Final Assignment

You will design a small Data Explorer Program that allows users to load, search, and analyze text-based data (CSV or TXT). The program should demonstrate your ability to work with variables, data structures, loops, conditionals, functions, file input/output, Pandas, and regular expressions.

1. Setup and File Handling

- Write a function that asks the user for a filename (CSV or TXT).
- If the file exists, load its content.
- **EXTRA**: If the file does not exist, handle the error gracefully.

2. Menu System (Problem Solving + Loops + Conditionals)

Create a menu loop that lets the user choose between different actions until they decide to quit.
Example menu:

1. Show basic file information
2. Search for a keyword
3. Count word frequencies
4. Extract information with regular expressions
5. Load as a DataFrame and show basic stats (requirement: CSV file)
6. Quit

3. Strings, Lists, Dictionaries, and Sets

- Implement keyword search: ask the user for a word and show all lines containing it.
- Count how many times each unique word appears
  (e.g., dictionary for frequencies, set for unique words).

4. Functions & Modularization

- Each menu option must be implemented as a function.
- Organize the program into at least two modules (e.g., main.py and utils.py).

5. Pandas and DataFrames (for CSV files only)

- If the file is a CSV, load it into a Pandas DataFrame.
- Show:
  - First 5 rows
  - Summary statistics (e.g., *.describe()*)
  - Number of missing values per column.

6. Regular Expressions

Write a function that uses regex to:
- Extract all email addresses from the file (*if any*).
- **EXTRA**: Extract all dates (e.g., YYYY-MM-DD or DD/MM/YYYY).

7. Bonus Challenges (Optional for extra points)

- Generate a word cloud of the text.
- Save search results to a new file.
- Allow the user to choose columns when working with CSVs.

Submission Requirements

- Your program must run.
- Code should be clean, readable, and commented.
- Use functions wherever possible.
- Submit:
  - Your Python files (.py)
  - A short README.txt explaining how to run your program

Grading Criteria

- **30%** Code functionality (does it run correctly)
- **20%** Code clarity and readability
- **50%** Individual interview:
  - **20%** Briefly describe your project/code in one up to two minutes
  - **30%** Questions about your code

Reminder on Plagiarism

You are welcome to **ask anyone for help**, but you must:

- **Give credit** where it's due
- **Cite all sources** you use
- **Comment your code sufficiently** to show what you did and that you understand it.

! Do not present others' work as your own !
Failure to follow these rules will negatively affect your grade.