Essentials Of Data Science
Practical No - 3

(1) Name : Varun Balbudhe
(2) Roll No: 503 E1
(3) PRN No :202201040079

Code:

```
import numpy as np
arr1=np.loadtxt(r"C:\Users\\varun\\Downloads\\testmarks1.csv",delimiter=",",dtype=str)
arr2=np.loadtxt(r"C:\Users\\varun\\Downloads\\testmarks2.csv",delimiter=",",dtype=str)

# Now deleting the first row from both the file a1 and a2.
  a1=np.delete(arr1,0,axis=0)
  a2=np.delete(arr2,0,axis=0)

  # Now all left up Numbers Converted to (float).
  array1 = a1.astype(float)
  array2 = a2.astype(float)

  # Printing the array 1 and array 2.
  print("Array 1 : \n",array1,"\n")
  print("Array 2 : \n",array2)

  # printing the shape of the both arrays.
  print("\n\nShape : Array_1 : ",array1.shape)
  print("\nShape : Array_2 : ",array2.shape)


  # Performing Arithmetic Operations On Both Array.
  print("\n\n",np.empty_like(array1).reshape(10,5))

  print("\nAdding Two Array :\n",np.add(array1,array2))
  print("\nSubtracting Two Array :\n",np.subtract(array1,array2))
  print("\nMultiplication Two Array :\n",np.multiply(array1,array2))
  print("\nDividing Two Array :\n",np.divide(array2,array1),"\n\n")

  # Vertical Stacking Two Arrays:
  print("Vertical Stacking :\n\n",np.vstack((array1,array2)))

  # Horizontal Stacking Two Arrays:
  print("\n\nHorizontal Stacking :\n\n",np.hstack((array1,array2)))

  # max And min Operations on Both Array 1 and 2.
  print("\nMax :",np.max(array1,1),"\nMin :",(np.min(array1,0)))
```

```
  print("\nMax In Array_2 : ",np.max(array2))
  print("\nMin In Array_2 : ",np.min(array2),"\n")


  print(np.sin(array1*np.pi/180))
print("\n\n")

# Converting The ara1 and ara2 to int type.
ara1 = array1.astype(int)
ara2 = array2.astype(int)

ar1=np.array(ara1,dtype=np.uint8)
print("\n\nPrinting The Array_1 (Int) : \n",ar1)
ar2=np.array(ara2,dtype=np.uint8)
print("\n\nPrinting The Array_2 (Int) : \n",ar2)


print(np.bitwise_and(ar1,ar2))
array_view=ara1.view()
print("\n\nPrinting Array (View) : \n",array_view)
array_copy=ara2.copy()
print("\n\nPrinting Array (Copy) : \n",array_copy)


print("\n\nSorting The Array(Copy) : ",np.sort(array_copy))
print("\n\nPrinting The Non-Zero in Array : ",np.nonzero(array_view>50))

print("\n\nPrinting All values Greater Than 100 : ",np.where(array_copy>100))
arr_=np.array([-1])
res=arr_*array1
print("\n\nPrinting : \n",res)
```

Output

```
Microsoft Windows [Version 10.0.22621.1778]
(c) Microsoft Corporation. All rights reserved.

C:\Users\varun\OneDrive\Desktop\Code-Play\Ipynb>python -u "c:\Users\varun\OneDrive\Desktop\Code-Play\PYTHON\EDS\EDS_3.py"
Array 1 :
 [[801.    43.05  27.79  28.7   27.79]
 [802.    43.47  28.52  28.98  27.89]
 [803.    42.24  28.16  28.16  25.63]
 [804.    39.24  26.16  26.16  26.16]
 [805.    40.9   26.03  27.27  25.65]
 [806.    39.47  26.31  26.31  25.21]
 [807.    41.68  25.63  27.79  25.46]
 [808.    42.19  27.61  28.13  26.21]
 [809.    44.75  28.35  29.83  28.21]
 [810.    46.95  28.88  31.3   28.53]]

Array 2 :
 [[801.    28.48  34.18  30.56  22.23]
 [802.    28.1   33.72  30.68  22.82]
 [803.    26.16  31.39  28.2   22.53]
 [804.    26.16  31.39  28.78  20.93]
 [805.    26.1   31.32  28.22  20.82]
 [806.    25.45  30.54  27.73  21.05]
 [807.    26.16  31.39  28.01  20.51]
 [808.    27.44  32.93  28.83  22.08]
 [809.    28.63  34.35  31.03  22.68]
 [810.    30.35  36.42  31.38  23.1 ]]


Shape : Array_1 :  (10, 5)

Shape : Array_2 :  (10, 5)
```


```
 [[801.    28.48  34.18  30.56  22.23]
 [802.    28.1   33.72  30.68  22.82]
 [803.    26.16  31.39  28.2   22.53]
 [804.    26.16  31.39  28.78  20.93]
 [805.    26.1   31.32  28.22  20.82]
 [806.    25.45  30.54  27.73  21.05]
 [807.    26.16  31.39  28.01  20.51]
 [808.    27.44  32.93  28.83  22.08]
 [809.    28.63  34.35  31.03  22.68]
 [810.    30.35  36.42  31.38  23.1 ]]
```

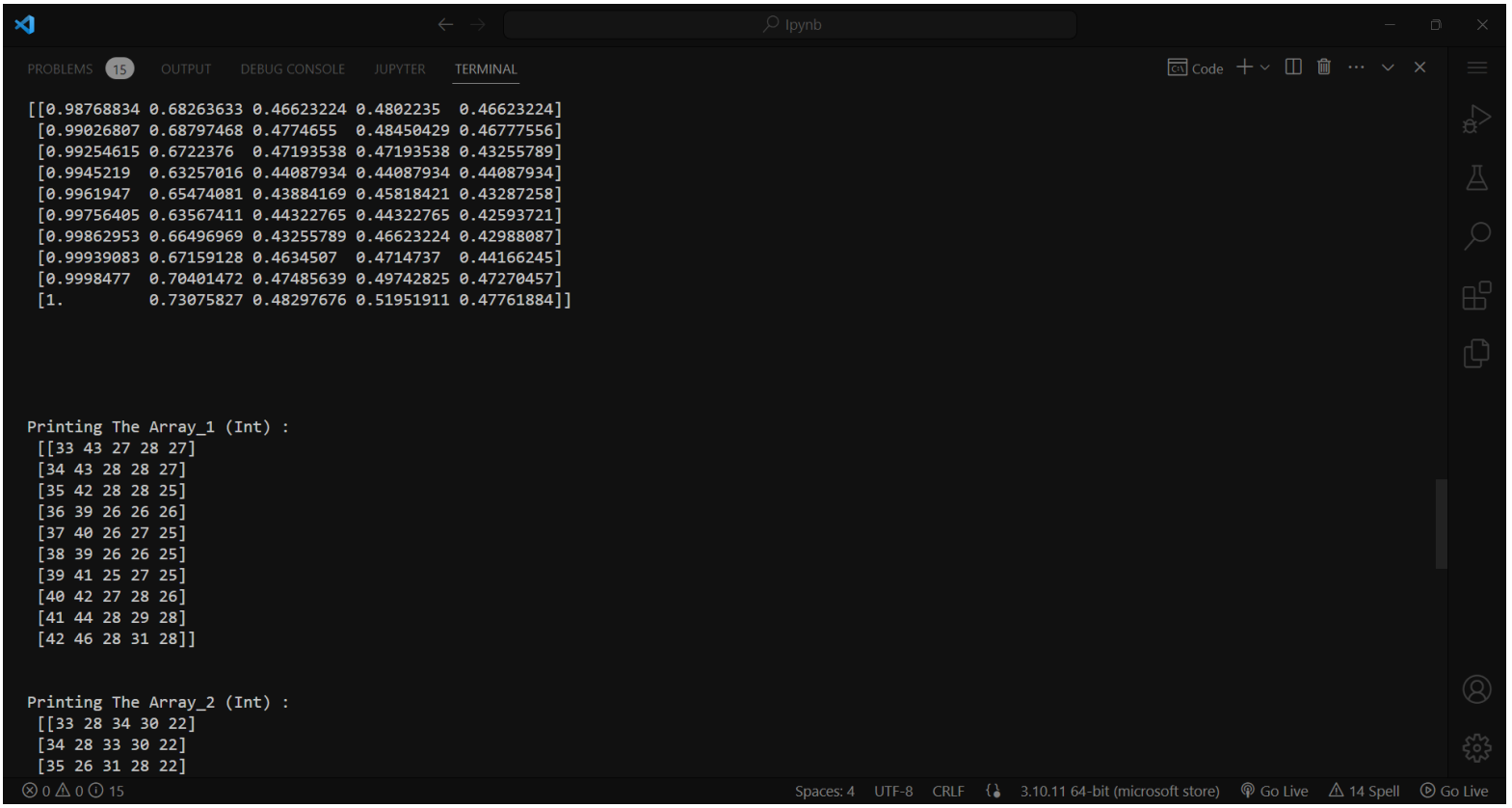
```
Adding Two Array :
 [[1602.     71.53   61.97   59.26   50.02]
 [1604.     71.57   62.24   59.66   50.71]
 [1606.     68.4    59.55   56.36   48.16]
 [1608.     65.4    57.55   54.94   47.09]
 [1610.     67.     57.35   55.49   46.47]
 [1612.     64.92   56.85   54.04   46.26]
 [1614.     67.84   57.02   55.8    45.97]
 [1616.     69.63   60.54   56.96   48.29]
 [1618.     73.38   62.7    60.86   50.89]
 [1620.     77.3    65.3    62.68   51.63]]

Subtracting Two Array :
 [[ 0.   14.57 -6.39 -1.86  5.56]
 [ 0.   15.37 -5.2  -1.7   5.07]
 [ 0.   16.08 -3.23 -0.04  3.1 ]
 [ 0.   13.08 -5.23 -2.62  5.23]
 [ 0.   14.8  -5.29 -0.95  4.83]
 [ 0.   14.02 -4.23 -1.42  4.16]
 [ 0.   15.52 -5.76 -0.22  4.95]
 [ 0.   14.75 -5.32 -0.7   4.13]
```

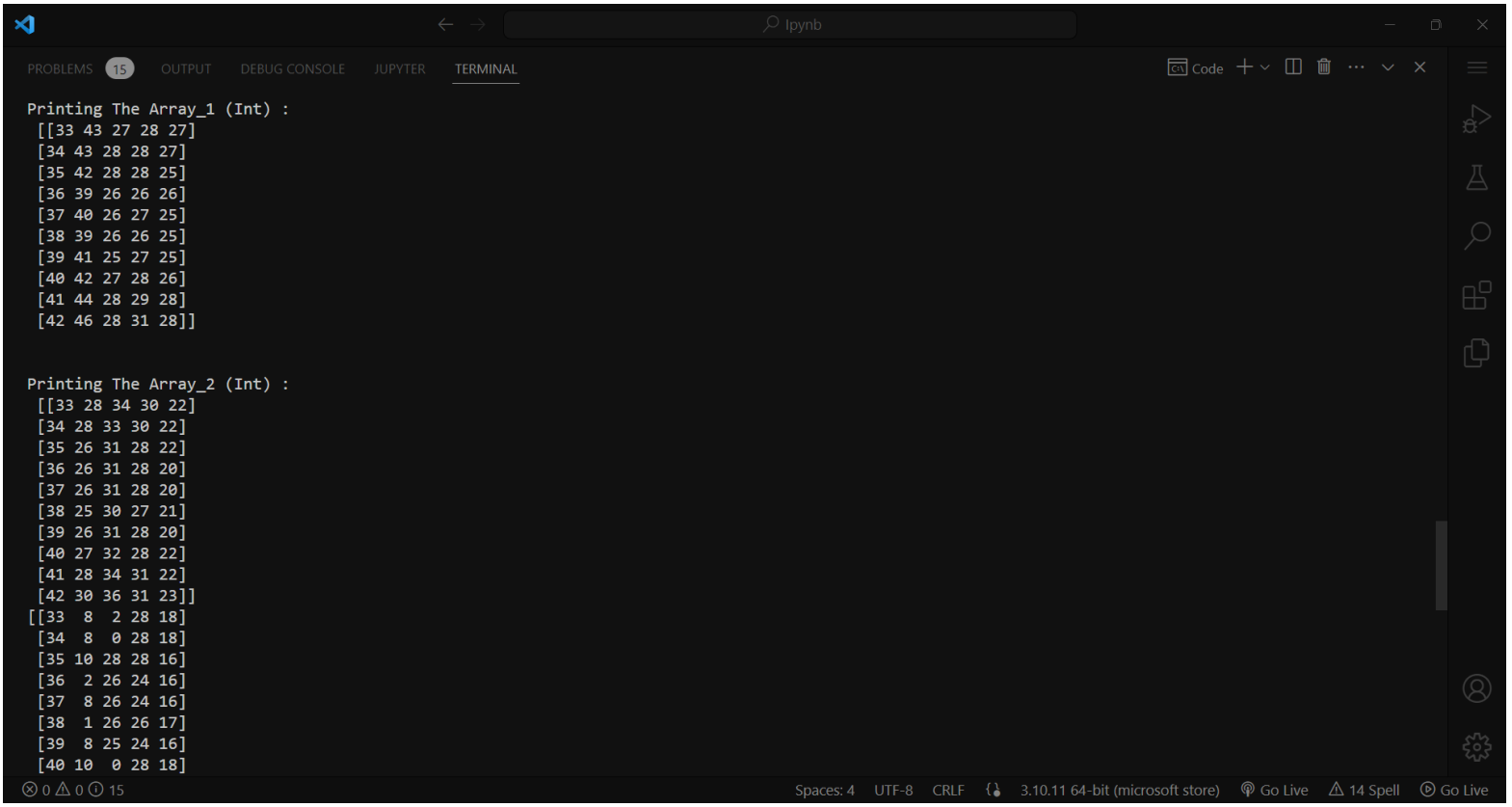
```
 [ 0.   13.08 -5.23 -2.62  5.23]
 [ 0.   14.8  -5.29 -0.95  4.83]
 [ 0.   14.02 -4.23 -1.42  4.16]
 [ 0.   15.52 -5.76 -0.22  4.95]
 [ 0.   14.75 -5.32 -0.7   4.13]
 [ 0.   16.12 -6.   -1.2   5.53]
 [ 0.   16.6  -7.54 -0.08  5.43]]

Multiplication Two Array :
 [[6.4160100e+05 1.2260640e+03 9.4986220e+02 8.7707200e+02 6.1777170e+02]
 [6.4320400e+05 1.2215070e+03 9.6169440e+02 8.8910640e+02 6.3644980e+02]
 [6.4480900e+05 1.1049984e+03 8.8394240e+02 7.9411200e+02 5.7744390e+02]
 [6.4641600e+05 1.0265184e+03 8.2116240e+02 7.5288480e+02 5.4752880e+02]
 [6.4802500e+05 1.0674900e+03 8.1525960e+02 7.6955940e+02 5.3403300e+02]
 [6.4963600e+05 1.0045115e+03 8.0350740e+02 7.2957630e+02 5.3067050e+02]
 [6.5124900e+05 1.0903488e+03 8.0452570e+02 7.7839790e+02 5.2218460e+02]
 [6.5286400e+05 1.1576936e+03 9.0919730e+02 8.1098790e+02 5.7871680e+02]
 [6.5448100e+05 1.2811925e+03 9.7382250e+02 9.2562490e+02 6.3980280e+02]
 [6.5610000e+05 1.4249325e+03 1.0518096e+03 9.8219400e+02 6.5904300e+02]]

Dividing Two Array :
 [[1.         0.66155633 1.22993883 1.06480836 0.79992803]
 [1.         0.64642282 1.18232819 1.05866115 0.81821441]
 [1.         0.61931818 1.1147017  1.00142045 0.87904799]
 [1.         0.66666667 1.19992355 1.10015291 0.80007645]
 [1.         0.63814181 1.20322705 1.03483682 0.81169591]
 [1.         0.64479351 1.16077537 1.05397187 0.83498612]
 [1.         0.62763916 1.22473664 1.00791652 0.80557738]
 [1.         0.65039109 1.19268381 1.02488446 0.84242655]
 [1.         0.63977654 1.21164021 1.04022796 0.80397022]
 [1.         0.64643237 1.26108033 1.00255591 0.80967403]]
```

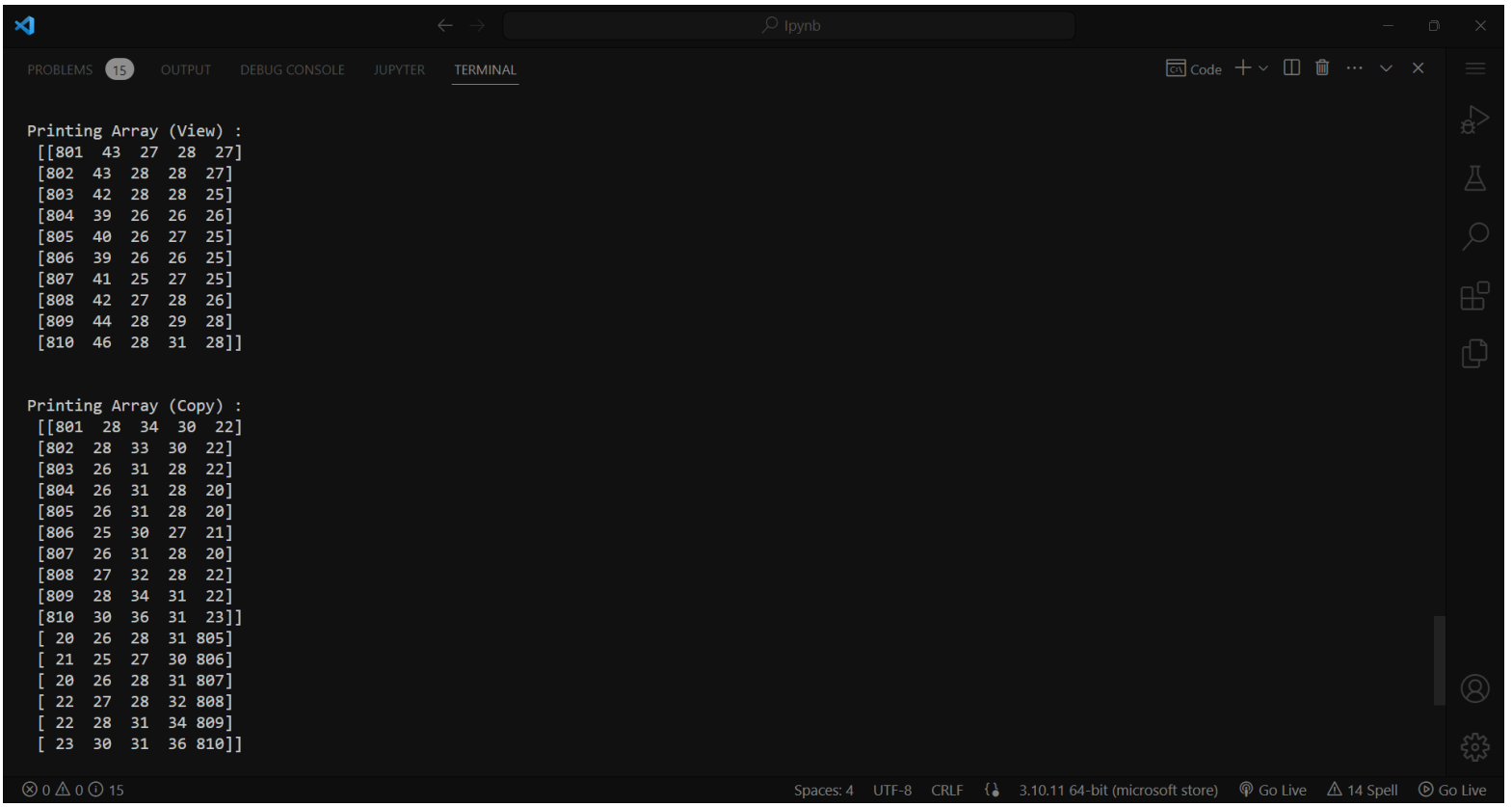

```
 [1.         0.64479351 1.16077537 1.05397187 0.83498612]
 [1.         0.62763916 1.22473664 1.00791652 0.80557738]
 [1.         0.65039109 1.19268381 1.02488446 0.84242655]
 [1.         0.63977654 1.21164021 1.04022796 0.80397022]
 [1.         0.64643237 1.26108033 1.00255591 0.80967403]]
```

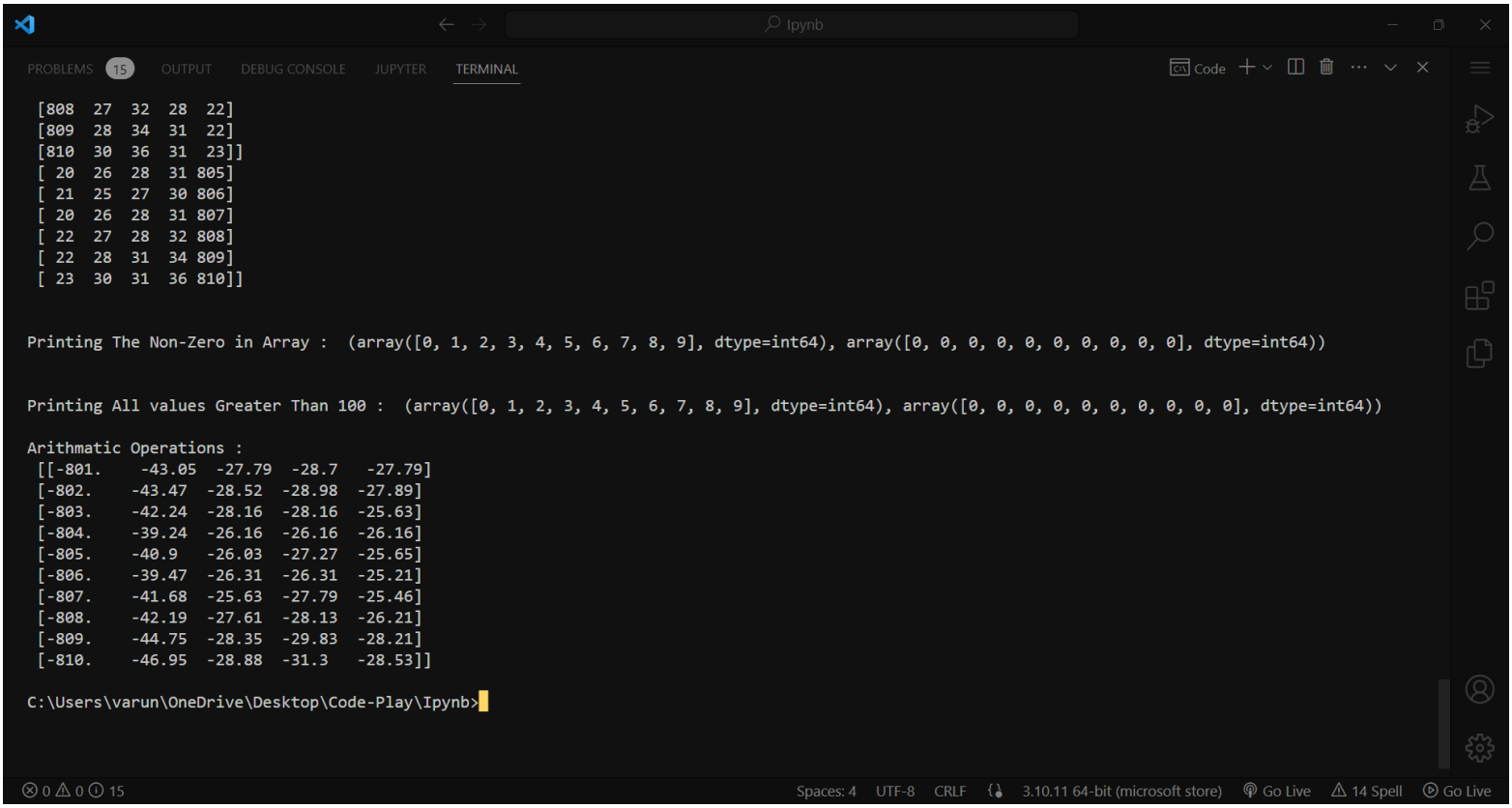
```
Vertical Stacking :

 [[801.    43.05  27.79  28.7   27.79]
 [802.    43.47  28.52  28.98  27.89]
 [803.    42.24  28.16  28.16  25.63]
 [804.    39.24  26.16  26.16  26.16]
 [805.    40.9   26.03  27.27  25.65]
 [806.    39.47  26.31  26.31  25.21]
 [807.    41.68  25.63  27.79  25.46]
 [808.    42.19  27.61  28.13  26.21]
 [809.    44.75  28.35  29.83  28.21]
 [810.    46.95  28.88  31.3   28.53]
 [801.    28.48  34.18  30.56  22.23]
 [802.    28.1   33.72  30.68  22.82]
 [803.    26.16  31.39  28.2   22.53]
 [804.    26.16  31.39  28.78  20.93]
 [805.    26.1   31.32  28.22  20.82]
 [806.    25.45  30.54  27.73  21.05]
 [807.    26.16  31.39  28.01  20.51]
 [808.    27.44  32.93  28.83  22.08]
 [809.    28.63  34.35  31.03  22.68]
 [810.    30.35  36.42  31.38  23.1 ]]


Horizontal Stacking :
```

```
Horizontal Stacking :

 [[801.    43.05  27.79  28.7   27.79 801.    28.48  34.18  30.56  22.23]
 [802.    43.47  28.52  28.98  27.89 802.    28.1   33.72  30.68  22.82]
 [803.    42.24  28.16  28.16  25.63 803.    26.16  31.39  28.2   22.53]
 [804.    39.24  26.16  26.16  26.16 804.    26.16  31.39  28.78  20.93]
 [805.    40.9   26.03  27.27  25.65 805.    26.1   31.32  28.22  20.82]
 [806.    39.47  26.31  26.31  25.21 806.    25.45  30.54  27.73  21.05]
 [807.    41.68  25.63  27.79  25.46 807.    26.16  31.39  28.01  20.51]
 [808.    42.19  27.61  28.13  26.21 808.    27.44  32.93  28.83  22.08]
 [809.    44.75  28.35  29.83  28.21 809.    28.63  34.35  31.03  22.68]
 [810.    46.95  28.88  31.3   28.53 810.    30.35  36.42  31.38  23.1 ]]

Max : [801. 802. 803. 804. 805. 806. 807. 808. 809. 810.]
Min : [801.    39.24  25.63  26.16  25.21]

Max In Array_2 :  810.0

Min In Array_2 :  20.51

[[0.98768834 0.68263633 0.46623224 0.4802235  0.46623224]
 [0.99026807 0.68797468 0.4774655  0.48450429 0.46777556]
 [0.99254615 0.6722376  0.47193538 0.47193538 0.43255789]
 [0.9945219  0.63257016 0.44087934 0.44087934 0.44087934]
 [0.9961947  0.65474081 0.43884169 0.45818421 0.43287258]
 [0.99756405 0.63567411 0.44322765 0.44322765 0.42593721]
 [0.99862953 0.66496969 0.43255789 0.46623224 0.42988087]
 [0.99939083 0.67159128 0.4634507  0.4714737  0.44166245]
 [0.9998477  0.70401472 0.47485639 0.49742825 0.47270457]
 [1.         0.73075827 0.48297676 0.51951911 0.47761884]]
```




Ipynb
PROBLEMS 15
OUTPUT
DEBUG CONSOLE
JUPYTER
TERMINAL
Code
Printing The Array_1 (Int) :
[[33 43 27 28 27]
[34 43 28 28 27]
[35 42 28 28 25]
[36 39 26 26 26]
[37 40 26 27 25]
[38 39 26 26 25]
[39 41 25 27 25]
[40 42 27 28 26]
[41 44 28 29 28]
[42 46 28 31 28]]
Printing The Array_2 (Int) :
[[33 28 34 30 22]
[34 28 33 30 22]
[35 26 31 28 22]
[36 26 31 28 20]
[37 26 31 28 20]
[38 25 30 27 21]
[39 26 31 28 20]
[40 27 32 28 22]
[41 28 34 31 22]
[42 30 36 31 23]]
[[33 8 2 28 18]
[34 8 0 28 18]
[35 10 28 28 16]
[36 2 26 24 16]
[37 8 26 24 16]
[38 1 26 26 17]
[39 8 25 24 16]
[40 10 0 28 18]
0 0 15
Spaces: 4
UTF-8
CRLF
3.10.11 64-bit (microsoft store)
Go Live
14 Spell
Go Live


Ipynb
PROBLEMS 15
OUTPUT
DEBUG CONSOLE
JUPYTER
TERMINAL
Code
Printing Array (View) :
[[801 43 27 28 27]
[802 43 28 28 27]
[803 42 28 28 25]
[804 39 26 26 26]
[805 40 26 27 25]
[806 39 26 26 25]
[807 41 25 27 25]
[808 42 27 28 26]
[809 44 28 29 28]
[810 46 28 31 28]]
Printing Array (Copy) :
[[801 28 34 30 22]
[802 28 33 30 22]
[803 26 31 28 22]
[804 26 31 28 20]
[805 26 31 28 20]
[806 25 30 27 21]
[807 26 31 28 20]
[808 27 32 28 22]
[809 28 34 31 22]
[810 30 36 31 23]]
[20 26 28 31 805]
[21 25 27 30 806]
[20 26 28 31 807]
[22 27 28 32 808]
[22 28 31 34 809]
[23 30 31 36 810]]
0 0 15
Spaces: 4
UTF-8
CRLF
3.10.11 64-bit (microsoft store)
Go Live
14 Spell
Go Live


Ipynb
PROBLEMS 15
OUTPUT
DEBUG CONSOLE
JUPYTER
TERMINAL
Code
[808 27 32 28 22]
[809 28 34 31 22]
[810 30 36 31 23]]
[20 26 28 31 805]
[21 25 27 30 806]
[20 26 28 31 807]
[22 27 28 32 808]
[22 28 31 34 809]
[23 30 31 36 810]]
Printing The Non-Zero in Array : (array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9], dtype=int64), array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0], dtype=int64))
Printing All values Greater Than 100 : (array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9], dtype=int64), array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0], dtype=int64))
Arithmatic Operations :
[[-801. -43.05 -27.79 -28.7 -27.79]
[-802. -43.47 -28.52 -28.98 -27.89]
[-803. -42.24 -28.16 -28.16 -25.63]
[-804. -39.24 -26.16 -26.16 -26.16]
[-805. -40.9 -26.03 -27.27 -25.65]
[-806. -39.47 -26.31 -26.31 -25.21]
[-807. -41.68 -25.63 -27.79 -25.46]
[-808. -42.19 -27.61 -28.13 -26.21]
[-809. -44.75 -28.35 -29.83 -28.21]
[-810. -46.95 -28.88 -31.3 -28.53]]
C:\Users\varun\OneDrive\Desktop\Code-Play\Ipynb>
0 0 15
Spaces: 4
UTF-8
CRLF
3.10.11 64-bit (microsoft store)
Go Live
14 Spell
Go Live