Name Surname:

Student Number:

Section No:

10.01.2025

14:00 – 16:00 FCIS (BBBF) 206 – 207

COMP 303 – Advance Python Programming

Final Examination

Question 1 (24 pts):

Suppose you are in the library where there is **a stack of 10 books**. The librarian implements a rule that another student picks it when one student keeps the book. In order, the stack number remains unchanged. But since the librarian is not paying close attention. So, **the students break this order**, and **now the sequence is broken**. Now any number can keep the book simultaneously, while the same can keep it.

To solve this problem:

a) Write two functions with names ***add_book( )*** and ***take_book( )***, which keeps track of books taken, and the other one keeps.

b) Instantiate **two thread objects** where one simulates ***add_book( )*** and the other simulates ***take_book( )*** action respectively.

c) Print the final value of the stack number after running each thread in **part b**.

Question 2 (36 pts):

In this question, **Automobile Dataset** will be used for data analysis. This Dataset has different characteristics of an auto such as ***body-style***, ***wheel-base***, ***engine-type***, ***price***, ***average-mileage***, ***horsepower***, etc. Solve each part of the question based on the data in this Dataset.

a) From the given dataset, print the first and last five rows. Expected outputs:

	index	company	body-style	wheel-base	length	engine-type	num-of-cylinders	horsepower	average-mileage	price
0	0	alfa-romero	convertible	88.6	168.8	dohc	four	111	21	13495.0
1	1	alfa-romero	convertible	88.6	168.8	dohc	four	111	21	16500.0
2	2	alfa-romero	hatchback	94.5	171.2	ohcv	six	154	19	16500.0
3	3	audi	sedan	99.8	176.6	ohc	four	102	24	13950.0
4	4	audi	sedan	99.4	176.6	ohc	five	115	18	17450.0

	index	company	body-style	wheel-base	length	engine-type	num-of-cylinders	horsepower	average-mileage	price
56	81	volkswagen	sedan	97.3	171.7	ohc	four	85	27	7975.0
57	82	volkswagen	sedan	97.3	171.7	ohc	four	52	37	7995.0
58	86	volkswagen	sedan	97.3	171.7	ohc	four	100	26	9995.0
59	87	volvo	sedan	104.3	188.8	ohc	four	114	23	12940.0
60	88	volvo	wagon	104.3	188.8	ohc	four	114	23	13415.0

b) Print the most expensive car's ***company*** name and ***price***. Expected output:

	company	price
35	mercedes-benz	45400.0

c) Print all Toyota cars data details. Expected output:

	index	company	body-style	wheel-base	length	engine-type	num-of-cylinders	horsepower	average-mileage	price
48	66	toyota	hatchback	95.7	158.7	ohc	four	62	35	5348.0
49	67	toyota	hatchback	95.7	158.7	ohc	four	62	31	6338.0
50	68	toyota	hatchback	95.7	158.7	ohc	four	62	31	6488.0
51	69	toyota	wagon	95.7	169.7	ohc	four	62	31	6918.0
52	70	toyota	wagon	95.7	169.7	ohc	four	62	27	7898.0
53	71	toyota	wagon	95.7	169.7	ohc	four	62	27	8778.0
54	79	toyota	wagon	104.5	187.8	dohc	six	156	19	15750.0

d) Count total cars per ***company***. Expected output:

```
toyota          7
bmw             6
mazda           5
nissan          5
volkswagen      4
audi            4
mitsubishi      4
mercedes-benz   4
chevrolet       3
porsche         3
jaguar          3
honda           3
alfa-romero     3
isuzu           3
dodge           2
volvo           2
Name: company, dtype: int64
```

e) Find each ***company***'s highest ***price*** car. Expected output:

company	company	price
alfa-romero	alfa-romero	16500.0
audi	audi	18920.0
bmw	bmw	41315.0
chevrolet	chevrolet	6575.0
dodge	dodge	6377.0
honda	honda	12945.0
isuzu	isuzu	6785.0
jaguar	jaguar	36000.0
mazda	mazda	18344.0
mercedes-benz	mercedes-benz	45400.0
mitsubishi	mitsubishi	8189.0
nissan	nissan	13499.0
porsche	porsche	37028.0
toyota	toyota	15750.0
volkswagen	volkswagen	9995.0
volvo	volvo	13415.0

f) Find the ***average-mileage*** of each car making ***company***. Expected output:

company	average-mileage
alfa-romero	20.333333
audi	20.000000
bmw	19.000000
chevrolet	41.000000
dodge	31.000000
honda	26.333333
isuzu	33.333333
jaguar	14.333333
mazda	28.000000
mercedes-benz	18.000000
mitsubishi	29.500000
nissan	31.400000
porsche	17.000000
toyota	28.714286
volkswagen	31.750000
volvo	23.000000

g) Sort all cars by ***price*** column. Expected output:

	index	company	body-style	wheel-base	length	engine-type	num-of-cylinders	horsepower	average-mileage	price
35	47	mercedes-benz	hardtop	112.0	199.2	ohcv	eight	184	14	45400.0
11	14	bmw	sedan	103.5	193.8	ohc	six	182	16	41315.0
34	46	mercedes-benz	sedan	120.9	208.1	ohcv	eight	184	14	40960.0
46	62	porsche	convertible	89.5	168.9	ohcf	six	207	17	37028.0
12	15	bmw	sedan	110.0	197.0	ohc	six	182	15	36880.0

h) Create five ***data frames*** using the following five dictionaries. Then ***concatenate*** five ***data frames*** and create a key for each ***data frame***.

GermanCars = {'Company': ['Volkswagen', 'Mercedes', 'BMW', 'Audi', 'Opel'], 'Price': [23845, 171995, 135925 , 71400, 18830]}

JapaneseCars = {'Company': ['Toyota', 'Honda', 'Nissan', 'Mitsubishi'], 'Price': [29995, 23600, 61500 , 58900]}

AmericanCars = {'Company': ['Chevrolet', 'Dodge', 'Ford', 'Tesla', 'Chrysler', 'Cadillac'], 'Price': [19645, 21550, 22800, 45600, 32450, 67500]}

ItalianCars = {'Company': ['Alfa Romeo', 'Maserati', 'Fiat', 'Ferrari', 'Lamborghini', 'Lancia', 'Pagani'], 'Price': [17300, 225600, 13500, 955000, 849500, 42765, 775750]}

FrenchCars = {'Company': ['Renault', 'Citroen', 'Peugeot', 'Bugatti'], 'Price': [20450, 16570, 18155 , 1250450]}

		Company	Price
Germany	0	Ford	23845
	1	Mercedes	171995
	2	BMV	135925
	3	Audi	71400
Japan	0	Toyota	29995
	1	Honda	23600
	2	Nissan	61500
	3	Mitsubishi	58900

i) Create two ***data frames*** using the following two dictionaries. Then ***merge*** two ***data frames*** and append the second data frame as a new column to the first data frame.

Car_Price = {'Company': ['Toyota', 'Honda', 'BMW', 'Audi'], 'Price': [23845, 17995, 135925 , 71400]}

Car_Horsepower = {'Company': ['Toyota', 'Honda', 'BMW', 'Audi'], 'Horse-power': [141, 80, 182 , 160]}

	Company	Price	horsepower
0	Toyota	23845	141
1	Honda	17995	80
2	BMV	135925	182
3	Audi	71400	160

Question 3 (40 pts):

In this question, you need to visit the following webpage:

https://coinmarketcap.com/historical/

Historical Snapshot - 05 January 2025

Market Cap: All | Price: All | Volume (24h): All

USD | ← Previous Week | View All

Rank	Name	Symbol	Market Cap	Price	Circulating Supply	volume (24h)	% 1h	% 24h	% 7d
1	Bitcoin	BTC	$1,947,252,730,094.69	$98,314.96	19,806,271 BTC	$20,525,254,824.64	-0.35%	0.08%	5.12%
2	Ethereum	ETH	$437,815,296,057.84	$3,634.10	120,474,080 ETH *	$12,830,306,907.97	-0.26%	-0.65%	8.50%
3	XRP	XRP	$137,768,877,225.67	$2.3997	57,410,227,039 XRP *	$4,125,625,773.45	-0.03%	-0.82%	14.65%
4	Tether USDt	USDT	$137,251,157,315.40	$0.9999	137,271,150,956 USDT *	$64,028,140,855.09	0.01%	-0.01%	0.17%
5	Solana	SOL	$103,068,854,420.13	$213.39	482,998,742 SOL *	$2,403,765,298.79	-0.33%	-1.49%	12.46%
6	BNB	BNB	$102,137,951,637.67	$709.26	144,006,710 BNB *	$1,411,808,758.78	0.11%	-0.64%	2.23%
7	Dogecoin	DOGE	$56,429,516,618.45	$0.3826	147,503,706,384 DOGE	$2,594,267,525.17	-0.18%	-3.09%	21.71%
8	USDC	USDC	$45,614,641,261.00	$1.0001	45,611,801,903 USDC *	$3,837,973,654.76	0.01%	-0.01%	<0.01%

a. Download the web page and create an instance of the **BeautifulSoup** class to parse the page.

b. Write a Python function that iterates over each cryptocurrency elements in the given table (**URL**, **date** and **number of cryptocurrency elements** should be given as parameters to this function for getting different tables that include **weekly changed datasets of a particular date**) and returns a group of lists which includes each information about each cryptocurrency elements as a different list. Then, create a ***dataframe*** containing a different list of each piece of information respectively by using the ***Pandas*** library and display all stored information in it.

If we print out that Pandas dataframe, the output should look like:

```
Command Prompt

E:\Fall2024\Fall2024_COMP303_LectureNotes_Slides_LabAssignments\Fall2024_COMP303_S1S2_FinalExam_10012025>python FQ3.py

           Name Symbol            Market Cap        Price   Circulating Supply              Volume
0       Bitcoin    BTC  $1,947,252,730,094.69   $98,314.96           19,806,271  $20,525,254,824.64
1      Ethereum    ETH    $437,815,296,057.84    $3,634.10          120,474,080  $12,830,306,907.97
2           XRP    XRP    $137,768,877,225.67       $2.3997       57,410,227,039   $4,125,625,773.45
3   Tether USDt   USDT    $137,251,157,315.40       $0.9999      137,271,150,956  $64,028,140,855.09
4        Solana    SOL    $103,068,854,420.13     $213.39            482,998,742   $2,403,765,298.79
5           BNB    BNB    $102,137,951,637.67     $709.26            144,006,710   $1,411,808,758.78
6      Dogecoin   DOGE     $56,429,516,618.45       $0.3826      147,503,706,384   $2,594,267,525.17
7          USDC   USDC     $45,614,641,261.00       $1.0001       45,611,801,903   $3,837,973,654.76
8       Cardano    ADA     $38,436,618,849.44       $1.0935       35,151,153,813   $1,241,263,862.09
9          TRON    TRX     $22,637,910,683.26       $0.2626       86,191,829,447     $573,935,339.05
10    Avalanche   AVAX     $17,671,334,468.43         $43.08          410,205,396     $394,208,894.71
11          Sui    SUI     $15,769,236,666.25       $5.2397        3,009,569,342   $1,301,513,431.18
12    Chainlink   LINK     $15,081,641,043.56         $23.64          638,099,970     $503,863,576.16
13      Toncoin    TON     $14,540,299,012.20       $5.7257        2,539,486,512     $133,098,202.04
14    Shiba Inu   SHIB     $14,102,304,945.32  $0.00002393  589,255,325,935,474     $386,456,257.53
15      Stellar    XLM     $13,373,293,388.16       $0.4407       30,346,238,890     $554,743,054.93
16     Polkadot    DOT     $11,794,642,489.85       $7.6791        1,535,937,768     $233,645,714.92
17       Hedera   HBAR     $11,525,246,861.91       $0.3013       38,252,162,539     $512,694,156.84
18  Bitcoin Cash    BCH      $9,329,609,963.77        $470.91           19,811,819     $256,046,098.70
19       Uniswap    UNI      $9,099,651,024.14         $15.15          600,483,074     $247,754,591.99

E:\Fall2024\Fall2024_COMP303_LectureNotes_Slides_LabAssignments\Fall2024_COMP303_S1S2_FinalExam_10012025>
```

Good Luck