Started on	Monday, 19 May 2025, 1:44 PM
State	Finished
Completed on	Monday, 19 May 2025, 4:18 PM
Time taken	2 hours 34 mins
Overdue	34 mins 8 secs
Grade	**80.00** out of 100.00

Question **1**

Not answered

Mark 0.00 out of 20.00

Create a pandas program to filter words that contain atleast 2 vowels from a series?

For example:

Input	Result
['Apple', 'Orange', 'Plan', 'Python', 'Money']	Apple Orange Money

Answer: (penalty regime: 0 %)

```
1
```

Question **2**

Correct

Mark 20.00 out of 20.00

Create a numpy program to find the sum of last column in a given numpy array.

For example:

Input	Result
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]	[[1 2 3] [4 5 6] [7 8 9] [10 11 12]] 30

Answer: (penalty regime: 0 %)

```
import numpy as np
a=eval(input())
a=np.array(a).reshape(4,3)
print(a)
b=a[:,-1]
c=np.sum(b)
print(c)
```

	Input	Expected	Got	
✔	[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]	[[1 2 3] [4 5 6] [7 8 9] [10 11 12]] 30	[[1 2 3] [4 5 6] [7 8 9] [10 11 12]] 30	✔
✔	[2,4,6,8,10,12,14,16,18,20,22,24]	[[2 4 6] [8 10 12] [14 16 18] [20 22 24]] 60	[[2 4 6] [8 10 12] [14 16 18] [20 22 24]] 60	✔

Passed all tests! ✔

Correct

Marks for this submission: 20.00/20.00.

Question **3**

Correct

Mark 20.00 out of 20.00

Write a Numpy program to Convert a 1-D array into a 2-D array with 3 rows

For example:

Input	Result
[0, 1, 2, 3, 4, 5, 6, 7, 8] The original array: [0 1 2 3 4 5 6 7 8]	The original array: [0 1 2 3 4 5 6 7 8] 3 x 3 Array: [[0 1 2] [3 4 5] [6 7 8]]

Answer: (penalty regime: 0 %)

```
import numpy as np
a=eval(input())
arr = np.array(a)
reshaped_arr = arr.reshape(3, -1)
print("The original array:")
print("",arr)
print("\n3 x 3 Array:")
print("",reshaped_arr)
```

	Input	Expected	Got	
✔	[0, 1, 2, 3, 4, 5, 6, 7, 8] The original array: [0 1 2 3 4 5 6 7 8]	The original array: [0 1 2 3 4 5 6 7 8] 3 x 3 Array: [[0 1 2] [3 4 5] [6 7 8]]	The original array: [0 1 2 3 4 5 6 7 8] 3 x 3 Array: [[0 1 2] [3 4 5] [6 7 8]]	✔
✔	[11, 12, 13, 14, 15, 16, 17, 18, 19]	The original array: [11 12 13 14 15 16 17 18 19] 3 x 3 Array: [[11 12 13] [14 15 16] [17 18 19]]	The original array: [11 12 13 14 15 16 17 18 19] 3 x 3 Array: [[11 12 13] [14 15 16] [17 18 19]]	✔

Passed all tests! ✔

Correct

Marks for this submission: 20.00/20.00.

Question **4**

Correct

Mark 20.00 out of 20.00

Create a python program to create a student.csv file and read the content from the newly created file

For example:

Test	Input	Result

`with open("studentsq.csv", 'r') as file:`
`  csvreader = csv.reader(file)`
`  header = next(csvreader)`
`  for row in csvreader:`
`  rows.append(row)`
`print(header)`
`print(rows)`
	`[["SNo", "Name", "Subject"],[1, "Ash Ketchum", "English"],[2, "Gary Oak", "Mathematics"],[3, "Brock Lesner", "Physics"]]`	`['SNo', 'Name', 'Subject']`
		`[['1', 'Ash Ketchum', 'English'], ['2', 'Gary Oak', 'Mathematics'], ['3', 'Brock Lesner', 'Physics']]`

Answer: (penalty regime: 0 %)

```
import csv
rows=[]
data = eval(input())
with open("studentsq.csv", "w", newline="") as csvfile:
    writer = csv.writer(csvfile)
    writer.writerows(data)
```

	Test	Input	Expected	Got	

✔	with open("studentsq.csv", 'r') as file:
	  csvreader = csv.reader(file)
	  header = next(csvreader)
	  for row in csvreader:
	
	rows.append(row)
	print(header)
	print(rows)
		[["SNo", "Name", "Subject"],[1, "Ash Ketchum", "English"],[2, "Gary Oak", "Mathematics"],[3, "Brock Lesner", "Physics"]]	['SNo', 'Name', 'Subject']
			[['1', 'Ash Ketchum', 'English'], ['2', 'Gary Oak', 'Mathematics'], ['3', 'Brock Lesner', 'Physics']]
				['SNo', 'Name', 'Subject']
				[['1', 'Ash Ketchum', 'English'], ['2', 'Gary Oak', 'Mathematics'], ['3', 'Brock Lesner', 'Physics']]
					✔

✔	with open("studentsq.csv", 'r') as file:
	  csvreader = csv.reader(file)
	  header = next(csvreader)
	  for row in csvreader:
	
	rows.append(row)
	print(header)
	print(rows)
		[["SNo", "Name", "Subject"],[11, "Anto Ketchum", "DAA"],[21, "Geek Oak", "PYTHON"],[31, "John Lesner", "JAVA"]]	['SNo', 'Name', 'Subject']
			[['11', 'Anto Ketchum', 'DAA'], ['21', 'Geek Oak', 'PYTHON'], ['31', 'John Lesner', 'JAVA']]
				['SNo', 'Name', 'Subject']
				[['11', 'Anto Ketchum', 'DAA'], ['21', 'Geek Oak', 'PYTHON'], ['31', 'John Lesner', 'JAVA']]
					✔

Passed all tests! ✔

Correct

Marks for this submission: 20.00/20.00.

Question **5**

Correct

Mark 20.00 out of 20.00

Create a Pandas program to join the two dataframes with matching records from both sides where available.

For example:

Input	Result

{'s_id': ['S1','S2','S3','S4','S5'],'name': ['Danniella Fenton', 'Ryder Storey', 'Bryce Jensen', 'Ed Bernal', 'Kwame Morin'],'marks': [200, 210, 190, 222, 199]}
{'s_id': ['S4', 'S5', 'S6', 'S7', 'S8'],'name': ['Scarlette Fisher', 'Carla Williamson', 'Dante Morse', 'Kaiser William', 'Madeeha Preston'],'marks': [201, 200, 198, 219, 201]}
	Original DataFrames:
	  s_id name marks
	0 S1 Danniella Fenton 200
	1 S2 Ryder Storey 210
	2 S3 Bryce Jensen 190
	3 S4 Ed Bernal 222
	4 S5 Kwame Morin 199
	  s_id name marks
	0 S4 Scarlette Fisher 201
	1 S5 Carla Williamson 200
	2 S6 Dante Morse 198
	3 S7 Kaiser William 219
	4 S8 Madeeha Preston 201
	Merged data (outer join):
	  s_id name_x marks_x name_y marks_y
	0 S1 Danniella Fenton 200.0 NaN NaN
	1 S2 Ryder Storey 210.0 NaN NaN
	2 S3 Bryce Jensen

Input	Result
	190.0 NaN NaN 3 S4 Ed Bernal 222.0 Scarlette Fisher 201.0 4 S5 Kwame Morin 199.0 Carla Williamson 200.0 5 S6 NaN NaN Dante Morse 198.0 6 S7 NaN NaN Kaiser William 219.0 7 S8 NaN NaN Madeeha Preston 201.0

Answer: (penalty regime: 0 %)

```
import pandas as pd
s1=pd.DataFrame(eval(input()))
s2=pd.DataFrame(eval(input()))
print("Original DataFrames:")
print(s1)
print(s2)
r=pd.merge(s1,s2,on ="s_id",how="outer")
print("Merged data (outer join):")
print(r)
```

	Input	Expected	Got	
✔	{'s_id': ['S1','S2','S3','S4','S5'],'name': ['Danniella Fenton', 'Ryder Storey', 'Bryce Jensen', 'Ed Bernal', 'Kwame Morin'],'marks': [200, 210, 190, 222, 199]} {'s_id': ['S4', 'S5', 'S6', 'S7', 'S8'],'name': ['Scarlette Fisher', 'Carla Williamson', 'Dante Morse', 'Kaiser William', 'Madeeha Preston'],'marks': [201, 200, 198, 219, 201]}	Original DataFrames: s_id name marks 0 S1 Danniella Fenton 200 1 S2 Ryder Storey 210 2 S3 Bryce Jensen 190 3 S4 Ed Bernal 222 4 S5 Kwame Morin 199 s_id name marks 0 S4 Scarlette Fisher 201 1 S5 Carla Williamson 200 2 S6 Dante Morse 198 3 S7 Kaiser William 219 4 S8 Madeeha Preston 201 Merged data (outer join): s_id name_x marks_x name_y marks_y 0 S1 Danniella Fenton 200.0	Original DataFrames: s_id name marks 0 S1 Danniella Fenton 200 1 S2 Ryder Storey 210 2 S3 Bryce Jensen 190 3 S4 Ed Bernal 222 4 S5 Kwame Morin 199 s_id name marks 0 S4 Scarlette Fisher 201 1 S5 Carla Williamson 200 2 S6 Dante Morse 198 3 S7 Kaiser William 219 4 S8 Madeeha Preston 201 Merged data (outer join): s_id name_x marks_x name_y marks_y 0 S1 Danniella Fenton 200.0	✔

	Input	Expected	Got	
		NaN NaN 1 S2 Ryder Storey 210.0 NaN NaN 2 S3 Bryce Jensen 190.0 NaN NaN 3 S4 Ed Bernal 222.0 Scarlette Fisher 201.0 4 S5 Kwame Morin 199.0 Carla Williamson 200.0 5 S6 NaN NaN Dante Morse 198.0 6 S7 NaN NaN Kaiser William 219.0 7 S8 NaN NaN Madeeha Preston 201.0	NaN NaN 1 S2 Ryder Storey 210.0 NaN NaN 2 S3 Bryce Jensen 190.0 NaN NaN 3 S4 Ed Bernal 222.0 Scarlette Fisher 201.0 4 S5 Kwame Morin 199.0 Carla Williamson 200.0 5 S6 NaN NaN Dante Morse 198.0 6 S7 NaN NaN Kaiser William 219.0 7 S8 NaN NaN Madeeha Preston 201.0	

Passed all tests! ✔

Correct

Marks for this submission: 20.00/20.00.