

SCHOOL OF COMPUTER SCIENCE AND ARTIFICIAL INTELLIGENCE		DEPARTMENT OF COMPUTER SCIENCE ENGINEERING	
Program Name: M.Tech. and MCA		Assignment Type: Lab	AcademicYear: 2025-2026
Course Coordinator Name		Venkataramana Veeramsetty	
Course Code		Course Title	AI Assisted Problem Solving Using Python
Year/Sem	I/I	Regulation	R24
Date and Day of Assignment	Week3 - Monday	Time(s)	
Duration	2 Hours	Applicable to Batches	M.Tech. and MCA

AssignmentNumber: 4.3(Present assignment number)/24(Total number of assignments)

ASSIGNMENT – 4

Name: Rimsha Mujeeb

Roll Number:2503B05138(M.Tech -CSE)

Q.No.	Question
Q1.	<p>Task Description#1</p> <ul style="list-style-type: none"> Zero-shot: Prompt AI to write a function that checks whether a given year is a leap year. <p>Expected Output#1</p> <ul style="list-style-type: none"> AI-generated function with no examples provided <p>PROMPT:</p> <p>Write a Python program that defines a function to check whether a given year is a leap year. The program should take input from the user, call the function, and print whether the entered year is a leap year or not.(ZERO SHOT EXAMPLE)</p>

[1]
✓ 6s

```
def is_leap_year(year):  
    """  
    Checks if a given year is a leap year.  
  
    A year is a leap year if:  
    - It is divisible by 4, but not by 100  
    - OR it is divisible by 400  
    """  
    if (year % 4 == 0 and year % 100 != 0) or (year % 400 == 0):  
        return True  
    else:  
        return False  
  
    # Take input from the user  
    try:  
        year_input = int(input("Enter a year: "))  
  
        # Call the function and print the result  
        if is_leap_year(year_input):  
            print(f"{year_input} is a leap year.")  
        else:  
            print(f"{year_input} is not a leap year.")  
    except ValueError:  
        print("Invalid input. Please enter a valid year (a whole number).")
```

```
Enter a year: 2020  
2020 is a leap year.
```

Q2.

Task Description#2

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

Expected Output#2

- Function with correct conversion logic

PROMPT:

write a Python program that defines a function to convert centimeters to inches. The program should take input from the user in centimeters, call the function, and print the converted value in inches.

Example: Input: 10
Output: 3.937 inches

```
[2]
✓ 5s def cm_to_inches(cm):
      """
      Converts a value from centimeters to inches.

      1 inch = 2.54 centimeters
      """
      inches = cm * 0.393701 # Approximately 1 inch = 2.54 cm, so 1 cm = 0.393701 inches
      return inches

# Take input from the user
try:
    cm_input = float(input("Enter a value in centimeters: "))

    # Call the function and print the result
    converted_inches = cm_to_inches(cm_input)
    print(f"{cm_input} centimeters is equal to {converted_inches:.3f} inches.")
except ValueError:
    print("Invalid input. Please enter a valid number for centimeters.")
```

Enter a value in centimeters: 100
100.0 centimeters is equal to 39.370 inches.

Q3.

Task Description#3

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

Expected Output#3

- Well-structured function respecting the examples

PROMPT

Write a Python program that defines a function to format a full name as “Last, First”. The program should take a full name as input from the user, call the function, and print the formatted name.

Examples: Input: "Rimsha Mujeeb" Output: "Mujeeb, Rimsha"

Input: "Priya Raj" Output: "Raj, Priya"

Input: "Pujitha sham" Output: "sham, Pujitha"

Q4.

Task Description#4

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

Expected Output#4

- Functional output and comparative reflection

PROMPT:

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

Expected Output#4

- Functional output and comparative reflection

ZERO SHOT PROMPT:

Write a Python function that counts the number of vowels (a, e, i, o, u) in a given string. The program should take input from the user and print the number of vowels.

[4]
✓ 13s

```
def count_vowels(input_string):  
    """  
    Counts the number of vowels (a, e, i, o, u) in a given string.  
    The count is case-insensitive.  
    """  
  
    vowels = "aeiou"  
    vowel_count = 0  
    for char in input_string:  
        if char.lower() in vowels:  
            vowel_count += 1  
    return vowel_count  
  
# Take input from the user  
user_string = input("Enter a string: ")  
  
# Call the function and print the result  
num_vowels = count_vowels(user_string)  
print(f"The number of vowels in the string is: {num_vowels}")
```

```
Enter a string: Rimsha  
The number of vowels in the string is: 2
```

FEW SHOT PROMPT:

write a Python function that counts the number of vowels in a string. The program should take input from the user and print the number of vowels.

Examples: Input: "hello" Output: 2

Input: "World" Output: 1

Input: "education" Output: 5

```
[5]
✓ 14s
def count_vowels(input_string):
    """
    Counts the number of vowels (a, e, i, o, u) in a given string.
    The count is case-insensitive.
    """
    vowels = "aeiou"
    vowel_count = 0
    for char in input_string:
        if char.lower() in vowels:
            vowel_count += 1
    return vowel_count

# Take input from the user
user_string = input("Enter a string: ")

# Call the function and print the result
num_vowels = count_vowels(user_string)
print(f"The number of vowels in the string is: {num_vowels}")

Enter a string: Accept
The number of vowels in the string is: 2
```

COMPARITIVE REFLECTION BETWEEN ZERO SHOT AND FEW SHOT PROMPT AND CODE .

ZERO SHOT :

1. The AI **receives no examples**, so it relies purely on **its general understanding** of what vowels are.
2. The function works correctly but may not be optimized or styled consistently (e.g., using loops instead of comprehensions).
3. The logic is fully correct but **lacks contextual guidance** — it infers what's expected.

FEW SHOT :

1. The **examples** clarify the expected pattern — input is a word or sentence, output is an integer count.
2. The AI generalizes from examples and writes **more concise and Pythonic code** (using list comprehension and sum()).
3. The result is **cleaner, efficient, and stylistically aligned** with the examples provided.

Q5.

Task Description#5

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

Expected Output#5

- Working file-processing function with AI-guided logic

PROMPT:

Write a Python program that defines a function to read a .txt file and return the number of lines it contains. The program should take the file name as input from the user, call the function, and print the total number of lines.

FILE NAME: "example.txt"

Hello

world

Line one

Line two

Line three

[6]
✓ 17s

```
def count_lines_in_file(file_path):  
    """  
    Reads a text file and returns the number of lines it contains.  
    Handles FileNotFoundError if the file does not exist.  
    """  
    try:  
        with open(file_path, 'r') as file:  
            lines = file.readlines()  
            return len(lines)  
    except FileNotFoundError:  
        return -1 # Indicate that the file was not found  
    except Exception as e:  
        print(f"An error occurred: {e}")  
        return -2 # Indicate other errors  
  
# Take file name input from the user  
file_name = input("Enter the name of the .txt file (e.g., my_document.txt): ")  
  
# Call the function and print the result  
line_count = count_lines_in_file(file_name)  
  
if line_count == -1:  
    print(f"Error: The file '{file_name}' was not found.")  
elif line_count == -2:  
    print(f"An unexpected error occurred while reading '{file_name}'.")  
elif line_count is not None:  
    print(f"The file '{file_name}' contains {line_count} lines.")
```

Enter the name of the .txt file (e.g., my_document.txt): example.txt
The file 'example.txt' contains 6 lines.