COURSE4_SPRINT2_PRACTICE

Task - 1

The university wishes to know the total number of students whose GRE score is greater than 320. Help the university perform this task.

```
import pandas as pd

import numpy as np

from numpy import genfromtxt

std = genfromtxt('s2_practice.csv',delimiter=',',skip_header=1)
std
```

```
array([[  1.  , 337.  , 118.  , ...,   9.65,   1.  ,   0.92],
       [  2.  , 324.  , 107.  , ...,   8.87,   1.  ,   0.76],
       [  3.  , 316.  , 104.  , ...,   8.  ,   1.  ,   0.72],
       ...,
       [398.  , 330.  , 116.  , ...,   9.45,   1.  ,   0.91],
       [399.  , 312.  , 103.  , ...,   8.78,   0.  ,   0.67],
       [400.  , 333.  , 117.  , ...,   9.66,   1.  ,   0.95]])
```

```
gr=std[0:,1]

index1=np.where(gr > 320)
std[index1].size
```

```
1467
```

Task - 2

The university wishes to know the admission possibility of those students with TOEFL scores greater than 110. Help the university accomplish this task.

```
tof = std[0:,2]
t1=std[0:,8]

index2 = np.where(tof>110)
index2
```

```
(array([  0,   5,  11,  12,  21,  22,  23,  24,  25,  32,  33,  34,
43,
         44,  46,  47,  49,  52,  53,  64,  65,  66,  69,  70,  71,
```

```
72,
        75,  76,  81,  83,  84,  97,  98,  99, 104, 106, 107, 108,
120,
       121, 126, 127, 128, 129, 130, 133, 134, 138, 141, 142, 143,
144,
       145, 147, 148, 150, 151, 152, 164, 171, 173, 174, 175, 176,
185,
       187, 188, 189, 190, 192, 193, 202, 203, 212, 213, 214, 215,
216,
       222, 228, 229, 234, 235, 236, 237, 242, 243, 249, 253, 254,
259,
       263, 268, 276, 283, 284, 285, 286, 287, 297, 298, 299, 307,
318,
       319, 325, 328, 330, 335, 337, 361, 362, 365, 372, 384, 385,
392,
       394, 397, 399], dtype=int64),)
```

```
t1[index2]
```

```
array([0.92, 0.9 , 0.84, 0.78, 0.7 , 0.94, 0.95, 0.97, 0.94, 0.91, 0.9
,
       0.94, 0.87, 0.91, 0.86, 0.89, 0.78, 0.78, 0.72, 0.52, 0.55,
0.61,
       0.78, 0.94, 0.96, 0.93, 0.72, 0.74, 0.96, 0.92, 0.94, 0.86, 0.9
,
       0.79, 0.74, 0.87, 0.91, 0.93, 0.94, 0.94, 0.85, 0.78, 0.84,
0.92,
       0.96, 0.79, 0.89, 0.8 , 0.9 , 0.92, 0.97, 0.8 , 0.81, 0.83,
0.96,
       0.93, 0.94, 0.86, 0.81, 0.89, 0.89, 0.87, 0.85, 0.9 , 0.89,
0.93,
       0.93, 0.88, 0.9 , 0.86, 0.94, 0.97, 0.97, 0.95, 0.96, 0.94,
0.93,
       0.91, 0.76, 0.71, 0.82, 0.91, 0.88, 0.85, 0.86, 0.7 , 0.76,
0.77,
       0.93, 0.85, 0.9 , 0.7 , 0.83, 0.89, 0.8 , 0.94, 0.93, 0.92,
0.89,
       0.86, 0.9 , 0.71, 0.8 , 0.79, 0.8 , 0.81, 0.8 , 0.8 , 0.83,
0.94,
       0.93, 0.91, 0.86, 0.95, 0.96, 0.96, 0.84, 0.89, 0.91, 0.95])
```

Task - 3

The university decides to offer 100% scholarship to the top 11 highest GRE scorers and the top 8 highest TOEFL scorers. Help the university find the list of these students along with their serial numbers.

```
j=std[0:,0:2]
k = np.sort(j)
k[389:400]
```

```
array([[320., 390.],
       [314., 391.],
       [318., 392.],
       [326., 393.],
       [317., 394.],
       [329., 395.],
       [324., 396.],
       [325., 397.],
       [330., 398.],
       [312., 399.],
       [333., 400.]])
```

```
s2 = std[0:,0:3]
s4 = np.delete(s2,1,axis=1)
kl = np.sort(s4[0:,1])
jj = kl[392:401]
indx = np.where(jj)
```

```
(array([0, 1, 2, 3, 4, 5, 6, 7], dtype=int64),)
```

Task - 4

To offer fellowship, the university wishes to know the serial numbers and TOEFL scores of the students with: ● GRE scores between 300 and 305, ● SOP scores of 3.5, and ● Research experience.

```
gre_ind = np.where((std[0:,1]>300) & (std[0:,1]<305))
gre_ind
```

```
(array([  8,  19,  38,  62,  77,  93,  94,  95, 109, 112, 124, 131,
166,
        209, 233, 279, 292, 313, 346, 348, 352, 357, 369, 375, 378,
386],
       dtype=int64),)
```

```
std[gre_ind,0:3]
```

```
array([[[  9., 302., 102.],
        [ 20., 303., 102.],
        [ 39., 304., 105.],
        [ 63., 304., 105.],
        [ 78., 301.,  99.],
        [ 94., 301.,  97.],
```

```
        [ 95., 303.,  99.],
        [ 96., 304., 100.],
        [110., 304., 103.],
        [113., 301., 107.],
        [125., 301., 106.],
        [132., 303., 105.],
        [167., 302., 102.],
        [210., 301., 104.],
        [234., 304., 100.],
        [280., 304., 102.],
        [293., 302.,  99.],
        [314., 301., 100.],
        [347., 304.,  97.],
        [349., 302.,  99.],
        [353., 303., 100.],
        [358., 301., 104.],
        [370., 301.,  98.],
        [376., 304., 101.],
        [379., 303.,  98.],
        [387., 302., 101.]]])
```

```
sop_ind = np.where(std[0:,4]==3.5)
sop_ind
```

```
(array([  3,   9,  10,  14,  15,  19,  40,  48,  65,  67,  68,  87,
100,
        104, 107, 112, 114, 123, 128, 132, 135, 139, 149, 163, 166,
177,
        180, 186, 194, 197, 201, 204, 206, 207, 209, 219, 221, 226,
227,
        231, 242, 243, 248, 249, 261, 275, 293, 296, 303, 305, 308,
309,
        313, 319, 325, 330, 332, 339, 341, 351, 353, 356, 357, 364,
366,
        371, 380, 389, 395, 398], dtype=int64),)
```

```
std[sop_ind,0:3]
```

```
array([[[  4., 322., 110.],
        [ 10., 323., 108.],
        [ 11., 325., 106.],
        [ 15., 311., 104.],
        [ 16., 314., 105.],
        [ 20., 303., 102.],
        [ 41., 308., 110.],
        [ 49., 321., 110.],
        [ 66., 325., 112.],
        [ 68., 316., 107.],
        [ 69., 318., 109.],
        [ 88., 317., 107.],
```

```
       [101., 322., 107.],
       [105., 326., 112.],
       [108., 338., 117.],
       [113., 301., 107.],
       [115., 311., 105.],
       [124., 308., 108.],
       [129., 326., 112.],
       [133., 309., 105.],
       [136., 314., 109.],
       [140., 318., 109.],
       [150., 311., 106.],
       [164., 317., 105.],
       [167., 302., 102.],
       [178., 319., 110.],
       [181., 300., 104.],
       [187., 317., 107.],
       [195., 316., 109.],
       [198., 310., 106.],
       [202., 315., 110.],
       [205., 298., 105.],
       [207., 315.,  99.],
       [208., 310., 102.],
       [210., 301., 104.],
       [220., 312., 104.],
       [222., 316., 110.],
       [227., 306., 110.],
       [228., 312., 110.],
       [232., 319., 106.],
       [243., 324., 115.],
       [244., 325., 114.],
       [249., 324., 110.],
       [250., 321., 111.],
       [262., 312., 104.],
       [276., 322., 110.],
       [294., 312.,  98.],
       [297., 310., 107.],
       [304., 323., 107.],
       [306., 321., 109.],
       [309., 312., 108.],
       [310., 308., 110.],
       [314., 301., 100.],
       [320., 327., 113.],
       [326., 326., 116.],
       [331., 327., 113.],
       [333., 308., 106.],
       [340., 324., 107.],
       [342., 326., 110.],
       [352., 325., 110.],
       [354., 300., 102.],
```

```
        [357., 327., 109.],
        [358., 301., 104.],
        [365., 313., 102.],
        [367., 320., 104.],
        [372., 324., 110.],
        [381., 322., 104.],
        [390., 320., 108.],
        [396., 324., 110.],
        [399., 312., 103.]])
```

Task - 5

The university decides to offer the scholarship amounts between $5,000 and $10,000 to all the students randomly. The university also wishes to add this data to the existing record of students. Help the company accomplish this task. Further, determine the number of students who receive a scholarship amount greater than $9,000

```
import numpy as np

import pandas as pd

std.shape
sch1 = np.random.randint(5000,10000,size=(400,1))
std1 = np.concatenate((std,sch1),axis=1)

indsch = np.where(std1[0:,9]>9000)
std1[indsch].shape
```

```
(91, 10)
```

Task - 6

The university wishes to know the SOP scores and the LOR scores of all students who have received more than $9,000 in scholarships. Help the university accomplish this task

```
std1[indsch,4:6]
```

```
array([[[4. , 4.5],
        [3. , 3.5],
        [2. , 3. ],
        [3. , 4. ],
        [4. , 4.5],
        [5. , 4.5],
        [4. , 4. ],
        [5. , 5. ],
        [2.5, 2. ],
```

```
       [3. , 4. ],
       [2.5, 4.5],
       [3. , 3.5],
       [3. , 3. ],
       [2. , 3. ],
       [3. , 3.5],
       [3.5, 4. ],
       [4.5, 4. ],
       [1.5, 2. ],
       [5. , 5. ],
       [4. , 4.5],
       [4.5, 4.5],
       [4.5, 3.5],
       [1.5, 2.5],
       [3. , 3. ],
       [5. , 4.5],
       [4. , 3.5],
       [4.5, 4. ],
       [3.5, 3. ],
       [4. , 3.5],
       [3. , 3.5],
       [4.5, 4.5],
       [3.5, 3.5],
       [2. , 3. ],
       [3.5, 3.5],
       [4. , 4. ],
       [4. , 3. ],
       [4. , 3.5],
       [2. , 2.5],
       [3. , 4. ],
       [1.5, 2. ],
       [4.5, 4. ],
       [2. , 4. ],
       [4. , 4.5],
       [4. , 5. ],
       [2.5, 2.5],
       [4. , 4.5],
       [3.5, 3. ],
       [3.5, 4. ],
       [5. , 4. ],
       [4.5, 4. ],
       [2.5, 2.5],
       [4. , 4.5],
       [3.5, 2.5],
       [2.5, 3.5],
       [5. , 4. ],
       [4.5, 4. ],
       [2. , 3.5],
       [2.5, 3. ],
```

```
        [4. , 4.5],
        [5. , 5. ],
        [2.5, 2.5],
        [4.5, 5. ],
        [1.5, 2. ],
        [1. , 1.5],
        [5. , 4.5],
        [2.5, 3. ],
        [4. , 3.5],
        [2.5, 3. ],
        [5. , 4.5],
        [3. , 3.5],
        [4.5, 4. ],
        [3. , 4. ],
        [3. , 2.5],
        [4. , 4. ],
        [1.5, 2. ],
        [1. , 1. ],
        [2. , 2. ],
        [3.5, 4. ],
        [3.5, 3.5],
        [2.5, 2. ],
        [2. , 1.5],
        [4.5, 3. ],
        [2. , 2. ],
        [3.5, 3. ],
        [2. , 2.5],
        [2.5, 3. ],
        [5. , 5. ],
        [2. , 3.5],
        [3.5, 3.5],
        [3. , 3.5],
        [5. , 4.5]]])
```

Task - 7

The university decides to offer $1,000 scholarships to all students with GRE scores above 315, TOEFL scores above 115, and research experience. Display the serial number of all the students who are eligible for this. How many such students are there? Further, update the scholarship amount for these students

```
offerind = np.where((std1[0:,1]>315)&(std1[0:,2]>115))
```

```
std1[offerind].shape
```

```
(44, 10)
```

```
std1[offerind,9]
```

```
array([[5178., 7955., 9279., 6447., 6838., 7218., 6029., 8471., 6636.,
        5363., 9607., 8073., 9764., 7989., 5043., 6603., 9432., 5730.,
        8696., 7776., 8256., 5455., 5010., 9063., 6908., 7081., 7885.,
        8687., 7445., 8904., 5131., 6473., 6683., 8100., 7757., 5179.,
        8534., 6057., 6214., 7215., 8696., 9241., 9389., 6042.]])

std1[offerind,9]=std1[offerind,9]+1000

std1[offerind,9]

array([[ 6178.,  8955., 10279.,  7447.,  7838.,  8218.,  7029.,
9471.,
         7636.,  6363., 10607.,  9073., 10764.,  8989.,  6043.,
7603.,
        10432.,  6730.,  9696.,  8776.,  9256.,  6455.,  6010.,
10063.,
         7908.,  8081.,  8885.,  9687.,  8445.,  9904.,  6131.,
7473.,
         7683.,  9100.,  8757.,  6179.,  9534.,  7057.,  7214.,
8215.,
         9696., 10241., 10389.,  7042.]])
```