Indian Institute of Technology Mandi

भारतीय प्रौद्योगिकी संस्थान मण्डी

IC252-Data Science 2

Assignment– 0

Question 1: Find out the size of array 'a', 'b', and 'c'. In the following code block.

```
import numpy as np

a = np.array(42)
b = np.array([1, 2, 3, 4, 5])
c = np.array([[1, 2, 3], [4, 5, 6]])
# Print size of all arrays
```

Question 2: Bins! Yes binning might be tricky if you want to plot meaningful plots, in the below histogram choose an appropriate binning size, expiriment with the bins style, location, borders and colors. Explore the np.arange function. Add meaningful labels to both axes.

```
import matplotlib.pyplot as plt
val=np.random.normal(size=(100), scale=3, loc=10)

# bins=np.arange(0,20,1)

plt.hist(val,bins=20)
plt.show()
```

Question 3: Array adressing and slicing. As the dimensions of the array increase it becomes very important how one indexes those arrays, slicing becomes important when one wants to get some desired subpart of a whole array. This Question has two parts.

3.1. Print the sum of left diagonal and right diagonal of a square matrix.

```
mat= np.array([[1,3,4,5,2],[1,5,2,4,3],[5,2,3,4,1,],[1,4,2,6,9],[4,5,2,1,7]])

def left_diagonal_sum(mat:np.ndarray)->float:
    #write your logic here
    pass


def right_diagonal_sum(mat:np.ndarray)->float:
    #write your logic here
    pass

print(f'Left Diagonal Sum of {mat=} is {left_diagonal_sum(mat)}')
print(f'Left Diagonal Sum of {mat=} is {right_diagonal_sum(mat)}')
```

3.2. From the matrix of previous part, print the 3x4 sub-matrix from 1st row 2nd column to 3rd row 5th column. Make use of the array slicing operations preferably , use of for loops is discouraged.

```
def submatrix_3x4(mat: np.ndarray)-> np.ndarray:
    #write your logic here
    pass

print(f'The desired submatrix of \n {mat} is \n {submatrix_3x4(mat)}')')
```

Question 4: array operations in np.array and np math functions. Plot the function:

$$f(x) = \frac{sin^7(x) + cos^5(x)}{e^x}$$

in the domain:

$$x \in [0, 4].$$

Do not use for loops for the same.

```python
def f(x:np.ndarray)->np.ndarray:
    # write your logic here
    pass

steps=1000 # experiment how step size matters in the graph shape
x= np.arange(0,4, steps)
y= f(x)
plt.plot(x,y) # label the axes, and choose appropriate plot format and style
plt.show()
```

Question 5: Create a subplot with two plots side by side. Plot a sine wave in the first subplot. Plot a cosine wave in the second subplot. Add labels, titles, and a legend to the plots.