ASSIGNMENT-4.3

M.HARIKRISHNA

2203A52100

B-50

Task 1

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.

Expected Output

• AI-generated leap year checking function

• Correct logical conditions

• Sample input and output

• Screenshot of AI-generated response (if required)

Prompt:

Generate a Python function that accepts a year as input, checks whether it is a leap year, and returns the result. Do not use examples.

Code:

```
def check_leap_year(year):     if (year % 4 == 0 and year %
100 != 0) or (year % 400 == 0):
        return "Leap Year"
else:
        return "Not a Leap Year"
```

```
year = int(input("Enter year: "))
print(check_leap_year(year))
```

Output:

Code Explanation:

This program checks whether a year is a leap year using standard leap year rules. A year must be divisible by 4 but not by 100 unless divisible by 400. Zero-shot prompting is used because no examples are given to the AI. The AI generates logic based only on instructions. This method works well for simple logical problems.

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

Expected Output

• Python function with correct conversion logic

• Accurate calculation

• Sample test cases and outputs

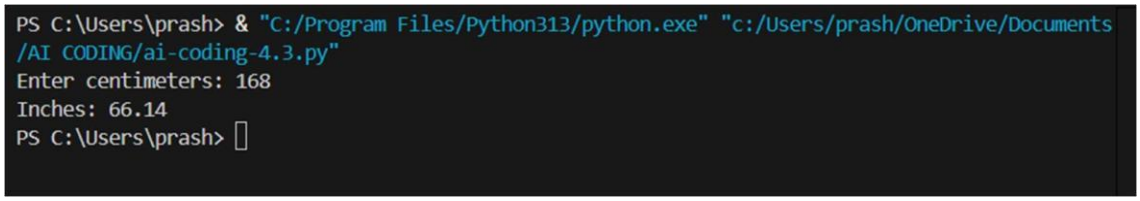
Prompt:

Generate Python function to convert centimeters to inches.
Example: 10 cm → 3.94 inches

Code:

```
def cm_to_inches(cm):
    return cm / 2.54


cm = float(input("Enter centimeters: "))
print("Inches:", round(cm_to_inches(cm), 2))
```

Output:



Code Explanation:

This program converts centimeters to inches using the formula Inches = CM / 2.54. One-shot prompting helps AI understand correct conversion using one example. This improves accuracy compared to zero-shot prompting.

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"

Expected Output

• Well-structured Python function

• Output strictly following example patterns

• Correct handling of names

• Sample inputs and outputs

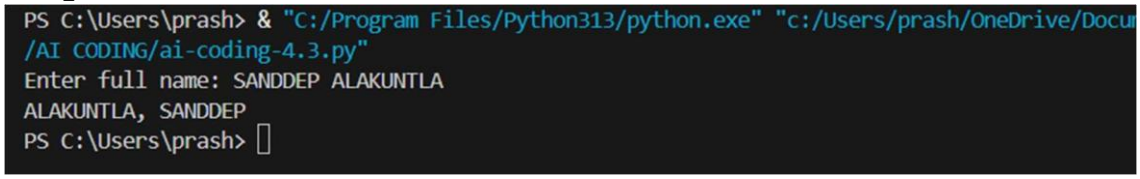
Prompt:

Generate Python function to format name as "Last, First". Examples: John Smith → Smith, John Anita Rao → Rao, Anita

Code:

```
def

format_name(full_name):

parts = full_name.split()

first = parts[0]     last =

parts[-1]     return f"{last},

{first}"


name = input("Enter full name: ")

print(format_name(name))
```

Output:



Code Explanation:

Few-shot prompting uses multiple examples, helping AI generate accurate formatting logic. The program splits the name and rearranges it into the required format.

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

Expected Output

• Two vowel-counting functions

• Comparison table or short reflection paragraph

• Conclusion on prompt effectiveness

Prompt:

Generate Python function to count vowels in a string.

Code:

```
def

count_vowels_zero(text):

vowels = "aeiouAEIOU"

count = 0 for ch in text:

if ch in vowels:

count += 1 return count
```

Code explanation:

Generate Python vowel count
function. Examples: Hello → 2
India → 3 Few-shot promp ng produces cleaner and more op mized code because examples guide AI logic.

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

Expected Output

• Working Python file-processing function

• Correct line count

• Sample .txt input and output

• AI-assisted logic explanation

Note: Report should be submitted as a word document for all tasks

in a single document with prompts, comments & code explanation,

and output and if required, screenshots

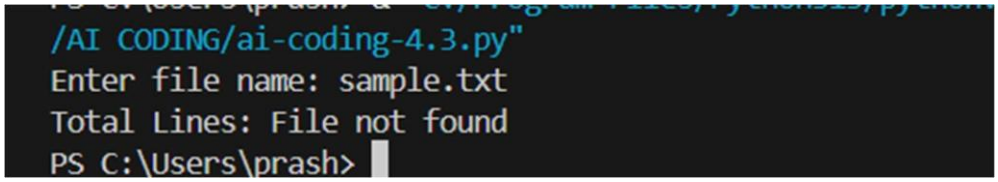
Prompt:

G enerate Python function to read .txt file and count lines.
Example: sample.txt → 4 lines

Code:

```
def count_lines(filename):
    try:          with open(filename, "r")
as   file:                                  return
len(file.readlines())      except:
        return "File not found"


file_name = input("Enter file name: ")
print("Total Lines:", count_lines(file_name))
```

Output:



Code explanation:

Few-shot prompting helps AI understand file reading and counting logic using examples. The program safely opens the file and counts total lines.