Indian Institute of Information Technology, Design & Manufacturing, Kurnool (IIITDMK)

Department of Computer Science and Engineering

Machine Learning Practice-JULY- NOV 2025 (S5-B.Tech CSE)

Assignment 2

Last date for submitting: 08/08/2025 18:00 Hrs

1. DataFrame Column Operations with Pandas

Problem Statement:
You are given a DataFrame containing student names, their math scores, and science scores. Calculate the total score (math + science) for each student and print the updated DataFrame.

Input Format:
First line: integer n, the number of students.
Next n lines: each containing name, math_score, science_score separated by spaces.

Constraints:
$1 \le n \le 1000$
Scores are integers between 0 and 100

Output Format:
Print the updated DataFrame with columns: name, math_score, science_score, total_score.

Sample Input:
3
Alice 80 90
Bob 70 85
Charlie 90 95

Sample Output:

	name	math_score	science_score	total_score
0	Alice	80	90	170
1	Bob	70	85	155
2	Charlie	90	95	185

2. Filter Rows in Pandas DataFrame

Problem Statement:
Given a DataFrame of employees with columns name and salary, filter and print the rows where salary is greater than a given threshold t.

Input Format:
First line: integer n, the number of employees.
Next n lines: each containing name salary.
Last line: integer t, the threshold salary.

Constraints:
$1 \le n \le 1000$
$1 \le salary, t \le 10^5$

Output Format:
Print the filtered DataFrame. If no employees match the condition, print an empty DataFrame.

Sample Input:
4
John 50000
Mary 70000
Sam 40000
Anna 90000
60000

Sample Output:
```
  name  salary
1  Mary   70000
3  Anna   90000
```

3. Group By and Aggregate in Pandas

Problem Statement:
You are given a DataFrame with department and salary columns. Compute and print the average salary for each department.

Input Format:

First line: integer n, the number of rows.
Next n lines: each containing department salary.

Constraints:
$1 \leq n \leq 1000$
$1 \leq salary \leq 10^5$

Output Format:
Print the resulting DataFrame with department and avg_salary columns.

Sample Input:
6
IT 60000
HR 40000
IT 70000
Finance 50000
HR 45000
Finance 55000

Sample Output:

	department	avg_salary
0	Finance	52500.0
1	HR	42500.0
2	IT	65000.0

4. Sorting a DataFrame

Problem Statement:
Given a DataFrame with columns product and price, sort the DataFrame by price in descending order and print it.

Input Format:
First line: integer n, the number of products.
Next n lines: each containing product price.

Constraints:
$1 \leq n \leq 1000$
$1 \leq price \leq 10^5$

Output Format:
Print the sorted DataFrame.

Sample Input:
4
Laptop 70000
Phone 30000
Tablet 20000
Monitor 15000

Sample Output:
  product price
0 Laptop 70000
1   Phone 30000
2 Tablet 20000
3 Monitor 15000

5. Merging Two DataFrames

Problem Statement:
You are given two DataFrames: one containing employee_id and name, and the other containing employee_id and salary. Merge the two DataFrames on employee_id and print the final DataFrame.

Input Format:
First line: integer n, the number of rows in the first DataFrame.
Next n lines: employee_id name.
Next line: integer m, the number of rows in the second DataFrame.
Next m lines: employee_id salary.

Constraints:
$1 \leq n, m \leq 1000$

Output Format:
Print the merged DataFrame.

Sample Input:
3

101 John
102 Mary
103 Sam
3
101 50000
102 70000
103 40000

Sample Output:

	employee_id	name	salary
0	101	John	50000
1	102	Mary	70000
2	103	Sam	40000

6. Multi-level Grouping and Aggregation

Problem Statement:
You are given a dataset of employee details with columns: department, gender, and salary. Using Pandas, calculate the average salary for each department and gender combination and print the result as a DataFrame.

Input Format:
First line: integer n, the number of employees.
Next n lines: department gender salary

Constraints:
$1 \leq n \leq 1000$
department: string
gender: M or F
salary: integer ($1 \leq salary \leq 10^5$)

Output Format:
Print the grouped DataFrame with columns department, gender, avg_salary.

Sample Input:
6
IT M 60000
HR F 40000
IT F 70000
Finance M 50000

HR M 45000
Finance F 55000

Sample Output:

	department	gender	avg_salary
0	Finance	F	55000.0
1	Finance	M	50000.0
2	HR	F	40000.0
3	HR	M	45000.0
4	IT	F	70000.0
5	IT	M	60000.0

7. Pivot Table Creation

Problem Statement:
You are given sales data with columns: `region`, `product`, and `sales`. Create a pivot table where:

- Rows represent region,
- Columns represent product,
- Values are the sum of sales,
- Missing values should be filled with 0.

Note: Not every region will have sales for every product. Ensure missing region-product combinations are represented with 0 in the pivot table.

Input Format:

First line: integer `n`, number of sales records.

Next `n` lines: region product sales

Constraints:

$1 \leq n \leq 1000$

region, product: strings

sales: integer ($1 \leq sales \leq 10^4$)

Output Format:
Print the pivot table, with:

Rows sorted in ascending order of region (default behavior),

Columns as product names (sorted alphabetically),

Missing combinations filled with `0`.

Sample Input:

7

North Laptop 200

South Laptop 300

North Phone 150

South Phone 250

East Laptop 100

East Phone 120

West Phone 180

Sample Output:

product Laptop Phone

region

East 100 120

North 200 150

South 300 250

West 0 180

8. Filter Big Countries

Problem Statement:

You are given data about countries, including their name, continent, area, population, and GDP. A country is considered **big** if:

- Its **area** is at least **3,000,000** km^2, or
- Its **population** is at least **25,000,000**.

Your task is to:

1. Take the number of rows as input.
2. Read each country's details from input.
3. Filter out only the **big countries**.
4. Print the resulting DataFrame containing only these columns:
   name, population, area — along with the default index.

Input Format

- First line: An integer n, the number of countries
- Next n lines: Each line contains — name continent area population gdp
  (All values are separated by spaces)

Constraints

- $1 \leq n \leq 1000$
- name, continent: strings with no spaces
- $1 \leq area \leq 10^7$
- $1 \leq population \leq 10^9$
- $1 \leq gdp \leq 10^{13}$

Output Format

Print the filtered DataFrame with columns: name, population, and area (including the index).

Sample Input:

5

Afghanistan Asia 652230 25500100 20343000000

Albania Europe 28748 2831741 12960000000

Algeria Africa 2381741 37100000 188681000000

Andorra Europe 468 78115 3712000000

Angola Africa 1246700 20609294 100990000000

Sample Output:

```
      name  population    area
0  Afghanistan   25500100   652230
1     Algeria   37100000  2381741
```

9. Boolean Filtering on Multiple Conditions

Problem Statement:
You are given information about various products including their IDs, whether they are low in fats, and whether they are recyclable. A product is considered eligible if:

- It is **low fat** (low_fats = 'Y'), **and**
- It is **recyclable** (recyclable = 'Y').

Write a program that reads product data from input, filters out only the eligible products, and returns a DataFrame containing just their product IDs.

Input Format:

- First line: Integer n, the number of product entries.
- Next n lines: Each line contains the following space-separated values:
  product_id low_fats recyclable
  where:
  - product_id is an integer (1 ≤ product_id ≤ 10000)
  - low_fats is a string ('Y' or 'N')
  - recyclable is a string ('Y' or 'N')

Output Format:
Print a DataFrame with a single column:

- product_id (of products that are both low fat and recyclable)
  Ensure that the output has a clean index starting from 0.

Sample Input:

```
5
0 Y N
```

1 Y Y

2 N Y

3 Y Y

4 N N

Sample Output:

product_id

0 1

1 3

10. Merging with Conditions

(The question is correct but the example output was incorrect, so it has been corrected. You have to sort them in descending order only)

Problem Statement:
You are given two DataFrames: Orders (order_id, customer_id, amount) and Customers (customer_id, name, city). Merge the two DataFrames on customer_id and filter only those customers whose total order amount is greater than 500. Print the final DataFrame sorted by total amount (descending).

Input Format:
First line: integer n, number of orders.
Next n lines: order_id customer_id amount
Next line: integer m, number of customers.
Next m lines: customer_id name city

Constraints:
$1 \leq n, m \leq 1000$

Output Format:
Print the merged and filtered DataFrame.

Sample Input:
4
1 101 300

2 101 400
3 102 200
4 103 600
3
101 John Delhi
102 Mary Mumbai
103 Sam Bangalore

Sample Output:

	customer_id	name	city	total_amount
0	101	John	Delhi	700
1	103	Sam	Bangalore	600

11. Handling Missing Data

Problem Statement:
You are given a dataset with columns name, age, and salary.
- Fill missing values in age with the mean age.
- Fill missing values in salary with 0.
- Print the final DataFrame.

Input Format:
First line: integer n, number of employees.
Next n lines: name age salary (if a value is missing, write NA)

Constraints:
$1 \leq n \leq 1000$
age, salary: integers or NA

Output Format:
Print the cleaned DataFrame.

Sample Input:
4
John 30 50000
Mary NA 70000

Sam 25 NA
Anna NA 90000

Sample Output:

	name	age	salary
0	John	30.0	50000.0
1	Mary	27.5	70000.0
2	Sam	25.0	0.0
3	Anna	27.5	90000.0

12. DateTime Operations in Pandas

Problem Statement:
You are given a dataset of date and sales. Convert the date column to datetime, extract the month name, and print the total sales for each month.

Input Format:
First line: integer n, number of records.
Next n lines: date sales (date in YYYY-MM-DD format).

Constraints:
$1 \le n \le 1000$
sales: integer ($1 \le sales \le 10^5$)

Output Format:
Print the total sales for each month (sorted by month order, Jan-Dec).

Sample Input:
5
2024-01-10 200
2024-01-25 300
2024-02-14 500
2024-02-28 400
2024-03-05 600

Sample Output:

	month	total_sales
0	Jan	500

1 Feb 900
2 Mar 600

13. Data Visualization with Pandas

Problem Statement:
You are given a dataset containing monthly sales for different regions.

- Convert the month column to datetime format.
- Group the data by month and calculate the total sales.

Input Format:

- First line: Integer n, the number of sales records.
- Next n lines: Each line contains a month and sales value (month in YYYY-MM format).

Constraints:

- $1 \leq n \leq 1000$
- sales: integer ($1 \leq sales \leq 10^5$)

Output Format:
Print the grouped DataFrame with columns month and total_sales.

Sample Input:

6

2024-01 200

2024-01 300

2024-02 400

2024-02 500

2024-03 600

2024-03 700

Sample Output:

month total_sales

0 2024-01-01 500

1 2024-02-01 900

2 2024-03-01 1300

14. String Operations in Pandas

Problem Statement:
You are given a dataset of product names in inconsistent formats. Perform the following operations:
1. Convert all names to lowercase.
2. Remove leading/trailing spaces.
3. Extract only the first word of each product name.
4. Count how many times each unique first word appears.
5. Print the resulting DataFrame sorted by count in descending order.

Input Format:
First line: integer n, number of products.
Next n lines: product names (strings, may contain spaces).

Constraints:
$1 \leq n \leq 1000$
Each name is at most 50 characters long.

Output Format:
Print a DataFrame with columns: first_word, count, sorted by count (descending).

Sample Input:
6
Apple iPhone 13
Samsung Galaxy S22
apple Watch
Samsung Galaxy S21
OnePlus Nord 2
apple AirPods

Sample Output:
first_word count
0 apple 3
1 samsung 2
2 oneplus 1

15. Nth Highest Salary

Problem Statement:

You are given employee salary data and a number **n**. Your task is to find the **nth highest distinct salary** from the given dataset.

If there are **fewer than n distinct salaries**, print null.

Input Format:

- First line: An integer m, the number of employee records
- Next m lines: Each line contains two values separated by a space:
  id salary
- Last line: An integer n, the rank of the highest distinct salary you need to find

Note: Salary values may repeat. Only **distinct** salaries should be considered when ranking.

Output Format:

- A single line containing the **nth highest distinct salary**, or null if it doesn't exist

---

Constraints:

- 1 ≤ m ≤ 1000
- 1 ≤ id ≤ 10000
- 1 ≤ salary ≤ 10^6
- 1 ≤ n ≤ 100

Sample Input:

3

1 100

2 200

3 300

2

Sample Output:

200

16. Rank Score

Problem Statement:

You are given a list of scores from a game. Your task is to rank these scores using the following rules:

1. Scores are ranked from **highest to lowest**.
2. If **two or more scores are equal**, they receive the **same rank**.
3. Use **dense ranking**, i.e., the next rank after a tie is the next consecutive number (no gaps).

Input Format:

- First line: An integer n, the number of score records
- Next n lines: Each line contains two values separated by a space:
  id score

Note: Scores are floating-point numbers rounded to two decimal places.

Output Format:

- n lines, each containing two columns:
  score rank
- Sorted in **descending order of score**

Constraints:

- $1 \leq n \leq 1000$
- $0.00 \leq score \leq 100.00$

- id values are unique integers

Sample Input:

6

1 3.50

2 3.65

3 4.00

4 3.85

5 4.00

6 3.65

Sample Output:

4.0 1

4.0 1

3.85 2

3.65 3

3.65 3

3.5 4

17. Count Salary Categories

Problem Statement:

You are given income data of several bank accounts. Your task is to **count how many accounts** fall into the following **salary categories**:

- **Low Salary**: income **strictly less than 20000**
- **Average Salary**: income in the **inclusive range [20000, 50000]**
- **High Salary**: income **strictly greater than 50000**

Your output must contain **all three categories**, even if the count for some is zero.

Input Format:

- First line: An integer n, the number of account records
- Next n lines: Each line contains two values separated by a space:
  account_id income

Output Format:

- Three lines, each containing two columns:
  category accounts_count
- The order of output can be any.

Constraints:

- 1 ≤ n ≤ 1000
- 0 ≤ income ≤ 10^6

Sample Input:

4

3 108939

2 12747

8 87709

6 91796

Sample Output:

Low Salary 1

Average Salary 0

High Salary 3

18. Managers with at least 5 direct reports.

Problem Statement:

You are given data about employees in a company. Each employee may or may not report to a manager. Your task is to **find the names of all employees who are managers with at least 5 direct reports.**

- A direct report means another employee whose managerId is equal to the manager's id.
- If an employee has **at least 5** such direct reports, they should be included in the result.

Input Format:

- First line: An integer n, the number of employee records
- Next n lines: Each line contains four values separated by spaces:
  id name department managerId
    - If managerId is "null", treat it as missing (no manager)

Output Format:

- Each line should contain a single name of a manager with at least 5 direct reports
- Output can be in any order

Constraints:

- $1 \leq n \leq 1000$
- id and managerId are integers
- name and department are non-empty strings
- No employee will be their own manager

Sample Input:

101 John A null

102 Dan A 101

103 James A 101

104 Amy A 101

105 Anne A 101

106 Ron B 101

Sample Output:

John