

▼ Name - Omkar Rajendra Shinde

Roll no 582

Division E

Batch E4

PRN - 202201040190

EDS ASSIGNMENT 3 : NumPy Operations

```
import csv
import numpy as np
#importing testmarks1.csv file to a numpy array and printing the array
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]]
```

```
# Operation 1 : Seperate the data into subjects and marks
Roll=data[:,0]
EDS=data[:,1]
SON=data[:,2]
DT=data[:,3]
ET=data[:,4]
print("Roll=",Roll)
print("EDS=",EDS)
print("SON=",SON)
print("DT=",DT)
print("ET=",ET)
```

```
Roll= [801. 802. 803. 804. 805. 806. 807. 808. 809. 810.]
EDS= [43.05 43.47 42.24 39.24 40.9 39.47 41.68 42.19 44.75 46.95]
SON= [27.79 28.52 28.16 26.16 26.03 26.31 25.63 27.61 28.35 28.88]
DT= [28.7 28.98 28.16 26.16 27.27 26.31 27.79 28.13 29.83 31.3 ]
ET= [27.79 27.89 25.63 26.16 25.65 25.21 25.46 26.21 28.21 28.53]
```

```
# Operation 2 : Maximum Marks in EDS Subject
print("Roll Number =",Roll[np.argmax(EDS)],"Got Maximum Marks for EDS Subject=",max(EDS))
```

```
Roll Number = 810.0 Got Maximum Marks for EDS Subject= 46.95
```

```
# Operation 3 : Minimum Marks in EDS Subject
print("Roll Number =",Roll[np.argmin(EDS)],"Got Minimum Marks for EDS Subject=",min(EDS))
```

```
Roll Number = 804.0 Got Minimum Marks for EDS Subject= 39.24
```

```
# Operation 4 : 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]]
```

```
# Operation 5 : Calculate the sum of all marks
sum_marks = np.sum(data[:, 1:])
print("Sum of all marks:", sum_marks)
```

```
Sum of all marks: 1246.75
```

```
# Operation 6 :Custom sequence generation
# Generate a sequence of integers from 0 to 9
sequence = np.arange(10)
print(sequence)
```

```
[0 1 2 3 4 5 6 7 8 9]
```

```
# Operation 7: Calculate the mean marks
mean_marks = np.mean(data[:, 1:])
```

```
print("Mean marks:", mean_marks)
```

Mean marks: 31.16875

```
# Operation 8: Calculate the 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]

```
# Operation 9: Calculate the 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]

```
# Operation 10: Calculate the standard deviation of marks in each subject
std_dev_marks = np.std(data[:, 1:], axis=0)
print("Standard deviation of marks in each subject:", std_dev_marks)
```

Standard deviation of marks in each subject: [2.21812173 1.13248576 1.47847252 1.21504074]

```
# Operation 11: Perform element-wise multiplication of marks by 2
multiplication_by_2 = data[:, 1:] * 2
print("Element-wise multiplication of marks by 2:\n", multiplication_by_2)
```

Element-wise multiplication of marks by 2:

```
[[86.1 55.58 57.4 55.58]
 [86.94 57.04 57.96 55.78]
 [84.48 56.32 56.32 51.26]
 [78.48 52.32 52.32 52.32]
 [81.8 52.06 54.54 51.3 ]
 [78.94 52.62 52.62 50.42]
 [83.36 51.26 55.58 50.92]
 [84.38 55.22 56.26 52.42]
 [89.5 56.7 59.66 56.42]
 [93.9 57.76 62.6 57.06]]
```

```
# Operation 12: Perform element-wise division of marks by 2
division_by_2 = data[:, 1:] / 2
print("Element-wise division of marks by 2:\n", division_by_2)
```

Element-wise division of marks by 2:

```
[[21.525 13.895 14.35  13.895]
 [21.735 14.26  14.49  13.945]
 [21.12  14.08  14.08  12.815]
 [19.62  13.08  13.08  13.08 ]
 [20.45  13.015 13.635 12.825]
 [19.735 13.155 13.155 12.605]
 [20.84  12.815 13.895 12.73 ]
 [21.095 13.805 14.065 13.105]
 [22.375 14.175 14.915 14.105]
 [23.475 14.44  15.65  14.265]]
```

Operation 13: Copy the marks array and assign it to a new variable

```
copy_marks = data[:, 1:].copy()
print("Copy of marks array:\n", copy_marks)
```

Copy of marks array:

```
[[43.05 27.79 28.7  27.79]
 [43.47 28.52 28.98 27.89]
 [42.24 28.16 28.16 25.63]
 [39.24 26.16 26.16 26.16]
 [40.9  26.03 27.27 25.65]
 [39.47 26.31 26.31 25.21]
 [41.68 25.63 27.79 25.46]
 [42.19 27.61 28.13 26.21]
 [44.75 28.35 29.83 28.21]
 [46.95 28.88 31.3  28.53]]
```

Operation 14: Calculate the sum of marks in each row

```
sum_marks_per_row = np.sum(data[:, 1:], axis=1)
print("Sum of marks in each row:", sum_marks_per_row)
```

Sum of marks in each row: [127.33 128.86 124.19 117.72 119.85 117.3 120.56 124.14 131.14 135.66]

End of Notebook

✓ 0s completed at 11:31 AM

