Ayush Tilekar

Roll no - 565

PRN - 202201070050

```
import csv
import numpy as np
data = np.loadtxt('testmarks1.csv', delimiter=',',dtype=float,skiprows=1)
print(data)
```

```
[[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]]
```

```
# Mean of marks in each subject
mean_subjects = np.mean(data[:, 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(data[:, 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]
```

```
# Cumulative sum of marks in each subject
cumulative_sum_subjects = np.cumsum(data[:, 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]]
```

```
# Transpose the matrix
transpose_data = np.transpose(data)
```

```
print(transpose_data)
```

```
[[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]]
```

```
#The exponential of each mark
exponential_marks = np.exp(data[:, 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]]
```

```
#Random matrix of the same shape as the data
random_matrix = np.random.random(data[:, 1:].shape)
print("Random matrix:\n", random_matrix)
```

```
Random matrix:
 [[0.3578862  0.48689323 0.02525061 0.82740057]
 [0.86606786 0.86693284 0.65685586 0.67849431]
 [0.2737587  0.13982185 0.1914023  0.14153692]
 [0.5190879  0.59732843 0.78089995 0.9126588 ]
 [0.95532293 0.56448727 0.2268515  0.74482143]
 [0.18211861 0.9076692  0.26761959 0.00379784]
 [0.53936819 0.57966006 0.37884128 0.52363593]
 [0.47448869 0.22023169 0.28878634 0.27182159]
 [0.81695964 0.67489043 0.59538095 0.43510818]
 [0.07019108 0.26700015 0.28298573 0.99322802]]
```

```
#element-wise rounding of marks to the nearest integer
rounded_marks = np.round(data[:, 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.]]
```

```
#Maximum marks in each subject
max_marks = np.max(data[:, 1:], axis=0)
print("Maximum marks in each subject:", max_marks)
```

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

```
#Minimum marks in each subject
min_marks = np.min(data[:, 1:], axis=0)
print("Minimum marks in each subject:", min_marks)
```

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

```
#sum of marks in each row
sum_marks_per_row = np.sum(data[:, 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]
```

```
#square root of each mark
square_root_marks = np.sqrt(data[:, 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]]
```

✓ 0s completed at 12:00 PM