ASSIGNMENT-4.3

Advanced Prompt Engineering – Zero-shot, One-shot, and Few-shot Techniques

---------------------------------------

**NAME: PRADEEP GUPTHA**

**HALL-TICKET: 2403A510C7**

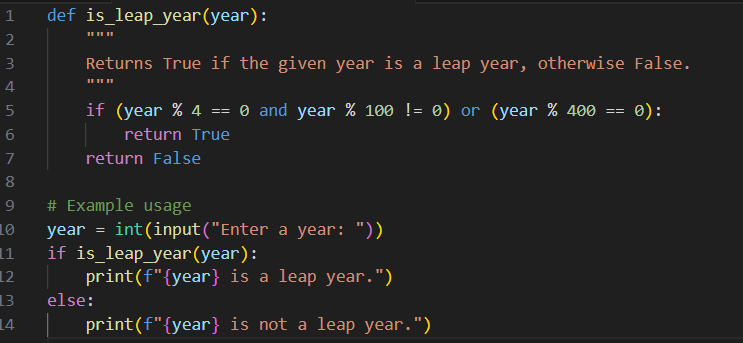
**BATCH NO: 05**

Task 1:   
● Zero-shot: Prompt AI to write a function that checks whether a given year is a leap  
year.

Prompt:

Zero-shot: write a function that checks whether a given year is a leap year. what does the prompt mean

CODE SCREENSHOT:



OUTPUT:



EXPLANATION:

The prompt **“Zero-shot: write a function that checks whether a given year is a leap year”** means:

* **Zero-shot** → You are asking the AI to solve a problem **without showing it any example** or prior context.
* The AI uses its **existing knowledge** to create a solution.

**How this relates to the prompt:**

1. The prompt asked for a function to check leap years.
2. The AI understood the rules of leap years and wrote this function **without seeing any example**—that’s **zero-shot**.
3. The function:
   * Takes a year as input.
   * Checks if it’s divisible by 4 but not by 100, **or** divisible by 400.
   * Returns True if leap year, otherwise False.

--------------------------------------------------------------------------------------------------------------

Task 2:

● One-shot: Give one input-output example to guide AI in writing a function that converts centimeters to inches.

Prompt:

write a function that converts centimeters to inches. For example : 1 inch = 2.54 centimeters.

CODE SCREENSHOT:

A computer screen shot of a program

AI-generated content may be incorrect.

Output:



Explanation:

Here’s what **“one-shot”** means using your example prompt:

* **One-shot** means the AI is asked to solve a task **after seeing exactly one example** of how it should be done.
* Unlike **zero-shot** (no examples) or **few-shot** (multiple examples), here you give **just one sample** to guide the AI.

A screenshot of a computer

AI-generated content may be incorrect.

----------------------------------------------------------------------------------------------------------

Task 3:

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

PROMPT:

Examples:

Input: "John Smith" → Output: "Smith, John"

Input: "Alice Johnson" → Output: "Johnson, Alice"

Input: "Michael Jordan" → Output: "Jordan, Michael"

Now, write a function that takes a full name and returns it in the format "Last, First".

CODE:

A screen shot of a computer program

AI-generated content may be incorrect.

OUTPUT:

A black background with white text

AI-generated content may be incorrect.

EXPLANATION:

1. **strip()** removes any extra spaces at the start or end.
2. **split()** breaks the string into a list of words using spaces.  
   Example: "John Smith" → ["John", "Smith"]
3. **Check if there are at least two words:**  
   If not, it returns the original input.
4. **first = parts[0]** stores the first word as the first name.
5. **last = " ".join(parts[1:])** joins all remaining words into one string (to support middle or double last names).
6. **return f"{last}, {first}"** returns the formatted string.

**Output Examples**

* Input: "John Smith" → Output: "Smith, John"
* Input: "Mary Ann Lee" → Output: "Ann Lee, Mary"

------------------------------------------------------------------------------------------------------------------------------------

TASK 4:

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

PROMPT:

i ) Write a Python function that counts the number of vowels (a, e, i, o, u) in a given string.

Ii ) Examples:

Input: "Hello" → Output: 2

Input: "Programming" → Output: 3

Input: "OpenAI" → Output: 3

Now write a Python function that counts the number of vowels (a, e, i, o, u) in any given string.

CODES:

A screen shot of a computer program

AI-generated content may be incorrect.

OUTPUT:



Code ii:

A screen shot of a computer code

AI-generated content may be incorrect.

Output ii:



EXPLANATION:

**1. Zero-shot scenario**

* **Prompt:** *“Write a Python function that counts the number of vowels (a, e, i, o, u) in a given string.”*
* **What happens:**  
  The AI is asked to solve the problem **without any examples**. It relies only on its general knowledge of Python and string handling.
  + Loops through each character.
  + Checks if it’s a vowel (case-insensitive).
  + Counts vowels and returns the result.
* **Example output:** "Programming" → 3

**2. Few-shot scenario**

* **Prompt:** *“Examples: ‘Hello’ → 2, ‘Programming’ → 3, ‘OpenAI’ → 3. Now write a Python function that counts the number of vowels in any string.”*
* **What happens:**  
  The AI is given **2–3 examples** before solving the problem. These examples show exactly what kind of input and output are expected. The AI uses these patterns to generate the function.
  + Uses a **compact approach** (generator expression) because the examples suggest a simple counting task.
  + Still counts vowels case-insensitively.
* **Example output:** "OpenAI" → 3

**Key Difference**

* **Zero-shot:** No examples, AI infers everything from the question. May write more step-by-step or longer code.
* **Few-shot:** Few examples help clarify expectations, often resulting in shorter or more precise code.

--------------------------------------------------------------------------------------------------------------

TASK 5:

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

Prompt:

Examples:

Input file content:

Line one

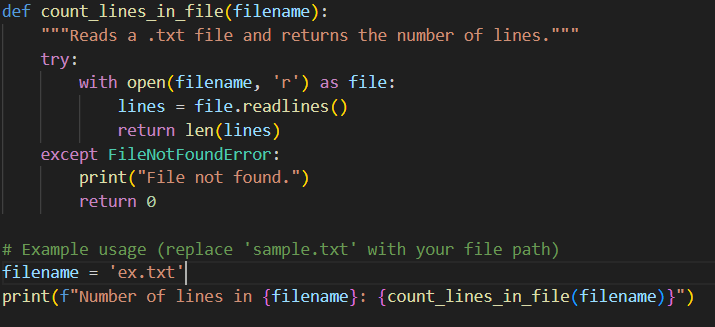
Line two

Line three

Output: 3

Now write a Python function that reads a .txt file and returns the number of lines.

CODE :



Txt file:



Output:



Explanation:

**Few-shot prompt meaning here**

* We give **2–3 examples** of what the input (file content) looks like and what output we expect (number of lines).
* Example shown in the prompt:
* Input file content:
* Hello world
* This is a test
* Output: 2
* These examples **teach the model the pattern** before asking it to generate the function. This is called **few-shot learning** because the model learns from a few examples.

**How it works:**

1. with open(filename, 'r') as file: – Safely opens the .txt file for reading.
2. file.readlines() – Reads the entire file into a list; each line is one element.
3. len(lines) – Counts how many elements (lines) are in the list.
4. try/except – Handles the case where the file is missing by showing a message and returning 0.