COGNIZANCE CLUB

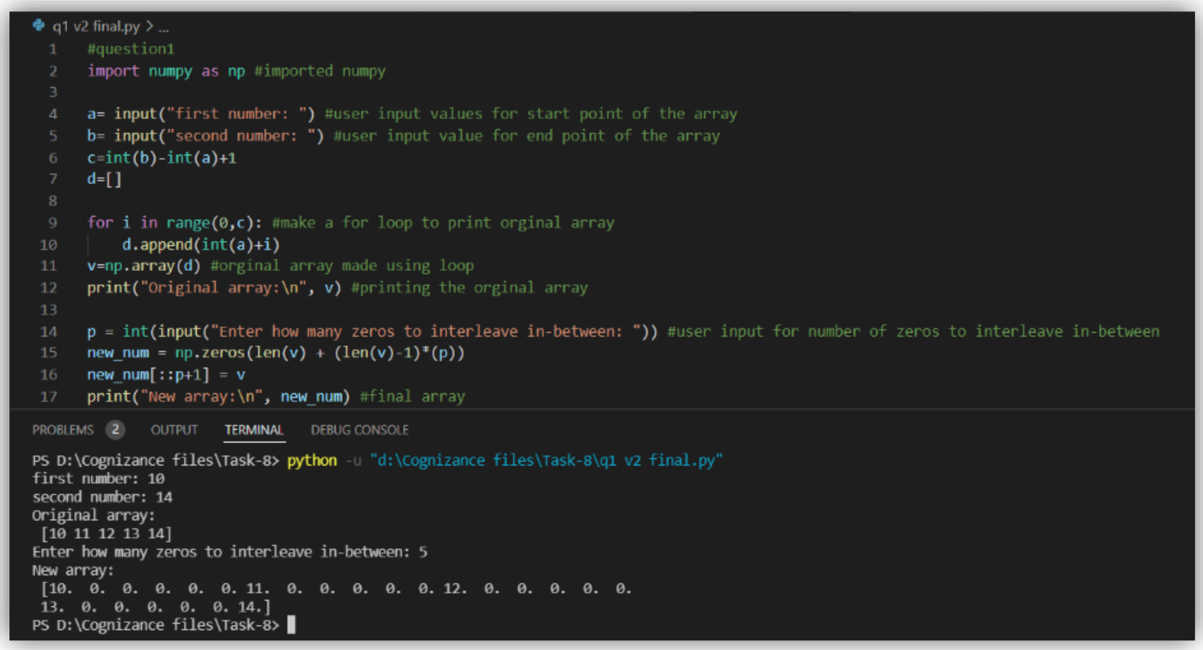
TASK-8 [AI, NumPy, Pandas, Python]

[PYTHON – MEDICORE LVL]

BY

Sakthivel V

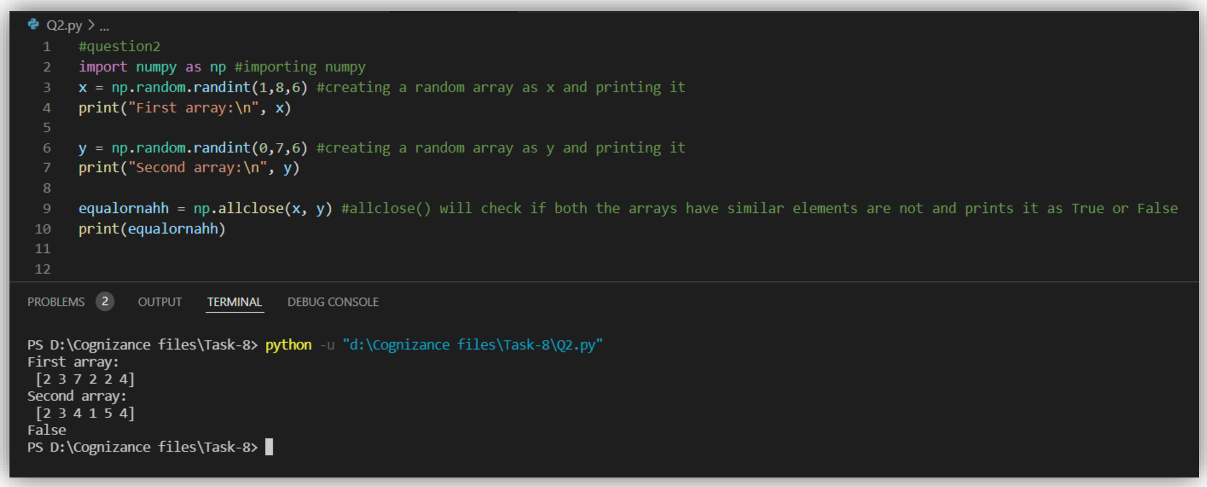
1) Consider the vector [10, 11, 12, 13, 14], how to build a new vector with 5 consecutive zeros interleaved between each value?



```
#question1
import numpy as np #imported numpy

a= input("first number: ") #user input values for start point of the array
b= input("second number: ") #user input value for end point of the array
c=int(b)-int(a)+1
d=[]

for i in range(0,c): #make a for loop to print orginal array
    d.append(int(a)+i)
v=np.array(d) #orginal array made using loop
print("Original array:\n", v) #printing the orginal array

p = int(input("Enter how many zeros to interleave in-between: ")) #user input for number of zeros to interleave in-between
new_num = np.zeros(len(v) + (len(v)-1)*(p))
new_num[::p+1] = v
print("New array:\n", new_num) #final array
```

```
PS D:\Cognizance files\Task-8> python -u "d:\Cognizance files\Task-8\q1 v2 final.py"
first number: 10
second number: 14
Original array:
 [10 11 12 13 14]
Enter how many zeros to interleave in-between: 5
New array:
 [10.  0.  0.  0.  0.  0. 11.  0.  0.  0.  0.  0. 12.  0.  0.  0.  0.  0.
 13.  0.  0.  0.  0.  0. 14.]
PS D:\Cognizance files\Task-8> 
```

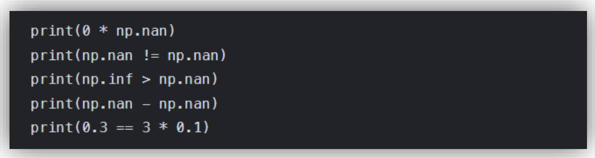
2) Consider two random array A and B, check if they are equal.



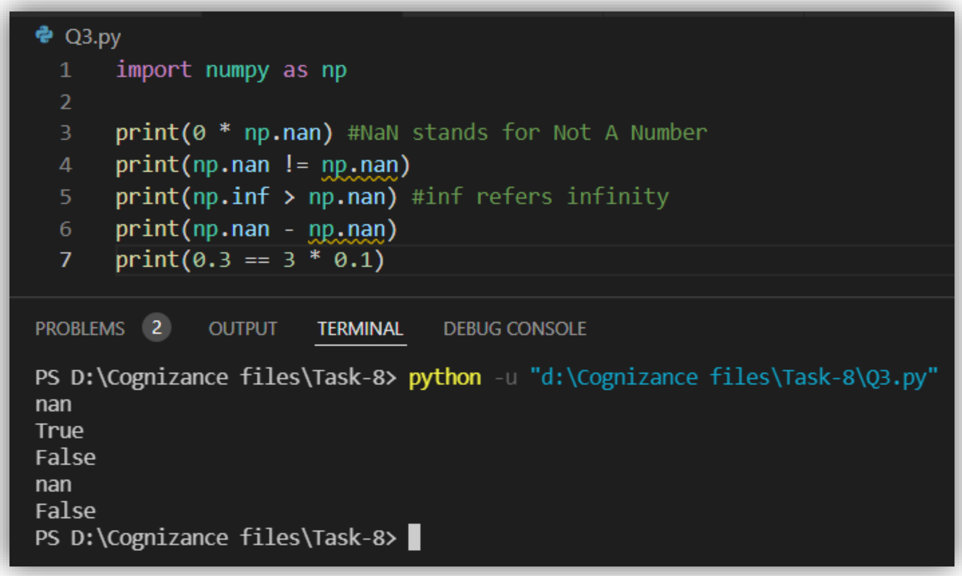
```
#question2
import numpy as np #importing numpy
x = np.random.randint(1,8,6) #creating a random array as x and printing it
print("First array:\n", x)

y = np.random.randint(0,7,6) #creating a random array as y and printing it
print("Second array:\n", y)

equalornahh = np.allclose(x, y) #allclose() will check if both the arrays have similar elements are not and prints it as True or False
print(equalornahh)


```

```
PS D:\Cognizance files\Task-8> python -u "d:\Cognizance files\Task-8\Q2.py"
First array:
 [2 3 7 2 2 4]
Second array:
 [2 3 4 1 5 4]
False
PS D:\Cognizance files\Task-8> 
```

3) What is the result of the following expression?



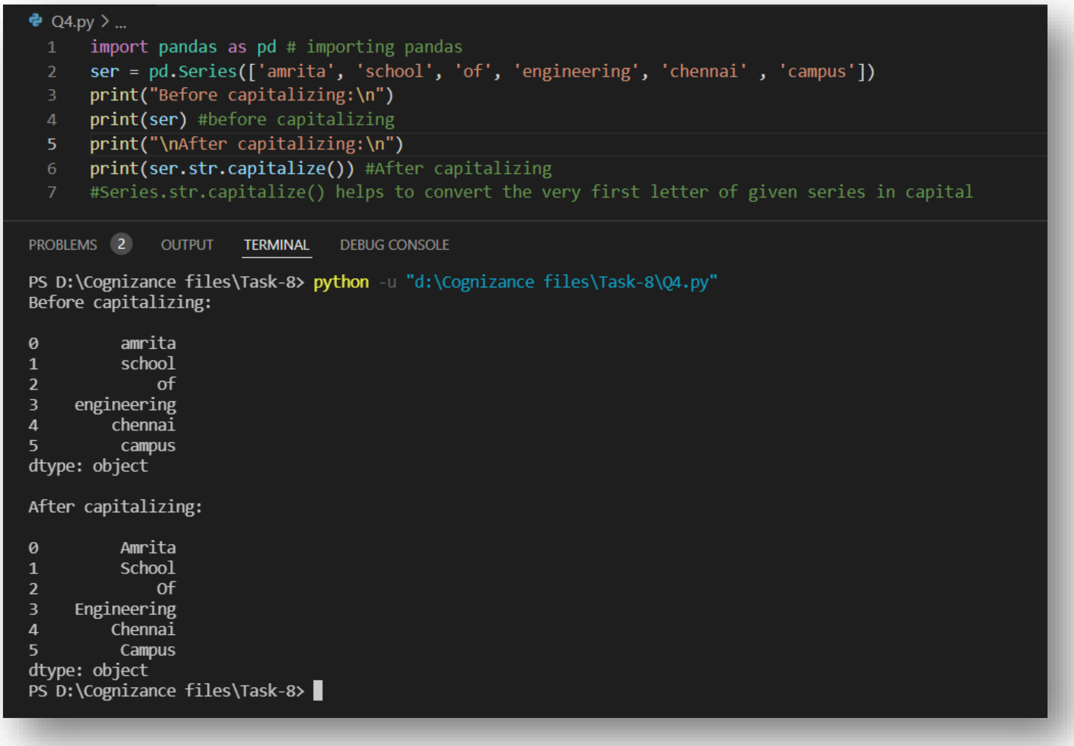
```
print(0 * np.nan)
print(np.nan != np.nan)
print(np.inf > np.nan)
print(np.nan - np.nan)
print(0.3 == 3 * 0.1)
```



```
import numpy as np

print(0 * np.nan) #NaN stands for Not A Number
print(np.nan != np.nan)
print(np.inf > np.nan) #inf refers infinity
print(np.nan - np.nan)
print(0.3 == 3 * 0.1)
```

```
PS D:\Cognizance files\Task-8> python -u "d:\Cognizance files\Task-8\Q3.py"
nan
True
False
nan
False
PS D:\Cognizance files\Task-8>
```

4) Convert the first character of each element in a series to uppercase?

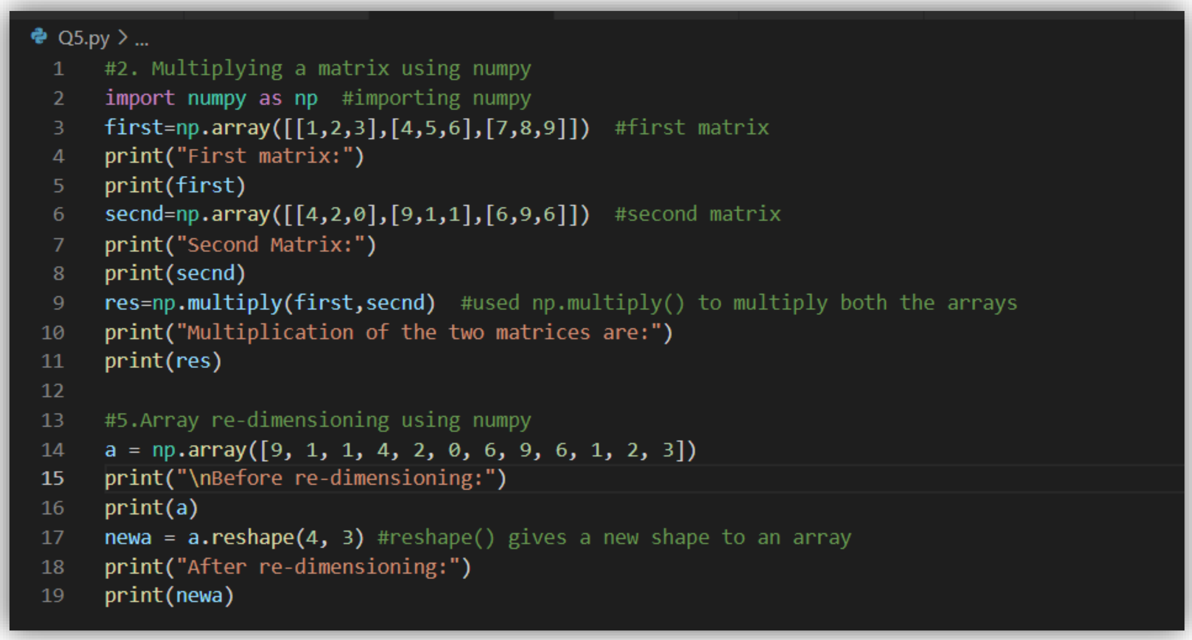


```
import pandas as pd # importing pandas
ser = pd.Series(['amrita', 'school', 'of', 'engineering', 'chennai' , 'campus'])
print("Before capitalizing:\n")
print(ser) #before capitalizing
print("\nAfter capitalizing:\n")
print(ser.str.capitalize()) #After capitalizing
#Series.str.capitalize() helps to convert the very first letter of given series in capital
```

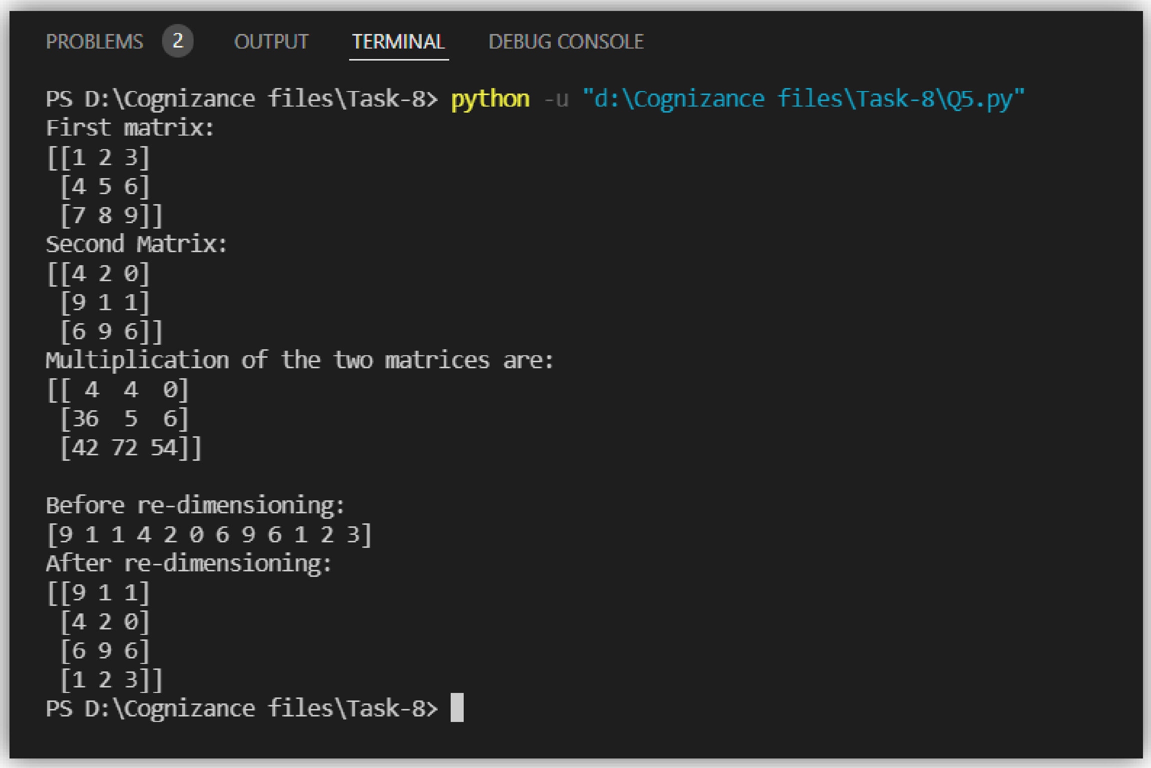
```
PS D:\Cognizance files\Task-8> python -u "d:\Cognizance files\Task-8\Q4.py"
Before capitalizing:

0         amrita
1         school
2             of
3    engineering
4        chennai
5         campus
dtype: object

After capitalizing:

0         Amrita
1         School
2             Of
3    Engineering
4        Chennai
5         Campus
dtype: object
PS D:\Cognizance files\Task-8>
```

5) Do any two Exercises using numpy.

- Multiplying a matrix using numpy
- Array re-dimensioning using numpy



```python
#2. Multiplying a matrix using numpy
import numpy as np  #importing numpy
first=np.array([[1,2,3],[4,5,6],[7,8,9]])  #first matrix
print("First matrix:")
print(first)
secnd=np.array([[4,2,0],[9,1,1],[6,9,6]])  #second matrix
print("Second Matrix:")
print(secnd)
res=np.multiply(first,secnd)  #used np.multiply() to multiply both the arrays
print("Multiplication of the two matrices are:")
print(res)

#5.Array re-dimensioning using numpy
a = np.array([9, 1, 1, 4, 2, 0, 6, 9, 6, 1, 2, 3])
print("\nBefore re-dimensioning:")
print(a)
newa = a.reshape(4, 3) #reshape() gives a new shape to an array
print("After re-dimensioning:")
print(newa)
```



```
PS D:\Cognizance files\Task-8> python -u "d:\Cognizance files\Task-8\Q5.py"
First matrix:
[[1 2 3]
 [4 5 6]
 [7 8 9]]
Second Matrix:
[[4 2 0]
 [9 1 1]
 [6 9 6]]
Multiplication of the two matrices are:
[[ 4  4  0]
 [36  5  6]
 [42 72 54]]

Before re-dimensioning:
[9 1 1 4 2 0 6 9 6 1 2 3]
After re-dimensioning:
[[9 1 1]
 [4 2 0]
 [6 9 6]
 [1 2 3]]
PS D:\Cognizance files\Task-8>
```