Mastery Exams

Answer the question or complete the task outlined below, use the specific method described if applicable

Note: Use [Alt + Enter] to execute the code.

1. Import Numpy and Pandas

```
In [1]:   import numpy as np
```

```
In [2]:   import pandas as pd
```

2. What is

a. The modulo of 25 divided by 7

b. THe 11th root of 528

c. The natural logarithm of 58

d. e^{12}

```
In [1]:   # a. modulo
```

```
In [ ]:
```

```
In [ ]:
```

```
In [2]:   # b. 11th root
```

```
In [ ]:
```

```
In [ ]:
```

```
In [3]:   # c. Natural logarithm
```

```
In [ ]:
```

```
In [ ]:
```

In [4]: `# d. Euler's Number`

In [ ]:

3. Split the the string into a list

str = "Machine Learning next semester is a fun subject"

In [ ]:

4. Extract the following strings from number 3

In [5]: `# Machine`

In [ ]:

In [6]: `# Learning`

In [ ]:

In [7]: `# next`

In [ ]:

In [8]: `# semester`

In [ ]:

In [9]: `# subject`

In [ ]:

5. Given the variables

name = "James"

work = "Programmer"

Use .format to print the following string:

James is a Programmer

In [ ]:

In [ ]:

6. Given the list below, use indexing to grab instances of 'target'

In [14]: `list_1 = [1,2,'target 1',[4,[5,6,7,[8,9,'target 2',[11,12,13,14,[15,'target 3']]],16,17]], 'target 4']`

In [11]: `# target 1`

In [ ]:

In [12]: `# target 2`

In [ ]:

In [13]: `# target 3`

In [ ]:

In [ ]: `# target 4`

In [ ]:

7. Given the dictionary below, use indexing to grab instances of 'tricky'

In [15]: `d = {'k1':[1,'tricky 1',3,{'k2':['tricky 2', 4, 5,{'k3':[1,2,3,'tricky 3']}]}]}`

In [16]: `# tricky 1`

In [ ]:

In [17]: `# tricky 2`

In [ ]:

In [18]: `# tricky 3`

8. Create a function that grabs the mobile phones operator identity.

0917-3453-234

e.g. So for example, passing '0917-3453-234' would return '0917'

```
In [19]:  # 0917
```

```
In [ ]:
```

9. Create a function that counts the number of times a "cat" occurs in a string. Ignore edge cases.

```
In [21]:  str = "Felix is a cat. The cat is owned by Felice bought at the Cat Store"
```

```
In [ ]:
```

10. Use lambda expressions and the filter() function to filter out words from a list that start with the letter 's' For example.

seq = ['silver', 'boat', 'shout', 'selenium', 'service', 'board']

```
In [23]:  # ['silver', 'shout', 'selenium', 'service']
```

```
In [ ]:
```

11. Create the arrays using Numpy .arange()

```
In [28]:  # .arange
Out[28]:  array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16,
                 17, 18, 19, 20])
```

```
In [31]:  # .arange
Out[31]:  array([5, 7, 9])
```

```
In [32]:  # .arange
Out[32]:  array([10, 13, 16, 19, 22, 25, 28, 31, 34, 37, 40, 43, 46, 49, 52, 55, 58,
                 61, 64, 67, 70, 73, 76, 79, 82, 85, 88, 91, 94, 97])
```

12. Create the array using np.ones

```
In [34]:
```

```
Out[34]: array([[5., 5., 5., 5., 5.],
                [5., 5., 5., 5., 5.],
                [5., 5., 5., 5., 5.],
                [5., 5., 5., 5., 5.],
                [5., 5., 5., 5., 5.]])
```

13. Using .randn, create a 5 x 5 normally distributed array

```
In [38]: np.random.seed(10) # execute this so that we can have the same values
```

```
In [39]:
```

```
Out[39]: array([[ 1.3315865 ,  0.71527897, -1.54540029, -0.00838385,  0.62133597],
                [-0.72008556,  0.26551159,  0.10854853,  0.00429143, -0.17460021],
                [ 0.43302619,  1.20303737, -0.96506567,  1.02827408,  0.22863013],
                [ 0.44513761, -1.13660221,  0.13513688,  1.484537  , -1.07980489],
                [-1.97772828, -1.7433723 ,  0.26607016,  2.38496733,  1.12369125]])
```

14. Given the array below, identify the maximum and minimum values

```
In [41]: np.random.seed(10)
         random_array = np.random.randint(0,25,10)
         random_array
```

```
Out[41]: array([ 9,  4, 15,  0, 17, 16, 17,  8,  9,  0])
```

```
In [ ]: # maximum value?
```

```
In [42]: # minimum value?
```

15. Given the array below, slice and select the following

```
In [43]: new_arr = np.arange(0,16)
         new_arr
```

```
Out[43]: array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15])
```

```
In [44]:
```

```
Out[44]: array([ 5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15])
```

```
In [60]:
```
Out[60]: array([2, 3, 4])

```
In [74]:
```
Out[74]: array([11, 12])

```
In [ ]:
```

16. Using NumPy create the following Matrices

$$a = \begin{bmatrix} 1 & -3 \\ 5 & 6 \end{bmatrix}$$

$$b = \begin{bmatrix} 6 & 7 & -2 \\ 4 & 5 & 8 \end{bmatrix}$$

```
In [75]: # matrix a
```

```
In [ ]:
```

```
In [76]: # matrix b
```

```
In [ ]:
```

17. Create the following NumPy array and then select the following submatrices:

$$c = \begin{bmatrix} 2 & 4 & 6 & 8 & 10 & 12 & 14 & 16 \\ 18 & 20 & 22 & 24 & 26 & 28 & 30 & 32 \\ 34 & 36 & 38 & 40 & 42 & 44 & 46 & 48 \\ 50 & 52 & 54 & 56 & 58 & 60 & 62 & 64 \end{bmatrix}$$

```
In [ ]:
```

$$subd1 = \begin{bmatrix} 24 & 26 & 28 & 30 & 32 \\ 40 & 42 & 44 & 46 & 48 \end{bmatrix}$$

```
In [ ]:
```

$$subd2 = \begin{bmatrix} 22 & 24 & 26 & 30 \\ 38 & 40 & 42 & 46 \end{bmatrix}$$

In [ ]:

$$subd3 = \begin{bmatrix} 26 \\ 42 \\ 58 \end{bmatrix}$$

In [ ]:

$$subd5 = \begin{bmatrix} 18 & 34 \\ 50 & 14 \\ 30 & 46 \end{bmatrix}$$ hint: use numpy.concatenate and numpy.transpose

In [ ]:

18. Matrix Operations

$$e = \begin{bmatrix} 3 & -3 \\ 1 & -2 \\ 6 & 4 \end{bmatrix}$$

$$f = \begin{bmatrix} 1 & 5 \\ -3 & 5 \\ 2 & 5 \end{bmatrix}$$

In [ ]:

In [ ]:

In [78]: # add matrix e and matrix f

In [ ]:

In [79]: # multipy matrix e by 5 (scalar multiplication)

In [ ]:

In [80]: # multipy matrix e by matrix f

In [ ]:

In [81]: # transpose matrix f

In [ ]:

19. Find the determinant of the Matrix

$$h = \begin{bmatrix} 2 & -3 & 4 \\ 2 & -3 & 2 \\ -1 & -4 & 3 \end{bmatrix}$$

In [ ]:

In [ ]:

20. Find the Inverse of the Matrix and prove that it is the Inverse of the Original

$$i = \begin{bmatrix} 4 & 1 & 5 \\ 0 & 3 & -3 \\ -2 & 2 & -2 \end{bmatrix}$$

In [ ]:

In [ ]:

21. Solve for the value of x: $e^{4x} = 132$

In [ ]:

In [ ]:

22. Solve the following problem using Systems of Linear Equations

1. The ticket for a ride is 1.50 (USD) for children and 4.00 (USD) for adults. During some instances, 2200 customers have enter the carnival and 5050 (USD) is collected. How many adults and children went to pay for the carnival?

In [ ]:

In [ ]:

1. At a store, May pays 34 (USD) for 2 pounds of apples, 1 pound of berries, and 4 pounds of cherries. Tom pays 35 (USD) for 3 pounds of apples, 2 pounds of berries, and 2 pounds of cherries. Lee pays 49 (USD) for 5 pounds of apples, 3 pounds of berries, and 2 pounds of cherries. What is the price per pound for apples, for berries, and for cheeries?

In [ ]:

In [ ]:

23. Find the Eigen Values and Eigen Vectors of the Matrix Below

$$eg = \begin{bmatrix} 3 & -2 \\ -5 & 3 \end{bmatrix}$$

In [ ]:

24. Create a Series object out of the following:

a. numpy array with [2,4,6,8,12]

b. dictionary {'Ilocos': 38, 'Bicol': 26, 'Quezon': 19, 'Batangas': 55}

In [89]: `# from numpy array`

In [90]:

```
Out[90]: 0     2
         1     4
         2     6
         3     8
         4    12
         dtype: int32
```

In [87]: `# from dictionary`

In [88]:

```
Out[88]: Ilocos      38
         Bicol       26
         Quezon      19
         Batangas    55
         dtype: int64
```

In [ ]:

25. Create the data frame as seen below

In [93]:

In [94]:

Out[94]:

	area	population	code
California	423967	38332521	CA
Texas	695662	26448193	TX
New York	141297	19651127	NY
Florida	170312	19552860	FL
Illinois	149995	12882135	IL

26. Create another column 'density' based on number 25

In [ ]:

In [96]:

Out[96]:

	area	population	code	density
California	423967	38332521	CA	90.413926
Texas	695662	26448193	TX	38.018740
New York	141297	19651127	NY	139.076746
Florida	170312	19552860	FL	114.806121
Illinois	149995	12882135	IL	85.883763

In [ ]:

In [ ]:

26. Permanently drop the 'Texas' row based on number 25

In [ ]:

In [98]:

Out[98]:

	area	population	code	density
California	423967	38332521	CA	90.413926
New York	141297	19651127	NY	139.076746
Florida	170312	19552860	FL	114.806121
Illinois	149995	12882135	IL	85.883763

In [ ]:

In [ ]:

27. Create another Data Frame

In [104]: ```np.random.seed(10) # execute this to have the same random values```

In [106]:

Out[106]:

	C1	C2	C3	C4	C5	C6	C7	C8
R1	8	3	7	15	16	10	2	2
R2	2	14	2	17	16	15	4	11
R3	16	9	2	12	4	1	13	19
R4	4	4	3	7	17	15	1	14
R5	7	16	2	9	19	2	14	17
R6	16	15	7	13	6	12	18	0
R7	2	10	17	8	13	10	17	4
R8	18	8	19	14	0	13	12	10

28. Select the following parts of the Data Frame from number 27 (use .loc or iloc)

In [108]:

Out[108]:

	C2	C3	C4	C5	C6
R1	3	7	15	16	10
R2	14	2	17	16	15
R3	9	2	12	4	1
R4	4	3	7	17	15
R5	16	2	9	19	2

In [112]:

Out[112]:

	C3	C4	C5	C6	C7
R3	2	12	4	1	13
R4	3	7	17	15	1
R5	2	9	19	2	14
R6	7	13	6	12	18
R7	17	8	13	10	17
R8	19	14	0	13	12

In [117]:

Out[117]:
```
R1      8
R2      2
R3     16
R4      4
R5      7
R6     16
R7      2
R8     18
Name: C1, dtype: int32
```

In [118]:

Out[118]:
```
C1     16
C2      9
C3      2
C4     12
C5      4
C6      1
C7     13
C8     19
Name: R3, dtype: int32
```

In [122]:

Out[122]:

	C4	C5	C6	C7
R6	13	6	12	18
R7	8	13	10	17

29. Fill in the Missing Data

In [128]:
```python
# Create the Data Frame below
```

In [129]:

Out[129]:

	A	B	C	D
0	NaN	10.0	25.0	20
1	25.0	20.0	NaN	40
2	85.0	30.0	55.0	60
3	50.0	NaN	NaN	80
4	NaN	50.0	NaN	100

In [ ]:
```python
# Replace missing values of column A using .mean
```

In [ ]:

In [130]:
```python
# Replace missing values of column B using .median
```

In [ ]:

In [131]:
```python
# Replace missing values of column C using random numbers (whole numbers only)
# from 20 to 60, Hint: use the .apply() method
```

In [ ]:

30. GroupBy

```
In [8]:    # Load data set advertising.csv (see our github Data Set folder)
```

Out[8]:

	Daily Time Spent on Site	Age	Area Income	Daily Internet Usage	Ad Topic Line	City	Male	Country	Timestamp
0	68.95	35	61833.90	256.09	Cloned 5thgeneration orchestration	Wrightburgh	0	Tunisia	3/27/2016 0:53
1	80.23	31	68441.85	193.77	Monitored national standardization	West Jodi	1	Nauru	4/4/2016 1:39
2	69.47	26	59785.94	236.50	Organic bottom-line service-desk	Davidton	0	San Marino	3/13/2016 20:35
3	74.15	29	54806.18	245.89	Triple-buffered reciprocal time-frame	West Terrifurt	1	Italy	1/10/2016 2:31
4	68.37	35	73889.99	225.58	Robust logistical utilization	South Manuel	0	Iceland	6/3/2016 3:36
...	...	...	...	...	...	...	...	...	...
995	72.97	30	71384.57	208.58	Fundamental modular algorithm	Duffystad	1	Lebanon	2/11/2016 21:49
996	51.30	45	67782.17	134.42	Grass-roots cohesive monitoring	New Darlene	1	Bosnia and Herzegovina	4/22/2016 2:07
997	51.63	51	42415.72	120.37	Expanded intangible solution	South Jessica	1	Mongolia	2/1/2016 17:24
998	55.55	19	41920.79	187.95	Proactive bandwidth-monitored policy	West Steven	0	Guatemala	3/24/2016 2:35
999	45.01	26	29875.80	178.35	Virtual 5thgeneration emulation	Ronniemouth	0	Brazil	6/3/2016 21:43

1000 rows × 10 columns

31. Based on the loaded data set, show the following:

- use groupby, sort_values, head

```
In [10]:   # Average age of Male (1) and Female (0)
```

Out[10]:   Male
           0   36.186898
           1   35.817048
           Name: Age, dtype: float64

```
In [26]:   # Country with the highest daily time spent on the site
```

Out[26]:   Country
           French Polynesia   91.43
           Name: Daily Time Spent on Site, dtype: float64

```
In [25]:   # Top 10 countries with the lowest daily time spent on the site
```

Out[25]:   Country
           Kiribati   36.37
           Marshall Islands   43.16
           Sao Tome and Principe   44.49
           Lithuania   46.98
           Romania   49.99
           Saint Kitts and Nevis   50.52
           Andorra   53.33
           British Indian Ocean Territory (Chagos Archipelago)   54.70
           New Caledonia   56.30
           Montserrat   56.64
           Name: Daily Time Spent on Site, dtype: float64

```
In [20]:   # How many Male (1) and Female are on the data set
```

Out[20]:   Male
           0   519
           1   481
           Name: Age, dtype: int64

```
In [37]:   # Which top 5 city in total has the highest daily internet usage
```

Out[37]:   City
           Williamsport   520.02
           Lisamouth   491.30
           Wrightburgh   449.89
           Benjaminchester   434.68
           Lake Patrick   430.09
           Name: Daily Internet Usage, dtype: float64

```
In [42]:   # Top 3 Country that clicked on an ad
```

Out[42]:   Country
           Australia   7
           Turkey   7
           Ethiopia   7
           Name: Clicked on Ad, dtype: int64
```

## 32. Exploratory Data Analysis

- you can use any method as long as the answer is correct
- you can use combination of functions, groupby, data frame filtering, etch., whatever method you will prefer
- remember if you are confused do not force to code at once, break down the problem into pieces
- i will not show you the answers, you are on your own on this one

In [46]: # Load data set summer.csv (see our github Data Set folder)

Out[46]:

| | Year | City | Sport | Discipline | Athlete | Country | Gender | Event | Medal |
|---|------|------|-------|------------|---------|---------|--------|-------|-------|
| 0 | 1896 | Athens | Aquatics | Swimming | HAJOS, Alfred | HUN | Men | 100M Freestyle | Gold |
| 1 | 1896 | Athens | Aquatics | Swimming | HERSCHMANN, Otto | AUT | Men | 100M Freestyle | Silver |
| 2 | 1896 | Athens | Aquatics | Swimming | DRIVAS, Dimitrios | GRE | Men | 100M Freestyle For Sailors | Bronze |
| 3 | 1896 | Athens | Aquatics | Swimming | MALOKINIS, Ioannis | GRE | Men | 100M Freestyle For Sailors | Gold |
| 4 | 1896 | Athens | Aquatics | Swimming | CHASAPIS, Spiridon | GRE | Men | 100M Freestyle For Sailors | Silver |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 31160 | 2012 | London | Wrestling | Wrestling Freestyle | JANIKOWSKI, Damian | POL | Men | Wg 84 KG | Bronze |
| 31161 | 2012 | London | Wrestling | Wrestling Freestyle | REZAEI, Ghasem Gholamreza | IRI | Men | Wg 96 KG | Gold |
| 31162 | 2012 | London | Wrestling | Wrestling Freestyle | TOTROV, Rustam | RUS | Men | Wg 96 KG | Silver |
| 31163 | 2012 | London | Wrestling | Wrestling Freestyle | ALEKSANYAN, Artur | ARM | Men | Wg 96 KG | Bronze |
| 31164 | 2012 | London | Wrestling | Wrestling Freestyle | LIDBERG, Jimmy | SWE | Men | Wg 96 KG | Bronze |

31165 rows × 9 columns

In [182]: # How many cities host the Olympics based from the data set?

In [183]: # What are the sports competed based from the data set?

```python
In [184]: # What is the name of the athlete who hauls the most gold medal for the country USA?
```

```python
In [185]: # What are the top 5 countries who hauls the most medal (Gold, Silver, Bronze) in Weightlifting?
```

```python
In [186]: # What is the name of the top 3 athletes who hauls the most gold medal for the country CHN in Gymnastics?
```

```python
In [187]: # Which country has the 3rd highest collective Bronze medal tally in the dataset?
```

```python
In [188]: # How many Men and Women participated in the Volleyball event during the Los Angeles Olympics?
```

```python
In [189]: # Who won the gold in the 75 - 81KG (Light-Heavyweight) event of Boxing in the Athens Olympics?
```