Name : Vaibhav .M. Nandgaye

Roll No. : 834

Div : H2

PNR: 202201090122

CODE

import numpy as np

a=np.loadtxt('testmarks1.csv',delimiter=',',skiprows=1,dtype=float)

print(a)

OUTPUT

```
[[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]]
```

CODE

import numpy as np

b=np.loadtxt('testmarks2.csv',delimiter=',',skiprows=1,dtype=float)

print(b)

OUTPUT

```
[[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 ]]
```

CODE

print("addition of a and b")

c=np.add(a,b)

print(c)

OUTPUT

```
addition of a and b
```

```
[[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]]
```

CODE

```
print("subtract of a and b")

c=np.subtract(a,b)

print(c)
```

OUTPUT

```
subtract of a and b
[[ 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.   16.12 -6.   -1.2   5.53]
 [ 0.   16.6  -7.54 -0.08  5.43]]
```

CODE

```
print("multiply of a and b")

c=np.multiply(a,b)

print(c)
```

OUTPUT

```
multiply of a and b
[[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
]]
```

CODE

```
print("Division of a and b")

a=np.divide(a,b)

print(a)
```

OUTPUT

```
Division of a and b
[[1.         1.51158708 0.81304857 0.93913613 1.25011246]
 [1.         1.54697509 0.84578885 0.94458931 1.22217353]
 [1.         1.6146789  0.89710099 0.99858156 1.13759432]
 [1.         1.5        0.83338643 0.90896456 1.24988055]
 [1.         1.56704981 0.83109834 0.96633593 1.23198847]
 [1.         1.55088409 0.86149312 0.94879192 1.1976247 ]
 [1.         1.59327217 0.81650207 0.99214566 1.24134569]
 [1.         1.53753644 0.83844519 0.97571974 1.1870471 ]
 [1.         1.56304576 0.82532751 0.96132775 1.24382716]
 [1.         1.54695222 0.7929709  0.99745061 1.23506494]]
```

CODE

```
print("Transpose of a is")

c=np.transpose(a)

print(c)
```

OUTPUT

```
Transpose of a is
[[1.         1.         1.         1.         1.         1.
  1.         1.         1.         1.        ]
 [1.51158708 1.54697509 1.6146789  1.5        1.56704981 1.55088409
  1.59327217 1.53753644 1.56304576 1.54695222]
 [0.81304857 0.84578885 0.89710099 0.83338643 0.83109834 0.86149312
  0.81650207 0.83844519 0.82532751 0.7929709 ]
 [0.93913613 0.94458931 0.99858156 0.90896456 0.96633593 0.94879192
  0.99214566 0.97571974 0.96132775 0.99745061]
 [1.25011246 1.22217353 1.13759432 1.24988055 1.23198847 1.1976247
  1.24134569 1.1870471  1.24382716 1.23506494]]
```

CODE

```
print("Transpose of A is")

c=np.transpose(a)

print(c)
```

OUTPUT

```
Transpose of A is
[[1.         1.         1.         1.         1.         1.
  1.         1.         1.         1.        ]
```

```
 [1.51158708 1.54697509 1.6146789  1.5        1.56704981 1.55088409
  1.59327217 1.53753644 1.56304576 1.54695222]
 [0.81304857 0.84578885 0.89710099 0.83338643 0.83109834 0.86149312
  0.81650207 0.83844519 0.82532751 0.7929709 ]
 [0.93913613 0.94458931 0.99858156 0.90896456 0.96633593 0.94879192
  0.99214566 0.97571974 0.96132775 0.99745061]
 [1.25011246 1.22217353 1.13759432 1.24988055 1.23198847 1.1976247
  1.24134569 1.1870471  1.24382716 1.23506494]]
```

CODE

```
print("Max in A")

c=np.max(a)

print(c)
```

OUTPUT

```
Max in A
1.614678899082569
```

CODE

```
print("min of B")

c=np.min(b)

print(c)
```

OUTPUT

```
min of B
20.51
```

CODE

```
print("Unique from A")

c=np.unique(a)

print(c)
```

OUTPUT

```
Unique from A
[0.7929709  0.81304857 0.81650207 0.82532751 0.83109834 0.83338643
 0.83844519 0.84578885 0.86149312 0.89710099 0.90896456 0.93913613
 0.94458931 0.94879192 0.96132775 0.96633593 0.97571974 0.99214566
 0.99745061 0.99858156 1.         1.13759432 1.1870471  1.1976247
 1.22217353 1.23198847 1.23506494 1.24134569 1.24382716 1.24988055
 1.25011246 1.5        1.51158708 1.53753644 1.54695222 1.54697509
 1.55088409 1.56304576 1.56704981 1.59327217 1.6146789 ]
```

CODE

```
print("Stadard deviation of B")

c=np.std(b,axis=0)

print(c)
```

OUTPUT

```
Stadard deviation of B
[2.87228132 1.47934479 1.77547768 1.33380508 0.9049116 ]
```

CODE

```
print("Mean of A")

c=np.mean(a,axis=0)

print(c)
```

OUTPUT

```
Mean of A
[1.         1.55319816 0.8355162  0.96330432 1.21966589]
```

CODE

```
print("Modulus of A and B")

c=np.mod(a,b)

print(c)
```

OUTPUT

```
Modulus of A and B
[[1.         1.51158708 0.81304857 0.93913613 1.25011246]
 [1.         1.54697509 0.84578885 0.94458931 1.22217353]
 [1.         1.6146789  0.89710099 0.99858156 1.13759432]
 [1.         1.5        0.83338643 0.90896456 1.24988055]
 [1.         1.56704981 0.83109834 0.96633593 1.23198847]
 [1.         1.55088409 0.86149312 0.94879192 1.1976247 ]
 [1.         1.59327217 0.81650207 0.99214566 1.24134569]
 [1.         1.53753644 0.83844519 0.97571974 1.1870471 ]
 [1.         1.56304576 0.82532751 0.96132775 1.24382716]
 [1.         1.54695222 0.7929709  0.99745061 1.23506494]]
```

CODE

```
print("Ravel of A is")

c=np.ravel(a)

print(C)
```

OUTPUT

```
Ravel of A is
[1.         1.51158708 0.81304857 0.93913613 1.25011246 1.
 1.54697509 0.84578885 0.94458931 1.22217353 1.         1.6146789
 0.89710099 0.99858156 1.13759432 1.         1.5        0.83338643
 0.90896456 1.24988055 1.         1.56704981 0.83109834 0.96633593
 1.23198847 1.         1.55088409 0.86149312 0.94879192 1.1976247
 1.         1.59327217 0.81650207 0.99214566 1.24134569 1.
 1.53753644 0.83844519 0.97571974 1.1870471  1.         1.56304576
 0.82532751 0.96132775 1.24382716 1.         1.54695222 0.7929709
 0.99745061 1.23506494]
```

CODE

print("Sorting of B is")

c=np.sort(b)

print(c)

OUTPUT

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

CODE

print("Transpose of A is")

c=np.transpose(a)

print(c)

OUTPUT

```
Transpose of A is
[[1.         1.         1.         1.         1.         1.
  1.         1.         1.         1.        ]
 [1.51158708 1.54697509 1.6146789  1.5        1.56704981 1.55088409
  1.59327217 1.53753644 1.56304576 1.54695222]
 [0.81304857 0.84578885 0.89710099 0.83338643 0.83109834 0.86149312
  0.81650207 0.83844519 0.82532751 0.7929709 ]
 [0.93913613 0.94458931 0.99858156 0.90896456 0.96633593 0.94879192
  0.99214566 0.97571974 0.96132775 0.99745061]
 [1.25011246 1.22217353 1.13759432 1.24988055 1.23198847 1.1976247
  1.24134569 1.1870471  1.24382716 1.23506494]]
```

THANK YOU

Name: Vaibhav .M. Nandgaye

Roll No.:834

Div:H2

PNR: 202201090122