CONFIDENTIAL

SCHOOL OF COMPUTING
Faculty of Engineering

UNIVERSITI TEKNOLOGI MALAYSIA

FINAL EXAM

SEMESTER I 2021/2022

SUBJECT CODE	**: SECP 3223**
SUBJECT NAME	**: DATA ANALYTICS PROGRAMMING**
TIME	**: 3 HOURS**
DATE	**:**

INSTRUCTION TO THE STUDENTS:

1. Answer all questions in Jupyter Notebook.
2. Save your answer notebook as *Name_MatricNo.*
3. There are two (2) submissions time as follow:
   i. Interim Submission at 11.30am
   ii. Final Submission at 1.10pm

   10 minutes are given for each submission. Please upload your answer file (*ipynb* format) in designated folder in e-Learning.

NAME	
IC NO. / MATRIC NO.	

(This question paper consists of 9 pages including this page)

QUESTION 1 **[25 MARKS]**

(a) Create a function name **verbing** which receive a string as the parameter. Upon receiving a string: (7 marks)

- if its length is at least 3: add *'ing'* to its end.
- Unless it already ends in *'ing'*, in which case add *'ly'* instead.
- If the string length is less than 3, leave it unchanged.
- Return the resulting string.

Test your code with these strings. You should get the outputs as follow:

Table 1

String	Expected Output
Hail	Hailing
Swimming	Swimmingly
Do	Do

(b) Convert the following code to list comprehension. (3 marks)

```
cords = [ ]
for x in range(4):
    for y in range(2):
        coordinate = (x, y)
        cords.append(coordinate)
print(cords)
```

(c) Given a paragraph as follows:

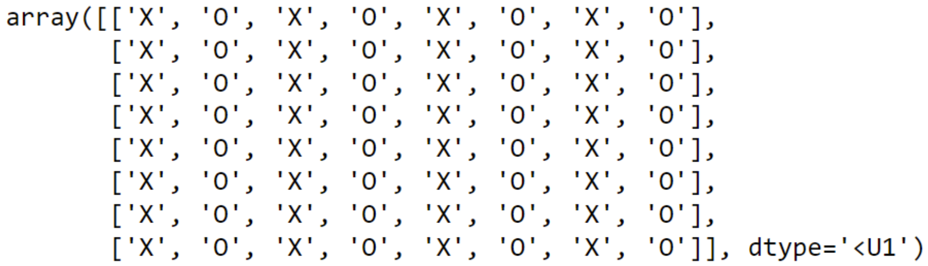
```
On most computer systems, localhost resolves to the IP address
10.100.11.121, which is the most commonly used IPv4 loopback
address, and to the IPv6 loopback address. The localhost IP address
is 192.168.11.10.
```

Using regular expression, write the python code that can find all the network IP address and replace it to 179.01.10.1. (5 marks)

(d) Using **Numpy**:

(i) Create a 10x10 array with random values and find the minimum, maximum and average values. (5 marks)

(ii) Create an 8×8 matrix and fill it with a checkerboard pattern. Output is shown in Figure 1. (5 marks)



```
array([['X', 'O', 'X', 'O', 'X', 'O', 'X', 'O'],
       ['X', 'O', 'X', 'O', 'X', 'O', 'X', 'O'],
       ['X', 'O', 'X', 'O', 'X', 'O', 'X', 'O'],
       ['X', 'O', 'X', 'O', 'X', 'O', 'X', 'O'],
       ['X', 'O', 'X', 'O', 'X', 'O', 'X', 'O'],
       ['X', 'O', 'X', 'O', 'X', 'O', 'X', 'O'],
       ['X', 'O', 'X', 'O', 'X', 'O', 'X', 'O'],
       ['X', 'O', 'X', 'O', 'X', 'O', 'X', 'O']], dtype='<U1')
```

Figure 1: Output with a checkerboard pattern

QUESTION 2 [25 MARKS]

Given a dataset (*Tallest Building.csv*) that contains the list of the world's tallest buildings.

(a) Read the dataset and save into a data frame named ***tallest***. Display the first 8 rows of the data frame. (2 marks)

(b) Find the number of rows and columns in ***tallest***. (1 mark)

(c) Find which columns have missing values and how many of them? (1 mark)

(d) Is there any duplicated data? Permanently remove the duplicate data if any. (2 marks)

(e) Permanently delete all rows which contains at most three-observation data. (3 marks)

(f) For missing data in column *Height (meters)*: (5 marks)

(i) Get the row index of all missing values in Height (meters) and save in a list named ***missing_Height***

(ii) Fill in the missing value with conversion of feet value in Height (feet) which having the index in ***missing_Height***. The conversion rate is 1 feet = 0.3 meters.

***Ignore any warning given by the Python*

(g) For missing data in column *City*: (5 marks)

(i) Get the row index of all missing values in City and save in a list named ***missing_City***

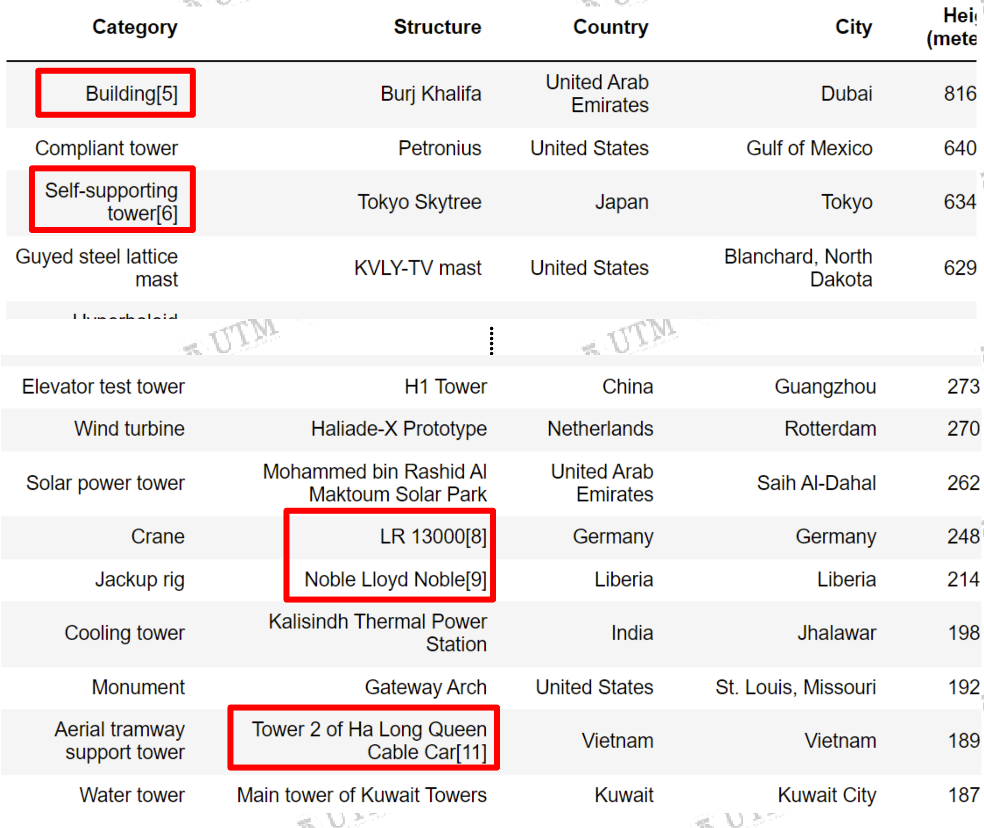
(ii) Fill in the missing value with the value in Country which having the index in ***missing_City***

***Ignore any warning given by the Python*

(h) Show that there are no missing values in tallest. (1 mark)

(i) Create a function named ***eliminate_ref***. This function will receive an array and perform these tasks: (5 marks)

(i) find whether there is any reference attached to any string in the array given. Commonly, any references can be detected by a square bracket containing a number like [5] as shown in Figure 2.



Category	Structure	Country	City	Hei (mete
Building[5]	Burj Khalifa	United Arab Emirates	Dubai	816
Compliant tower	Petronius	United States	Gulf of Mexico	640
Self-supporting tower[6]	Tokyo Skytree	Japan	Tokyo	634
Guyed steel lattice mast	KVLY-TV mast	United States	Blanchard, North Dakota	629
Hyperboloid				
	⋮			
Elevator test tower	H1 Tower	China	Guangzhou	273
Wind turbine	Haliade-X Prototype	Netherlands	Rotterdam	270
Solar power tower	Mohammed bin Rashid Al Maktoum Solar Park	United Arab Emirates	Saih Al-Dahal	262
Crane	LR 13000[8]	Germany	Germany	248
Jackup rig	Noble Lloyd Noble[9]	Liberia	Liberia	214
Cooling tower	Kalisindh Thermal Power Station	India	Jhalawar	198
Monument	Gateway Arch	United States	St. Louis, Missouri	192
Aerial tramway support tower	Tower 2 of Ha Long Queen Cable Car[11]	Vietnam	Vietnam	189
Water tower	Main tower of Kuwait Towers	Kuwait	Kuwait City	187

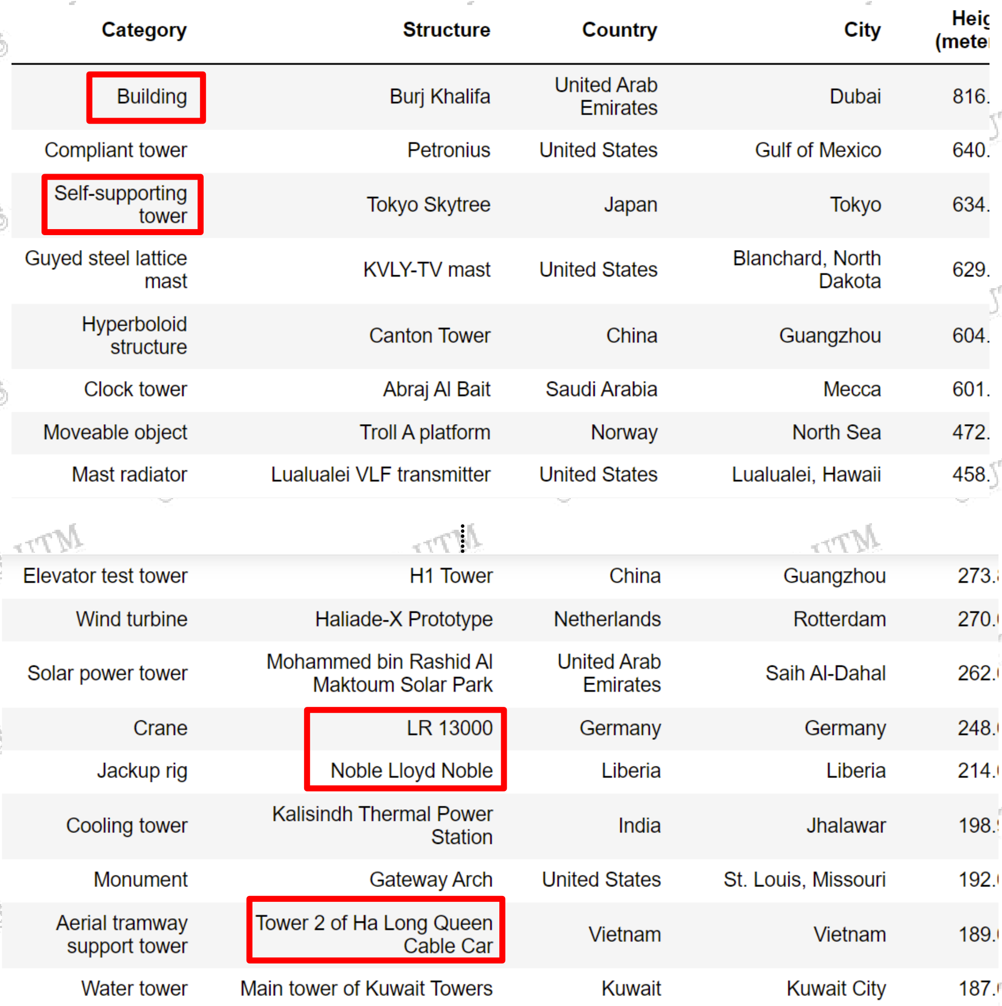
Figure 2: Reference attached to string

(ii) If there is any reference attached, replace the word with the original word but without the reference. Set *inplace=True* when performing the *replace* method.

Test your function using these codes

```
eliminate_ref(tallest['Category'])
eliminate_ref(tallest['Structure'])
tallest
```

The output should be as shown in Figure 3.



Category	Structure	Country	City	Heig (mete
Building	Burj Khalifa	United Arab Emirates	Dubai	816.
Compliant tower	Petronius	United States	Gulf of Mexico	640.
Self-supporting tower	Tokyo Skytree	Japan	Tokyo	634.
Guyed steel lattice mast	KVLY-TV mast	United States	Blanchard, North Dakota	629.
Hyperboloid structure	Canton Tower	China	Guangzhou	604.
Clock tower	Abraj Al Bait	Saudi Arabia	Mecca	601.
Moveable object	Troll A platform	Norway	North Sea	472.
Mast radiator	Lualualei VLF transmitter	United States	Lualualei, Hawaii	458.
⋮				
Elevator test tower	H1 Tower	China	Guangzhou	273.
Wind turbine	Haliade-X Prototype	Netherlands	Rotterdam	270.
Solar power tower	Mohammed bin Rashid Al Maktoum Solar Park	United Arab Emirates	Saih Al-Dahal	262.
Crane	LR 13000	Germany	Germany	248.
Jackup rig	Noble Lloyd Noble	Liberia	Liberia	214.
Cooling tower	Kalisindh Thermal Power Station	India	Jhalawar	198.
Monument	Gateway Arch	United States	St. Louis, Missouri	192.
Aerial tramway support tower	Tower 2 of Ha Long Queen Cable Car	Vietnam	Vietnam	189.
Water tower	Main tower of Kuwait Towers	Kuwait	Kuwait City	187.

Figure 3: The string without reference

QUESTION 3 **[30 MARKS]**

Given two dataset (*Death Male.xlsx* and *Death Female.xlsx*) that contains the data of recorded death in five states in Malaysia from 2011 to 2018.

(a) Task 1:

(i) Read the *Death Male.xlsx* and store it in a DataFrame named ***death_male***. Group the data by *Year* and name the result as ***dm_by_year*** and plot a pie chart as shown in Figure 4. (4 marks)

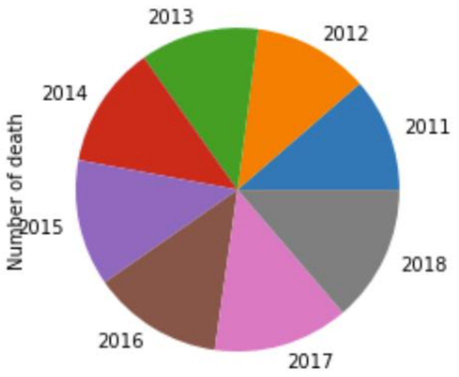


Figure 4: Pie Chart for ***dm_by_year***

(ii) Read the *Death Female.xlsx* and store it in a DataFrame named ***death_female***. Group the data by *State* and name the result as ***df_by_state*** and plot an area chart as shown in Figure 5. (4 marks)

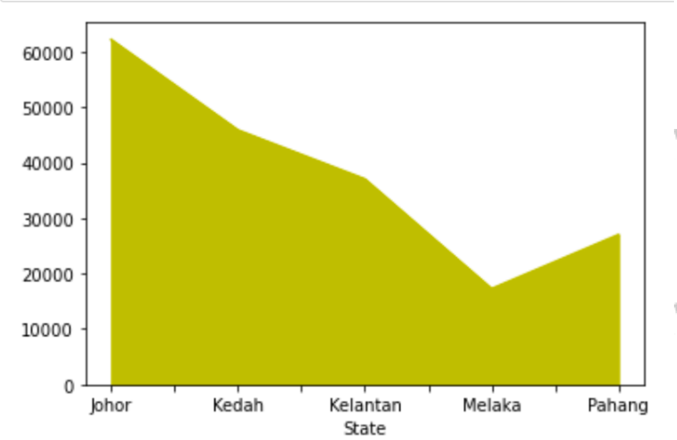


Figure 5: Area Chart for ***df_by_state***

(b) Task 2:

(i) Concatenate both ***death_male*** and ***death_female*** DataFrame and name the new DataFrame as ***death***. (1 mark)

(ii) Group the ***death*** by *State* and name the result as ***by_state***. (2 marks)

(iii) Group the ***death*** by *Year* name the result as ***by_year***. (2 marks)

(iv) Create a chart as shown in Figure 6 with figure size of 12 inches × 6 inches and save it as *Death in 2011-2018.png* with dpi value 200. (10 marks)

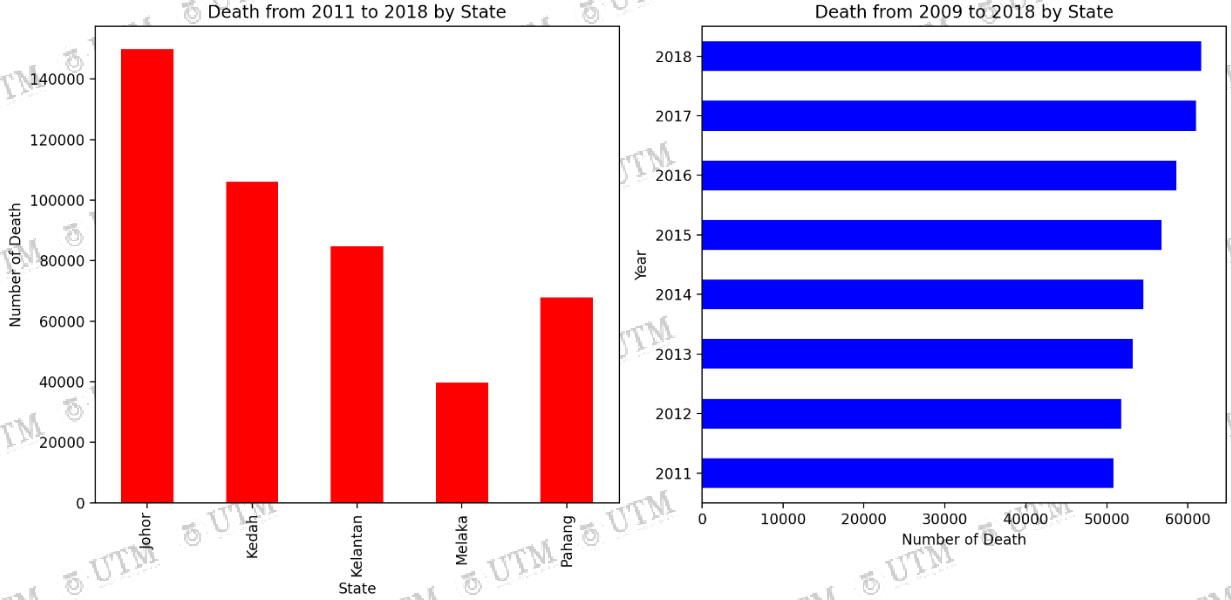
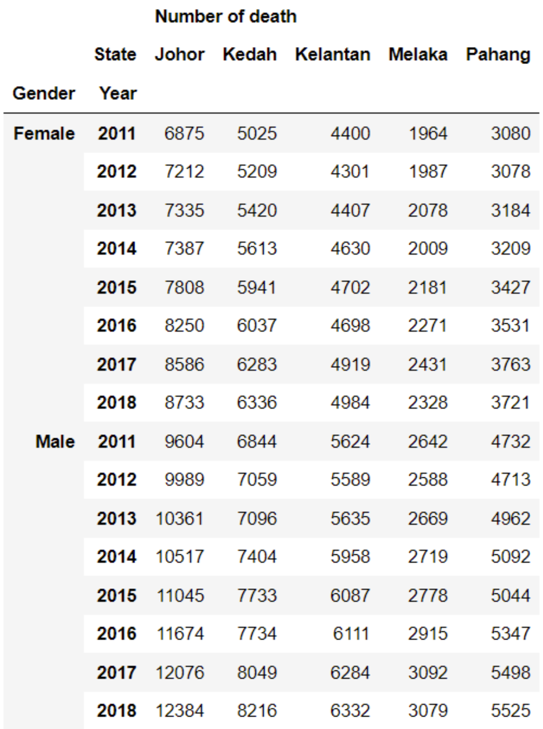


Figure 6: Death from 2011 to 2018 by State and Year

(c) Task 3:

(i) Create a pivot table with the name ***death_pivot*** as shown in Figure 7. (2 marks)



		Number of death				
	State	Johor	Kedah	Kelantan	Melaka	Pahang
Gender	Year					
Female	2011	6875	5025	4400	1964	3080
	2012	7212	5209	4301	1987	3078
	2013	7335	5420	4407	2078	3184
	2014	7387	5613	4630	2009	3209
	2015	7808	5941	4702	2181	3427
	2016	8250	6037	4698	2271	3531
	2017	8586	6283	4919	2431	3763
	2018	8733	6336	4984	2328	3721
Male	2011	9604	6844	5624	2642	4732
	2012	9989	7059	5589	2588	4713
	2013	10361	7096	5635	2669	4962
	2014	10517	7404	5958	2719	5092
	2015	11045	7733	6087	2778	5044
	2016	11674	7734	6111	2915	5347
	2017	12076	8049	6284	3092	5498
	2018	12384	8216	6332	3079	5525

Figure 7: A pivot table

(ii) Create a plot as shown in Figure 8 below. (5 marks)

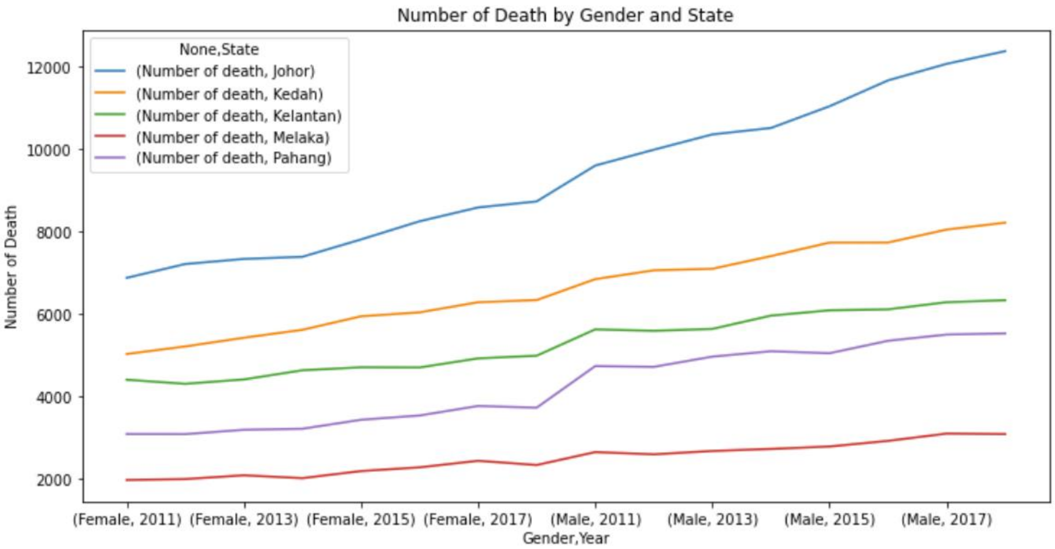


Figure 8: A plot based on ***death_pivot***

QUESTION 4 **[20 MARKS]**

(a) The file named *Expenditure.xlsx* describes the expenditure (in dollars) on recreation per month by employees at a certain company, and their corresponding monthly incomes.

(i) Using simple linear regression method, find the equation of the regression line. (8 marks)

(ii) Find the slope and interception values for the regression line. (1 mark)

(iii) Then estimate the monthly income of an employee at this company who spends 5000 dollars per month on recreation. (1 mark)

(b) Clustering Task:

(i) Load the *Titanic.csv* into a DataFrame named ***Titanic***. Below is the data dictionary for the dataset. (1 mark)

Table 2

Variable	**Definition**	**Key**
survived	**Survival**	**0 = No, 1 = Yes**
pclass	**Ticket class**	**1 = 1st, 2 = 2nd, 3 = 3rd**
sex	**Sex**	**0 = Male, 1 = Female**
age	**Age in years**	
sibsp	**Family Relations**	**0 = No family relation, 1 = Sibling, 2 = Spouse**
parch	**Family Relations**	**0 = No family relation, 1 = Parent, 2 = Child**
fare	**Passenger fare**	
cabin	**Cabin number**	**1 = Exist, -1 = Missing**
embarked	**Port of Embarkation**	**0 = Southampton, 1 = Cherbourg, 2 = Queenstown**

(ii) Perform dimensionality reduction to the dataset using the Principal Component Analysis (PCA) and next apply k-means clustering to the data. (9 marks)