ASSIGNMENT – 4.3

HT.NO: 2303A52493

BATCH: 50

TASK-01

Question:

Task 1: Zero-Shot Prompting – Leap Year Check

Scenario

Zero-shot prompting involves giving instructions without providing examples.

Task Description

Use zero-shot prompting to instruct an AI tool to generate a Python function that:

• Accepts a year as input

• Checks whether the given year is a leap year

• Returns an appropriate result

Note: No input-output examples should be provided in the prompt.

Prompt:

Write a Python function that takes a year as input, checks whether it is a leap year using correct logic, and returns the result. Also show sample input and output.

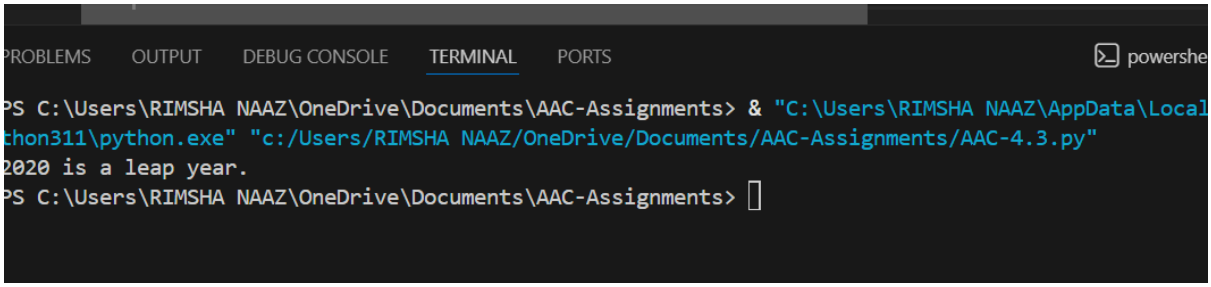
Generated Code:

```
def is_leap_year(year):
    """
    Check if a given year is a leap year.

    Args:
        year (int): The year to check.
        Returns:
        bool: True if leap year, False otherwise.
    """
    if (year % 4 == 0 and year % 100 != 0) or (year % 400 == 0):
        return True
```

```
    else:
        return False
# Sample Input and Output
year = 2020
if is_leap_year(year):
    print(f"{year} is a leap year.")
else:
    print(f"{year} is not a leap year.")
```

Output:



Explanation:

This function checks whether a year is a leap year using standard rules: a year must be divisible by 4, but not by 100 unless it is also divisible by 400.
If the condition is true, the function returns True; otherwise, it returns False.
The sample code tests the year 2020 and prints whether it is a leap year or not.

TASK-02:

Question:

Task 2: One-Shot Prompting – Centimeters to Inches Conversion

Scenario

One-shot prompting guides AI using a single example.

Task Description

Use one-shot prompting by providing one input-output example to generate a Python function that:

• Converts centimeters to inches

• Uses the correct mathematical formula

Example provided in prompt:

Input: 10 cm → Output: 3.94 inches

Prompt:

Write a Python function that converts centimeters to inches using the correct formula.

Example:

Input: 10 cm

Output: 3.94 inches

After writing the function, show a few sample test cases with their outputs

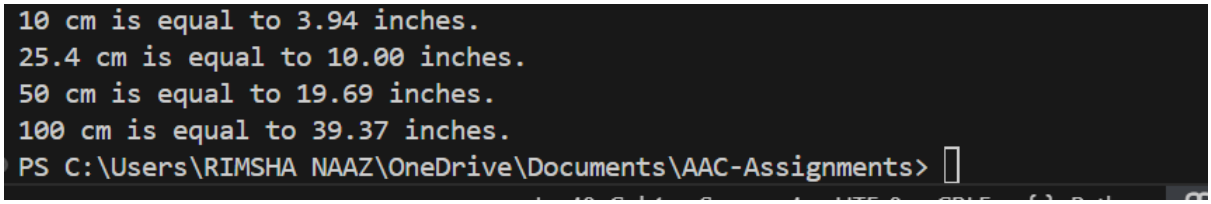
Generated Code:

```
def cm_to_inches(cm):
    """
    Convert centimeters to inches.

    Args:
        cm (float): Length in centimeters.

    Returns:
        float: Length in inches.
    """
    inches = cm / 2.54
    return inches
# Sample Test Cases
test_values = [10, 25.4, 50, 100]
for cm in test_values:
    inches = cm_to_inches(cm)
    print(f"{cm} cm is equal to {inches:.2f} inches.")
```

Output:



Explanation:

This function converts a value from centimeters to inches by dividing it by 2.54.
The sample code tests the function with different values and stores the result in inches.
It then prints the converted length in inches, rounded to two decimal places.

TASK-03:

Question:

Task 3: Few-Shot Prompting – Name Formatting

Scenario

Few-shot prompting improves accuracy by providing multiple examples.

Task Description

Use few-shot prompting with 2–3 examples to generate a Python function that:

• Accepts a full name as input

• Formats it as “Last, First”

Example formats:

• "John Smith" → "Smith, John"

• "Anita Rao" → "Rao, Anita"

Prompt:

Write a Python function that takes a full name as input and formats it as **“Last, First”**.

Examples:
Input: John Smith → Output: Smith, John
Input: Anita Rao → Output: Rao, Anita

Now apply the same format for other names.
After writing the function, show sample inputs and outputs.

Generated Code:

```
def format_name(full_name):
    """
```

```
    Format a full name as "Last, First".

    Args:
        full_name (str): The full name in "First Last" format.

    Returns:
        str: The formatted name in "Last, First" format.
    """
    parts = full_name.split()
    if len(parts) >= 2:
        first_name = parts[0]
        last_name = parts[-1]
        return f"{last_name}, {first_name}"
    else:
        return full_name  # Return as is if not enough parts
# Sample Inputs and Outputs
names = ["John Smith", "Anita Rao", "Michael Jordan", "Sara Connor"]
for name in names:
    formatted_name = format_name(name)
    print(f"Input: {name} -> Output: {formatted_name}")
```

Output:

```
Input: John Smith -> Output: Smith, John
Input: Anita Rao -> Output: Rao, Anita
Input: Michael Jordan -> Output: Jordan, Michael
Input: Sara Connor -> Output: Connor, Sara
```

Explanation:

This function takes a full name, splits it into parts, and rearranges it as **"Last, First."**
It uses the first word as the first name and the last word as the last name.
If the name has only one word, it returns the name without changing it.

TASK-04:

Question:

Task 4: Comparative Analysis – Zero-Shot vs Few-Shot

Scenario

Different prompt strategies may produce different code quality.

Task Description

• Use zero-shot prompting to generate a function that counts vowels in a string

• Use few-shot prompting for the same problem

• Compare both outputs based on:

o Accuracy

o Readability

o Logical clarity

Prompt-01:

Write a Python function that counts vowels in a string. Show sample input and output.

Generated Code:

```
def count_vowels(input_string):
    """
    Count the number of vowels in a given string.

    Args:
        input_string (str): The string to analyze.
    Returns:
        int: The count of vowels in the string.
    """
    vowels = "aeiouAEIOU"
    count = sum(1 for char in input_string if char in vowels)
    return count
# Sample Input and Output
sample_string = "Hello, World!"
```

vowel_count = count_vowels(sample_string)

print(f"The number of vowels in '{sample_string}' is: {vowel_count}")

Output:

```
The number of vowels in 'Hello, World!' is: 3
```

Explanation:

This function counts how many vowels are present in a given string.
It checks each character to see if it is a vowel (a, e, i, o, u in upper or lower case) and adds them up.
The sample code prints the total number of vowels in the given text.

Prompt-02:

Use the following examples and then write a Python function to count vowels in a string.
Examples:
Input: "hello" → Output: 2
Input: "AI" → Output: 2

Generated Code:

```
def count_vowels(input_string):
    """
    Count the number of vowels in a given string.

    Args:
        input_string (str): The string to analyze.
    Returns:
        int: The count of vowels in the string.
    """
    vowels = "aeiouAEIOU"
    count = sum(1 for char in input_string if char in vowels)
    return count
# Sample Inputs and Outputs
test_strings = ["hello", "AI", "Python Programming", "OpenAI"]
for s in test_strings:
```

```
vowel_count = count_vowels(s)
print(f"Input: '{s}' → Output: {vowel_count}")
```

Output:

```
Input: 'hello' → Output: 2
Input: 'AI' → Output: 2
Input: 'Python Programming' → Output: 4
Input: 'OpenAI' → Output: 4
```

Explanation:

This function counts vowels in a string by checking each character against a list of vowel letters.
It goes through each test string, calls the function, and counts how many vowels are found.
Finally, it prints each input string along with its vowel count.

TASK-05

Question:

Task 5: Few-Shot Prompting – File Handling

Scenario

File processing requires clear logical understanding.

Task Description

Use few-shot prompting to generate a Python function that:

• Reads a .txt file

• Counts the number of lines in the file

• Returns the line count

Prompt:

Write a Python function that reads a .txt file and counts the number of lines in it.

Examples:
File contains 3 lines → Output: 3
File contains 7 lines → Output: 7

The sample .txt file must be created and written within the Python code itself (no external files).

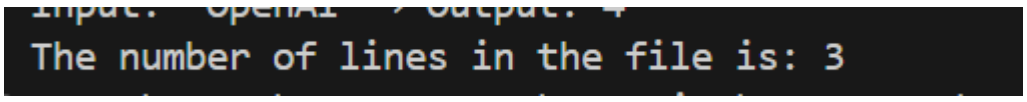
write the function.
After that, generate the sample file in code, show the function call, output, and briefly explain the logic.

Generated Code:

```
def count_lines_in_file(file_path):
    """
    Count the number of lines in a text file.

    Args:
        file_path (str): The path to the text file.
    Returns:
        int: The number of lines in the file.
    """
    with open(file_path, 'r') as file:
        lines = file.readlines()
        return len(lines)
# Create a sample .txt file
sample_file_path = 'sample.txt'
with open(sample_file_path, 'w') as file:
    file.write("This is the first line.\n")
    file.write("This is the second line.\n")
    file.write("This is the third line.\n")
# Call the function and display the output
line_count = count_lines_in_file(sample_file_path)
print(f"The number of lines in the file is: {line_count}")
```

Output:



Explanation:

This function opens a text file and counts how many lines it contains.
A sample file is created with three lines of text.
The function is then called, and it prints the total number of lines in the file.