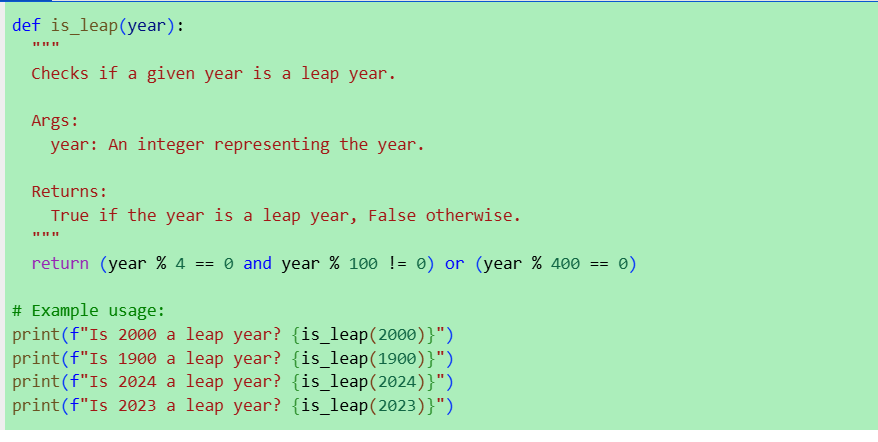
LAB Assignment (4.3)

Name: k. Sushanth

H.NO :2403A51348

Batch:14

Task Description#1  
• Zero-shot: Prompt AI to write a function that checks whether a given year is a leap  
year



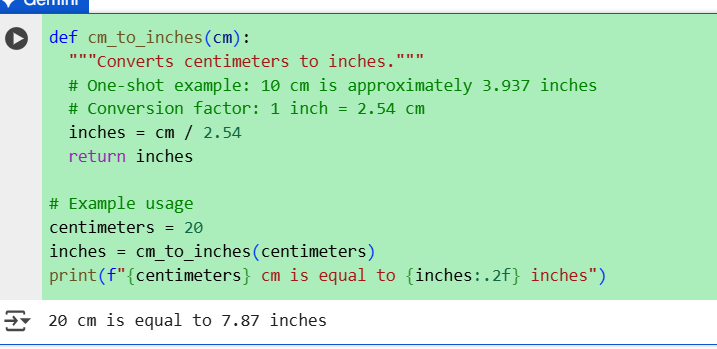
Observation:

1. A leap year occurs every 4 years to keep our calendar aligned with Earth's revolutions around the sun.
2. The function checks if a year is divisible by 4, but not by 100—unless it's also divisible by 400.
3. This logic ensures that century years like 1900 are not leap years, while 2000 is.

4.The function returns True for leap years and False otherwise, making it easy to use in programs.

5.It’s a handy tool for date calculations, calendar apps.

Task Description#2  
• One-shot: Give one input-output example to guide AI in writing a function that converts Centi meters to inches.



Observation:

1. This function helps convert lengths from centimeters to inches using a fixed conversion factor.

2. Since 1 centimeter equals approximately 0.3937 inches, the function multiplies the input by this value.

3. It's useful in applications where measurements need to be displayed in imperial units.

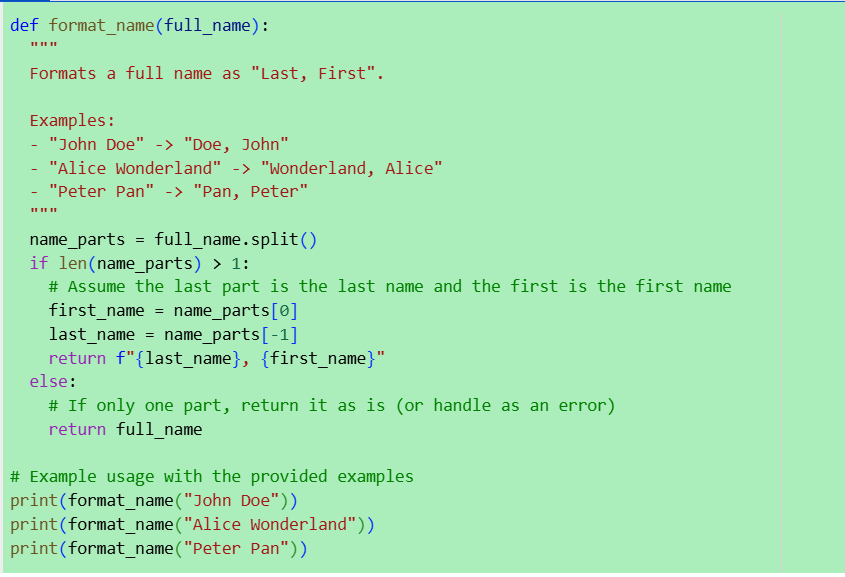
4. The function ensures accurate and consistent conversion for any numeric input.

5. It's a simple yet essential tool for international design,

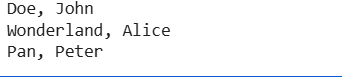
tailoring, or engineering tasks.

Task Description#3

Few-shot: Provide 2–3 examples to generate a function that formats full names as  
“Last, First”



OUTPUT:



Observation:

1. This function is designed to reformat names from the standard "First Last" structure into "Last, First", which is commonly used in formal documents, directories, and citations.

2. It works by splitting the input string into two parts and rearranging them using string formatting. The few-shot examples help illustrate the pattern clearly, making it easy for both humans and models to understand the transformation.

3. It's a simple yet effective way to standardize name formats across systems.

4.You could easily extend it to handle middle names or initials with a bit more logic.

Task Description#4

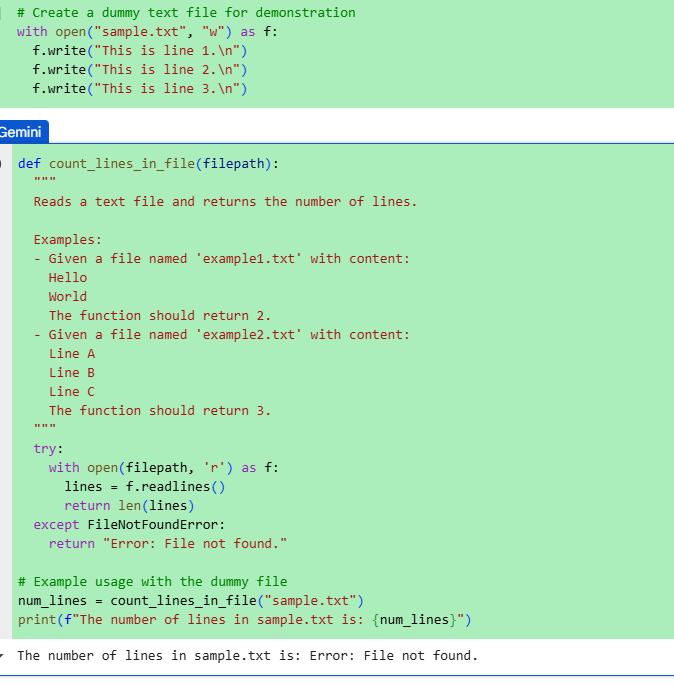
Compare zero-shot and few-shot prompts for writing a function that counts the number of vowels in a string.

Observation:

* **Generate zero-shot prompt and code**: Create a code cell with a zero-shot prompt and the resulting Python function to count vowels.
* **Generate few-shot prompt and code**: Create a code cell with a few-shot prompt (including examples) and the resulting Python function to count vowels.
* **Compare approaches**: Explain the differences between the zero-shot and few-shot approaches in terms of the prompt structure and how they might influence the generated code.

Task Description#5

Use few-shot prompting to generate a function that reads a .txt file and returns the number of lines.



Observation:

**1.**This function opens a .txt file in read mode and uses a generator expression to count each line efficiently.

2.The few-shot examples help establish the expected input-output making it easier for a model—or even a beginner—to infer the logic.

3. It's a clean and memory-friendly approach, especially useful for large files.

4. You can adapt it to count only non-empty lines or lines matching a pattern if needed.