linspace(1, 5, 10)

print("Evenly spaced array:", array)

**Output :**

****

1. **Given a 1D array, negate all elements which are between 3 and 8, in place.**

**Program :**

arr = np.array([float(i) for i in input("Enter values for Array (Space-Separated) : ").split(' ')])

print(f"Original array : {arr}")

indices = np.where((arr > 3) & (arr < 8))

arr[indices] = -arr[indices]

print(f"Array after negation : {arr}")

**Output :**

****

1. **How to compute ((A+B)\*(-A/2)) in place (without copy)? where A = [1. 1. 1.] and B=[2. 2. 2.]**

**Program :**

a = np.ones(3)\*1

b = np.ones(3)\*2

np.add(a,b,out=b)

np.divide(a,2,out=a)

np.negative(a,out=a)

np.multiply(a,b,out=a)

**Output :**

****

1. **Create random vector of size 10 and replace the maximum value by 0.**

**Program :**

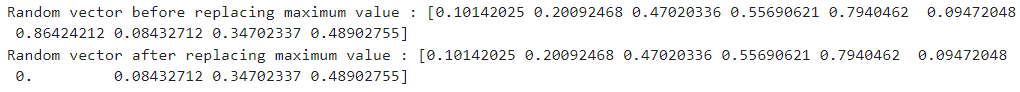
random\_vector = np.random.rand(10)

print(f"Random vector before replacing maximum value : {random\_vector}")

random\_vector[random\_vector == np.max(random\_vector)] = 0

print(f"Random vector after replacing maximum value : {random\_vector}")

**Output :**

****

**10. Write a NumPy program to subtract the mean of each row of a given matrix.**

**Program :**

size = int(input("Enter the size of the array : "))

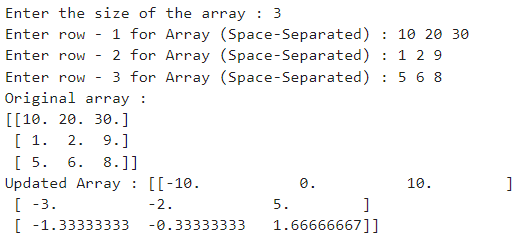
arr = np.array([[float(j) for j in input(f"Enter row - {i + 1} for Array (Space-Separated) : ").split(' ')] for i in range(size)])

print(f"Original array : \n{arr}")

arr = arr - np.mean(arr,axis=1,keepdims=True)

print(f"Updated Array : {arr}")

**Output :**

****