

NAME: NIKITA SANJAY SHELAR

PRN:202201070047

ROLL NO: 557

DIV: E3

```
import csv
```

```
import numpy as np
```

```
array = np.loadtxt('/content/testmarks1.csv', delimiter=',', dtype=float, skiprows=1)
```

```
print(array)
```

```
[[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]]
```

```
# Transpose the matrix
```

```
transpose_array = np.transpose(array)
```

```
print(transpose_array)
```

```
[[801. 802. 803. 804. 805. 806. 807. 808. 809. 810. ]
```

```
[ 43.05 43.47 42.24 39.24 40.9 39.47 41.68 42.19 44.75 46.95]
```

```
[ 27.79 28.52 28.16 26.16 26.03 26.31 25.63 27.61 28.35 28.88]
[ 28.7 28.98 28.16 26.16 27.27 26.31 27.79 28.13 29.83 31.3 ]
[ 27.79 27.89 25.63 26.16 25.65 25.21 25.46 26.21 28.21 28.53]]
```

0s

#Random matrix of the same shape as the array

```
random_matrix = np.random.random(array[:, 1:].shape)
```

```
print("Random matrix:\n", random_matrix)
```

Random matrix:

```
[[0.56631328 0.21551628 0.68454243 0.22818798]
[0.78143198 0.93264469 0.86719966 0.9813187 ]
[0.69833567 0.51174533 0.27634833 0.15864249]
[0.62862734 0.14174014 0.23138864 0.89985346]
[0.98978917 0.19454841 0.30928277 0.82089845]
[0.24945028 0.70438855 0.97138954 0.32778541]
[0.95624127 0.174448 0.98845587 0.79013429]
[0.33147007 0.77828972 0.74201058 0.17613757]
[0.95529652 0.0041208 0.51281124 0.97117346]
[0.09680881 0.81611902 0.59478513 0.81318438]]
```

#Maximum marks in each subject

```
max_marks = np.max(array[:, 1:], axis=0)
```

```
print("Maximum marks in each subject:", max_marks)
```

Maximum marks in each subject: [46.95 28.88 31.3 28.53]

[8]

0s

```
#Minimum marks in each subject
```

```
min_marks = np.min(array[:, 1:], axis=0)
```

```
print("Minimum marks in each subject:", min_marks)
```

Minimum marks in each subject: [39.24 25.63 26.16 25.21]

```
#Maximum marks in EDS
```

```
max_marks_eds = np.max(array[:, 0], axis=0)
```

```
print("Maximum marks in EDS:", max_marks_eds)
```

Maximum marks in EDS: 810.0

```
#Maximum marks in SON
```

```
max_marks_son = np.max(array[:, 1], axis=0)
```

```
print("Maximum marks in SON:", max_marks_son)
```

Maximum marks in SON: 46.95

```
#sum of marks in each row
```

```
sum_marks_per_row = np.sum(array[:, 1:], axis=1)
```

```
print(sum_marks_per_row)
```

[127.33 128.86 124.19 117.72 119.85 117.3 120.56 124.14 131.14 135.66]

```
#element-wise rounding of marks to the nearest integer
```

```
rounded_marks = np.round(array[:, 1:])
```

```
print("Rounded marks:\n", rounded_marks)
```

Rounded marks:

[[43. 28. 29. 28.]

[43. 29. 29. 28.]

[42. 28. 28. 26.]

[39. 26. 26. 26.]

[41. 26. 27. 26.]

[39. 26. 26. 25.]

[42. 26. 28. 25.]

[42. 28. 28. 26.]

[45. 28. 30. 28.]

[47. 29. 31. 29.]]

#The exponential of each mark

exponential_marks = np.exp(array[:, 1:])

print("Exponential of each mark:\n", exponential_marks)

Exponential of each mark:

[[4.97024098e+18 1.17231319e+12 2.91240408e+12 1.17231319e+12]

[7.56451570e+18 2.43264437e+12 3.85348866e+12 1.29560645e+12]

[2.21105179e+18 1.69719839e+12 1.69719839e+12 1.35197161e+11]

[1.10081787e+17 2.29690824e+11 2.29690824e+11 2.29690824e+11]

[5.78954335e+17 2.01690463e+11 6.96964281e+11 1.37928325e+11]

[1.38548938e+17 2.66862665e+11 2.66862665e+11 8.88308645e+10]

[1.26297282e+18 1.35197161e+11 1.17231319e+12 1.14061088e+11]

[2.10321752e+18 9.79198288e+11 1.64703859e+12 2.41467325e+11]

[2.72068377e+19 2.05233647e+12 9.01580262e+12 1.78421561e+12]

[2.45542077e+20 3.48678073e+12 3.92118456e+13 2.45709285e+12]]

Cumulative sum of marks in each subject

```
cumulative_sum_subjects = np.cumsum(array[:, 1:], axis=0)

print("Cumulative sum of marks in each subject:\n", cumulative_sum_subjects)
```

Cumulative sum of marks in each subject:

```
[[ 43.05 27.79 28.7 27.79]
 [ 86.52 56.31 57.68 55.68]
 [128.76 84.47 85.84 81.31]
 [168. 110.63 112. 107.47]
 [208.9 136.66 139.27 133.12]
 [248.37 162.97 165.58 158.33]
 [290.05 188.6 193.37 183.79]
 [332.24 216.21 221.5 210. ]
 [376.99 244.56 251.33 238.21]
 [423.94 273.44 282.63 266.74]]
```

#square root of each mark

```
square_root_marks = np.sqrt(array[:, 1:])

print("Square root of each mark:\n", square_root_marks)
```

Square root of each mark:

```
[[6.56124988 5.27162214 5.35723809 5.27162214]
 [6.59317829 5.34041197 5.38330753 5.28109837]
 [6.49923072 5.30659966 5.30659966 5.06260802]
 [6.26418391 5.11468474 5.11468474 5.11468474]
 [6.39531078 5.10196041 5.22206856 5.0645829 ]
 [6.28251542 5.12932744 5.12932744 5.02095608]]
```

```
[6.45600496 5.06260802 5.27162214 5.04579032]
[6.49538298 5.25452186 5.30377224 5.11957029]
[6.68954408 5.3244718 5.46168472 5.31130869]
[6.85200701 5.37401154 5.59464029 5.34134814]]
```

```
#Maximum marks in DT
```

```
max_marks_dt = np.max(array[:, 2], axis=0)
```

```
print("Maximum marks in DT:", max_marks_dt)
```

```
Maximum marks in DT: 28.88
```

```
#Maximum marks in ET
```

```
max_marks_et = np.max(array[:, 3], axis=0)
```

```
print("Maximum marks in ET:", max_marks_et)
```

```
Maximum marks in ET: 31.3
```

```
#Minimum marks in DT
```

```
min_marks_dt = np.min(array[:, 2], axis=0)
```

```
print("Minimum marks in DT:", min_marks_dt)
```

```
Minimum marks in DT: 25.63
```

```
#Minimum marks in ET
```

```
min_marks_et = np.min(array[:, 3], axis=0)
```

```
print("Minimum marks in ET:", min_marks_et)
```

```
Minimum marks in ET: 26.16
```

```
#Mean of marks in each subject
```

```
mean_subjects = np.mean(array[:, 1:], axis=0)
```

```
print("Mean of marks in each subject:", mean_subjects)
```

Mean of marks in each subject: [42.394 27.344 28.263 26.674]

Variance of marks in each subject

```
variance_subjects = np.var(array[:, 1:], axis=0)
```

```
print("Variance of marks in each subject:", variance_subjects)
```

Variance of marks in each subject: [4.920064 1.282524 2.185881 1.476324].