

# Assingment-4.3

## *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)*

The screenshot shows a Python IDE with a file named `leapyercheck.py` open. The script defines a function `is_leap_year` that checks if a year is a leap year based on the rules: divisible by 4 but not 100, or divisible by 400. The script prompts the user to enter a year and prints the result.

```

1 #write a code that accepts a year and to check if a year is a leap year or not user input function
2 def is_leap_year(year):
3     if (year % 4 == 0 and year % 100 != 0) or (year % 400 == 0):
4         return True
5     else:
6         return False
7 year = int(input("Enter a year: "))
8 if is_leap_year(year):
9     print(f"{year} is a leap year.")
10 else:
11     print(f"{year} is not a leap year.")
12
13

```

The terminal output shows the script being executed in a PowerShell window. The user enters `2024`, and the program outputs `2024 is a leap year.`

```

PS C:\Users\shahr\OneDrive\Desktop\AI> & C:\Users\shahr\AppData\Local\Programs\Python\Python313\python.exe c:/Users/shahr/OneDrive/Desktop/AI/leapyercheck.py
Enter a year: 2024
2024 is a leap year.
PS C:\Users\shahr\OneDrive\Desktop\AI>

```

### 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*

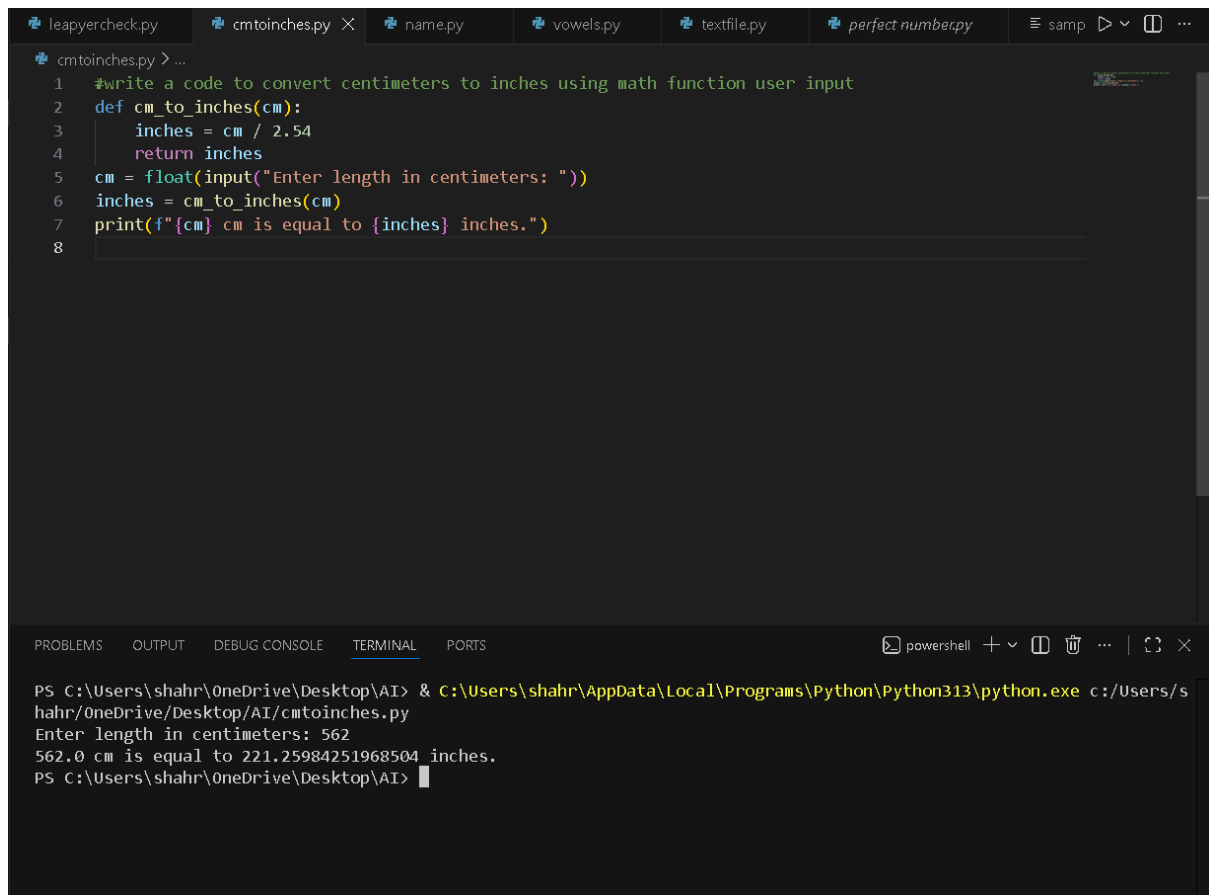
*Example provided in prompt:*

*Input: 10 cm → Output: 3.94 inches*

### Expected Output

- *Python function with correct conversion logic*
- *Accurate calculation*

- *Sample test cases and outputs*



The screenshot shows a Python IDE with several tabs. The active tab is `cmttoinches.py`, which contains the following code:

```
1 #write a code to convert centimeters to inches using math function user input
2 def cm_to_inches(cm):
3     inches = cm / 2.54
4     return inches
5 cm = float(input("Enter length in centimeters: "))
6 inches = cm_to_inches(cm)
7 print(f"{cm} cm is equal to {inches} inches.")
8
```

The terminal at the bottom shows the execution of the program:

```
PS C:\Users\shahr\OneDrive\Desktop\AI> & C:\Users\shahr\AppData\Local\Programs\Python\Python313\python.exe c:/Users/s
hahr/OneDrive/Desktop/AI/cmttoinches.py
Enter length in centimeters: 562
562.0 cm is equal to 221.25984251968504 inches.
PS C:\Users\shahr\OneDrive\Desktop\AI>
```

### *Task 3: Few-Shot Prompting – Name Formatting*

#### *Scenario*

*Few-shot prompting improves accuracy by providing multiple examples.*

#### *Task Description*

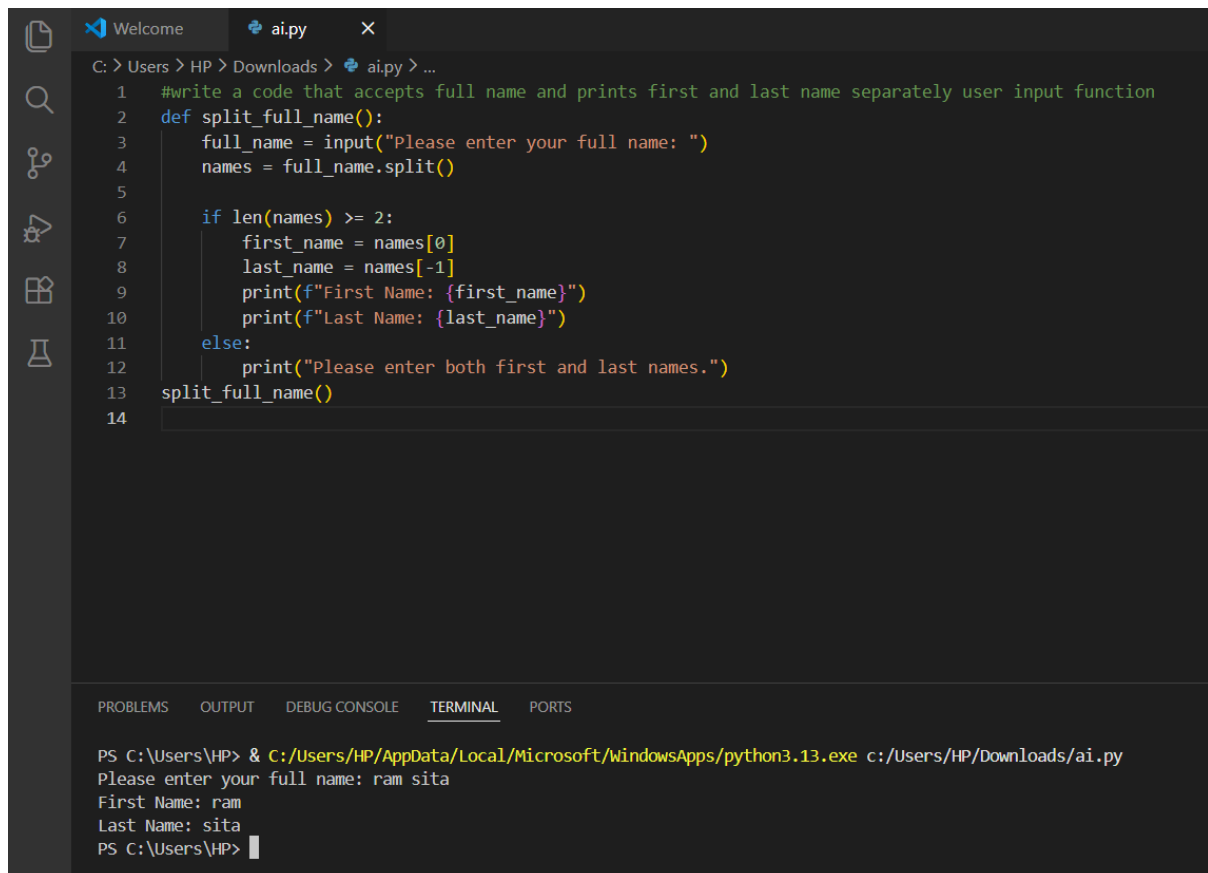
*Use few-shot prompting with 2–3 examples to generate a Python function that:*

- *Accepts a full name as input*
- *Formats it as “Last, First”*

*Example formats:*

- *"John Smith" → "Smith, John"*
- *"Anita Rao" → "Rao, Anita"*

## Expected Output



The screenshot shows a Visual Studio Code editor with a file named `ai.py` open. The code is a Python function `split_full_name()` that takes a full name as input and prints the first and last names separately. The function uses `input()` to get the full name, splits it into a list, and then checks if the list has at least two elements. If so, it prints the first and last names; otherwise, it prompts the user to enter both first and last names.

```
1 #write a code that accepts full name and prints first and last name separately user input function
2 def split_full_name():
3     full_name = input("Please enter your full name: ")
4     names = full_name.split()
5
6     if len(names) >= 2:
7         first_name = names[0]
8         last_name = names[-1]
9         print(f"First Name: {first_name}")
10        print(f>Last Name: {last_name}")
11    else:
12        print("Please enter both first and last names.")
13    split_full_name()
14
```

The terminal output shows the execution of the script. The prompt `Please enter your full name:`  is followed by the input `ram sita`. The output then shows `First Name: ram` and `Last Name: sita`.

```
PS C:\Users\HP> & C:/Users/HP/AppData/Local/Microsoft/WindowsApps/python3.13.exe c:/Users/HP/Downloads/ai.py
Please enter your full name: ram sita
First Name: ram
Last Name: sita
PS C:\Users\HP>
```

- Well-structured Python function
- Output strictly following example patterns
- Correct handling of names
- Sample inputs and outputs

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

### Scenario

Different prompt strategies may produce different code quality.

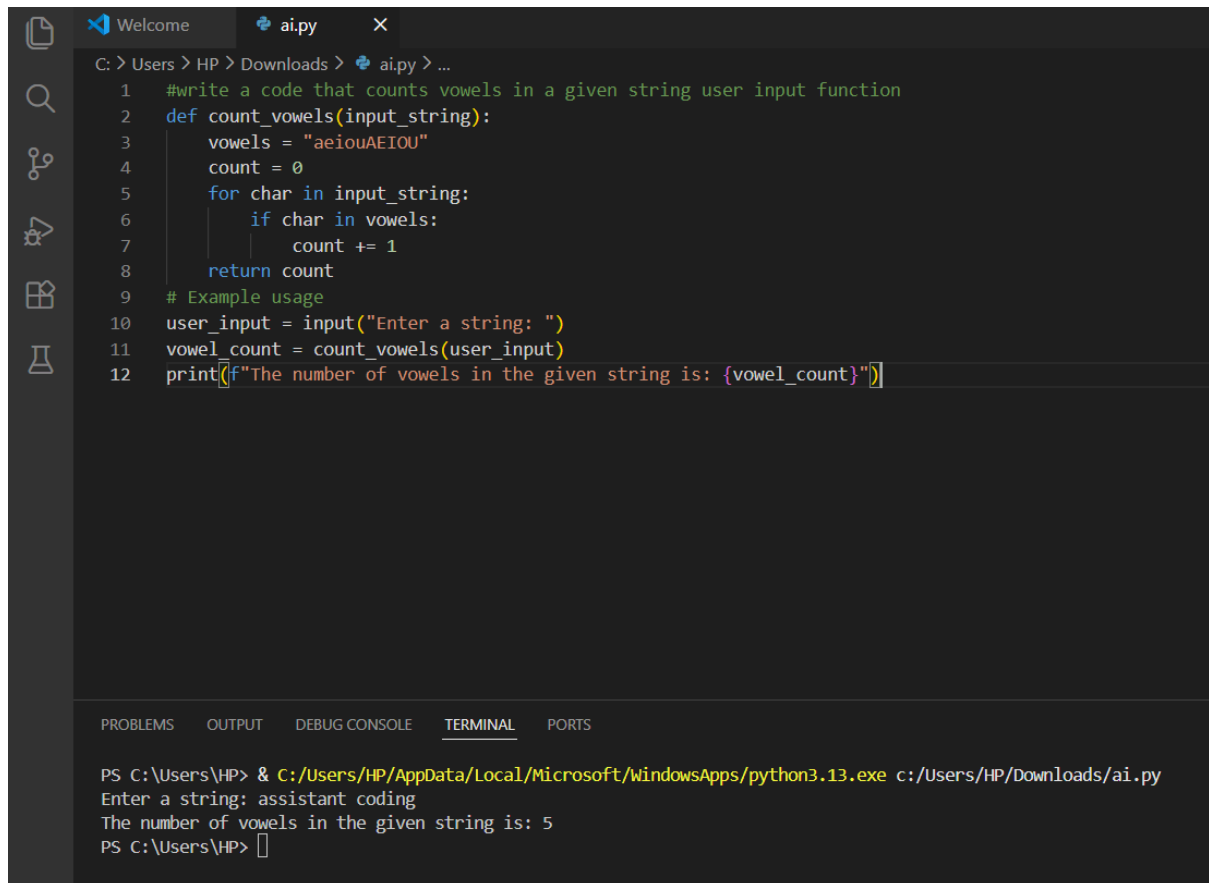
### Task Description

- 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*



The screenshot shows a Visual Studio Code editor window with a file named `ai.py`. The code is a Python script that defines a function `count_vowels` to count the number of vowels in a given string. The script includes a comment, a function definition, and an example usage section. The terminal at the bottom shows the command to run the script and the output, which is the number of vowels in the string "assistant coding".

```
1 #write a code that counts vowels in a given string user input function
2 def count_vowels(input_string):
3     vowels = "aeiouAEIOU"
4     count = 0
5     for char in input_string:
6         if char in vowels:
7             count += 1
8     return count
9 # Example usage
10 user_input = input("Enter a string: ")
11 vowel_count = count_vowels(user_input)
12 print(f"The number of vowels in the given string is: {vowel_count}")
```

Terminal Output:

```
PS C:\Users\HP> & C:/Users/HP/AppData/Local/Microsoft/windowsApps/python3.13.exe c:/Users/HP/Downloads/ai.py
Enter a string: assistant coding
The number of vowels in the given string is: 5
PS C:\Users\HP>
```

## *Task 5: Few-Shot Prompting – File Handling*

*Scenario*

*File processing requires clear logical understanding.*

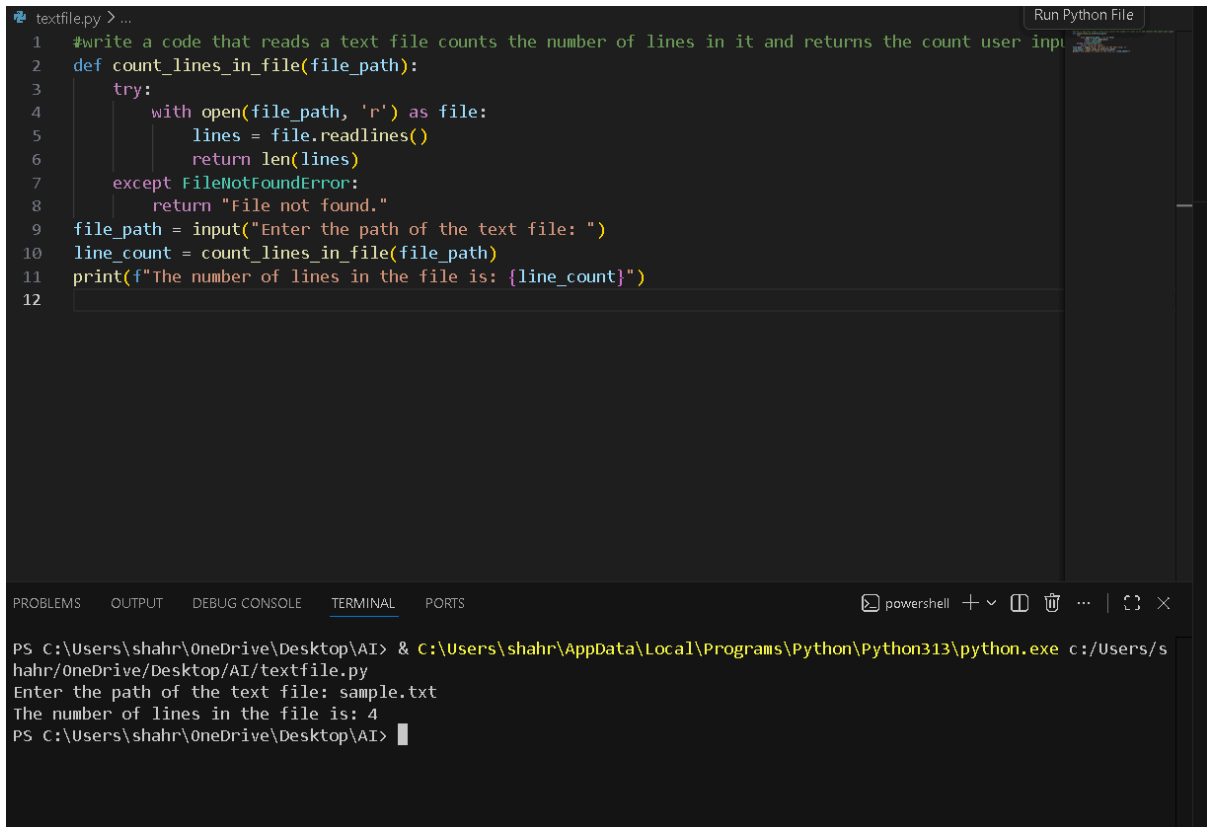
*Task Description*

*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*



The image shows a Python IDE with a file named `textfile.py`. The code defines a function `count_lines_in_file` that takes a file path and returns the number of lines. It uses a `try-except` block to handle `FileNotFoundError`. The main part of the script prompts the user for a file path, calls the function, and prints the result.

```
1 #write a code that reads a text file counts the number of lines in it and returns the count user input
2 def count_lines_in_file(file_path):
3     try:
4         with open(file_path, 'r') as file:
5             lines = file.readlines()
6             return len(lines)
7     except FileNotFoundError:
8         return "File not found."
9 file_path = input("Enter the path of the text file: ")
10 line_count = count_lines_in_file(file_path)
11 print(f"The number of lines in the file is: {line_count}")
12
```

The terminal output shows the execution of the script. The user enters `sample.txt` as the file path, and the program outputs `The number of lines in the file is: 4`.

```
PS C:\Users\shahr\OneDrive\Desktop\AI> & C:\Users\shahr\AppData\Local\Programs\Python\Python313\python.exe c:/Users/s
hahr/OneDrive/Desktop/AI/textfile.py
Enter the path of the text file: sample.txt
The number of lines in the file is: 4
PS C:\Users\shahr\OneDrive\Desktop\AI>
```