

ASSIGNMENT-4.3

Name: J.Vyshnavi

HT. No: 2303A51895

Batch: 08

Lab 4: Advanced Prompt Engineering – Zero-shot, One-shot, and Few-shot Techniques

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.

```
1 #Task 1: Zero-Shot Prompting - Leap Year Check
2 def is_leap_year(year):
3     if (year % 400 == 0) or (year % 4 == 0 and year % 100 != 0):
4         return "Leap Year"
5     else:
6         return "Not a Leap Year"
7 year = int(input("Enter a year: "))
8 print(is_leap_year(year))
```

```
Enter a year: 2004
```

```
Leap Year
```

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

```
1 #Task 2: One-Shot Prompting - Centimeters to Inches
2 def cm_to_inches(cm):
3     return cm / 2.54
4 cm = float(input("Enter length in centimeters: "))
5 print("Length in inches:", round(cm_to_inches(cm), 2))
```

```
Enter length in centimeters: 150
Length in inches: 59.06
```

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”

```
1 #Task 3: Few-Shot Prompting - Name Formatting
2 def format_name(full_name):
3     parts = full_name.split()
4     first_name = parts[0]
5     last_name = parts[-1]
6     return last_name + ", " + first_name
7 name = input("Enter full name: ")
8 print(format_name(name))
```

```
Enter full name: john nick
nick, john
```

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:
 - Accuracy
 - Readability
 - Logical clarity

```

1 #Task 4: Comparative Analysis – Zero-Shot vs Few-Shot (Vowel Count)
2 def count_vowels_zero_shot(text):
3     count = 0
4     for ch in text.lower():
5         if ch in "aeiou":
6             count += 1
7     return count
8 text = input("Enter a string: ")
9 print("Vowel count:", count_vowels_zero_shot(text))
10 def count_vowels_few_shot(text):
11     vowels = "aeiouAEIOU"
12     return sum(1 for ch in text if ch in vowels)
13 text = input("Enter a string: ")
14 print("Vowel count:", count_vowels_few_shot(text))

```

```

Enter a string: I am a very good girl
Vowel count: 7
Enter a string: hello
Vowel count: 2

```

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

```

1 #Task 5: Few-Shot Prompting – File Handling (Line Count)
2 def count_lines(filename):
3     with open(filename, "r") as file:
4         return len(file.readlines())
5 with open("sample.txt", "w") as file:
6     file.write("Hello\nWelcome to AI Assisted Coding\nThis is Lab 4\n")
7 print("Number of lines:", count_lines("sample.txt"))

```

```

Number of lines: 3

```