Q1. Correct way to form a dataframe?

Which of these would be the **correct** way to create a **dataframe**?

a.

```
df = pd.DataFrame([[1, 2], ["Ram", "Shyam"], ["IT", "Ops"]], columns = ["emp_id", "name", "dept"])
```

b.

```
df = pd.DataFrame([[1, "Ram", "IT"], [2, "Shyam", "Ops"]], columns = ["emp_id", "name", "dept"])
```

c.

```
df = pd.DataFrame([1, "Ram", "IT"], columns = ["emp_id", "name", "dept"])
```

d.

```
df = pd.DataFrame({'emp_id':[1, 2], 'name': ['Ram', 'Shyam'], 'dept':['IT', 'Ops']})
```

*There may be more than one correct option to this question so, select all which you feel correct.

Q2. As a Series

ou're working on collecting the text data for a Natural Language Processing(NLP) project. You come up with the idea of storing the unique words (case-sensitive) with their frequency in a Pandas Series object.

You are given the raw data in form of a string, Write a function which can take a string as an input and return the unique words and the corresponding frequency in form of a Pandas Series object.

The indices of the series should be the unique words and the values should be the frequency of those unique words.

Note that:

1. String contains no special character.
2. Always a Non-empty string.
3. Case sensitive i.e. He and he should be treated as two different word tokens.
4. Series returned is expected to be sorted by sort_index()for sorting all the words.

Input Format

```
Number of testcases
String with space separated words. (basically a sentence)
```

Output Format

```
space separated words in first line.
space separated values in the second line.
```

Sample Input

```
1
How much wood would a woodchuck chuck if a woodchuck could chuck wood
```

Sample Output

```
How          1
a            2
chuck        2
could        1
if           1
much         1
wood         2
woodchuck    2
would        1
```

```python
import pandas as pd
def solve(string):
    """
    returns a series object with word count as values and words as the indices.
    """
    # store the frequency of the strings in a series here

    # sort the indices here

    return res
```

Q3. Delete Constant columns

Given a dataframe named df, what will be the output of the following code:

```
for x in df.columns:
    if df[x].nunique()==1:
        df.drop(x, axis=1, inplace=True)
```

A. Given for loop will delete all the columns having only one unique value.
B. Given code will give an error
C. Given code will not make any changes in the original dataframe.
D. None of these

Q4. Add new columns

Given a dataframe named diamond, We want to add a new column named 'imported' (Either we can add it at the end or at the given position of the column)

We have a list called **imported** of length 53940 having values either 0 or 1 ; Here, values represent if the diamond is imported or not (0 = not imported, 1 = imported).

Which of the following statements are true?

a. diamond["imported"] = imported will add column named "imported" to diamond at the last position
b. diamond.insert(3, "imported", imported) will add column named "imported" to diamond at column index 3
c. diamond.insert(10, "imported", imported) will add column named "imported" at the last position

Note:

Google about how .insert function works in pandas

	carat	cut	color	clarity	depth	table	price	x	y	z
0	0.23	Ideal	E	SI2	61.5	55.0	326	3.95	3.98	2.43
1	0.21	Premium	E	SI1	59.8	61.0	326	3.89	3.84	2.31
2	0.23	Good	E	VS1	56.9	65.0	327	4.05	4.07	2.31
3	0.29	Premium	I	VS2	62.4	58.0	334	4.20	4.23	2.63
4	0.31	Good	J	SI2	63.3	58.0	335	4.34	4.35	2.75
...	...	...	...	...	...	...	...	...	...	...
53935	0.72	Ideal	D	SI1	60.8	57.0	2757	5.75	5.76	3.50
53936	0.72	Good	D	SI1	63.1	55.0	2757	5.69	5.75	3.61
53937	0.70	Very Good	D	SI1	62.8	60.0	2757	5.66	5.68	3.56
53938	0.86	Premium	H	SI2	61.0	58.0	2757	6.15	6.12	3.74
53939	0.75	Ideal	D	SI2	62.2	55.0	2757	5.83	5.87	3.64

A. a,b
B. a,c
C. a,b,c
D. b,c

Q5. Data Extraction

Given the following data frame "df", which of the following command(s) is/are the **correct** way to extract the mentioned columns in the order: **time**, **total_bill**, **tip**?

Note: If required the dataset can be downloaded from **here**.

	total_bill	tip	sex	smoker	day	time	size
0	16.99	1.01	Female	No	Sun	Dinner	2
1	10.34	1.66	Male	No	Sun	Dinner	3
2	21.01	3.50	Male	No	Sun	Dinner	3
3	23.68	3.31	Male	No	Sun	Dinner	2
4	24.59	3.61	Female	No	Sun	Dinner	4

A. pd.DataFrame(df, columns=['time', 'total_bill', 'tip'])
B. df[['time', 'total_bill', 'tip']]
C. df.loc[:, ['time', 'total_bill', 'tip']]
D. df.iloc[:,0:2]

*There may be more than one correct option to this question so, select all which you feel correct.

Q6. loc and iloc

Given, a dataframe **df**:

	Name	Gender	Profession
0	Jim	M	Athelete
1	Carrie	F	Tech
2	Chris	M	Cricketer
3	Morris	M	Actor

The following codes are executed on the data frame **df:**

df.iloc[:2,:2] #line a
df.loc[:2,"Name":"Profession"] #line b

From the above-given information, mark the option which is **true** regarding the following statements.

1. For line **a**, the output is the first two rows with the three columns ["Name", "Gender", "Profession"].

2. For line **a**, the output is the first two rows with the two columns ["Name", "Gender"].

3. For line **b**, the output is the row with labels 0, 1, and 2 with the columns ["Name", "Gender", "Profession"].

4. For line **b**, TypeError will be generated.

A. Only statements 1 and 3 are true.
B. Only statements 1 and 4 are true.
C. Only statements 2 and 4 are true.
D. Only statements 2 and 3 are true.

Q7. Get a hold of your data

```
import pandas as pd
def add_and_remove(df, data, out):
    '''
    Input:
    df -> The dataframe passed as input
    data -> The list of list variable containg the rows to append
    out -> the variable containing the row index value to be removed

    Output:
    df -> return the dataframe df after doing the above operations
    '''
    # Add the rows

    # Remove the out index row

return df
```

Problem Description:

Given a dataframe, a list of rows in the format of list of lists, and a number, "out".

Perform the following operations:

- Append the rows from the list of lists to the dataframe
- After appending, remove the row at the out position

Input Format:

A dataframe

A list of lists

A variable "out"

Output Format:

A dataframe

Sample Input:

	name	age
0	a	12
1	b	15
2	c	18

[['d', 20], ['e', 21], ['f', 22]]

4

Sample Output:

	name	age
0	a	12
1	b	15
2	c	18
3	d	20
5	f	22

Sample Explanation:

First the 3 rows are added to the existing dataframe, then the row at the 4th index (explicit) is dropped

Note:

out points to the explicit index position of the dataframe

Q8. Display all the rows

Given the dataset of 10 car models and their respective features:

model	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
Mazda RX4	21.0	6	160.0	110	3.90	2.620	16.46	0	1	4	4
Mazda RX4 Wag	21.0	6	160.0	110	3.90	2.875	17.02	0	1	4	4
Datsun 710	22.8	4	108.0	93	3.85	2.320	18.61	1	1	4	1
Hornet 4 Drive	21.4	6	258.0	110	3.08	3.215	19.44	1	0	3	1
Hornet Sportabout	18.7	8	360.0	175	3.15	3.440	17.02	0	0	3	2
Valiant	18.1	6	225.0	105	2.76	3.460	20.22	1	0	3	1
Duster 360	14.3	8	360.0	245	3.21	3.570	15.84	0	0	3	4
Merc 240D	24.4	4	146.7	62	3.69	3.190	20.00	1	0	4	2
Merc 230	22.8	4	140.8	95	3.92	3.150	22.90	1	0	4	2
Merc 280	19.2	6	167.6	123	3.92	3.440	18.30	1	0	4	4

What would be the correct code to print every row of the "**disp**" column?

The dataset link can be found **here**.

Note:

- Here, "data" variable is the DataFrame.

A. data.loc[:, 'disp']
B. data['disp']
C. data.iloc[:, 'disp']
D. data.loc['disp', :]

*There may be more than one correct option to this question so, select all which you feel correct.

Q9. Delete Duplicates

Given a pandas dataframe named "**titanic**", it contains duplicate rows and columns.

You can download the dataset shown in the image below from **here**.

Below are a few operations performed on this dataframe to remove the duplicates:

1. titanic.T.drop_duplicates(keep="first")
2. titanic.drop_duplicates(keep ="first")
3. titanic.drop_duplicates(keep ="last")

Based on the above operations, choose which of the followings are true.

A. Output of first operation contain index ["survived", "pclass", "sex", "age", "sibsp", "parch", "fare", "embarked", "class", adult_male", "embark_town"]

B. Output of second operation contains index [0,1,3,4,6,8]

C. Output of first operation contains index [0,1,2,3,4,5,6,7,8,9]

D. Output of the third operation contains index [2,5,6,7,8,9]

E. Output of first operation contain only columns ["pclass_1", "price","gender"]

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	adult_male	embark_town	pclass_1	price	gender
0	1	3	female	26.0	0	0	7.9250	S	Third	False	Southampton	3	7.9250	female
1	1	1	female	38.0	1	0	71.2833	C	First	False	Cherbourg	1	71.2833	female
2	1	3	female	26.0	0	0	7.9250	S	Third	False	Southampton	3	7.9250	female
3	0	3	male	NaN	0	0	8.4583	Q	Third	True	Queenstown	3	8.4583	male
4	1	2	female	14.0	1	0	30.0708	C	Second	False	Cherbourg	2	30.0708	female
5	0	3	male	NaN	0	0	8.4583	Q	Third	True	Queenstown	3	8.4583	male
6	0	1	male	54.0	0	0	51.8625	S	First	True	Southampton	1	51.8625	male
7	1	1	female	38.0	1	0	71.2833	C	First	False	Cherbourg	1	71.2833	female
8	1	3	female	27.0	0	2	11.1333	S	Third	False	Southampton	3	11.1333	female
9	1	2	female	14.0	1	0	30.0708	C	Second	False	Cherbourg	2	30.0708	female

Note: .T is used to transpose the dataframe.

A. A, B
B. B, C, E
C. A, B, D
D. C, E

Q10. Satisfied customers

Given a dataframe below about customer review & ratings.

	name	profession	gender	age	review	rating
0	Sam	dev	male	21	No comments	10
1	Roma	mle	female	20	hardworker	5
2	Mark	Data scientist	male	25	need improvement	7

Return a subset of the dataframe with records having **rating >= 6**, containing the columns "**profession**", "**gender**" and "**age**" only.

Input Format:

A DataFrame

Output Format:

Subset Dataframe

Sample Input:

{'name':["Sam","Roma","Mark"], "profession":['dev','mle','Data scientist'],"gender":['male','female','male'], "age":[21,20,25],"review":['No comments','hardworker','need improvement'],"rating":[10,5,7]}

Sample Output:

	profession	gender	age
0	dev	male	21
2	Data scientist	male	25

Sample Explanation:

The first and third rows with names 'Sam' and 'Mark' have ratings greater than or equal to 6.

```
import pandas as pd
def filtered_customers(df):
    ''' df is a dataframe with columns ['name', 'profession', 'gender', 'age', 'review', 'rating']
        Output -> A dataframe with required rows is expected to be returned'''

    # YOUR CODE GOES HERE

    # Filter the dataframe having ratings>=6 and choose the required columns

    return new_df
```

Q11. Select the required data

Given the dataset of 10 car models and their respective features:

model	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
Mazda RX4	21.0	6	160.0	110	3.90	2.620	16.46	0	1	4	4
Mazda RX4 Wag	21.0	6	160.0	110	3.90	2.875	17.02	0	1	4	4
Datsun 710	22.8	4	108.0	93	3.85	2.320	18.61	1	1	4	1
Hornet 4 Drive	21.4	6	258.0	110	3.08	3.215	19.44	1	0	3	1
Hornet Sportabout	18.7	8	360.0	175	3.15	3.440	17.02	0	0	3	2
Valiant	18.1	6	225.0	105	2.76	3.460	20.22	1	0	3	1
Duster 360	14.3	8	360.0	245	3.21	3.570	15.84	0	0	3	4
Merc 240D	24.4	4	146.7	62	3.69	3.190	20.00	1	0	4	2
Merc 230	22.8	4	140.8	95	3.92	3.150	22.90	1	0	4	2
Merc 280	19.2	6	167.6	123	3.92	3.440	18.30	1	0	4	4

What would be the correct code to get the resultant subset as the below image?

model	hp	drat
Mazda RX4 Wag	110	3.90
Datsun 710	93	3.85
Hornet 4 Drive	110	3.08
Hornet Sportabout	175	3.15

The dataset link can be found **here**.

Note:

1. Here, "data" variable is the DataFrame.
2. Please set the 'model' column as index while using the above dataset.

A. data.iloc[1:5, 3:5]
B. data.iloc[1:4, 3:4]
C. data.loc[1:5, 3:5]
D. data.loc[1:6, 3:6]