BOLAGANI SUSHANTH

2403A52L11

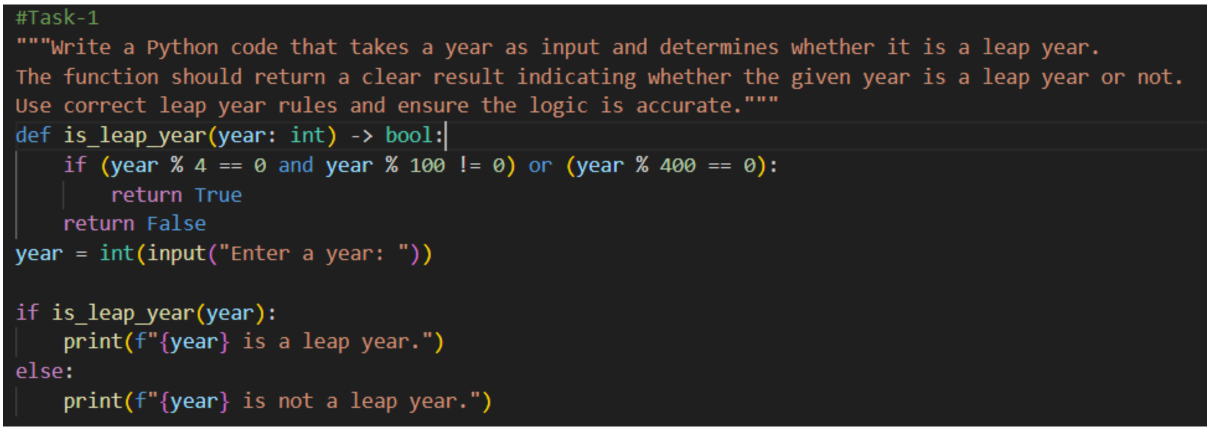
B-50

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

Techniques

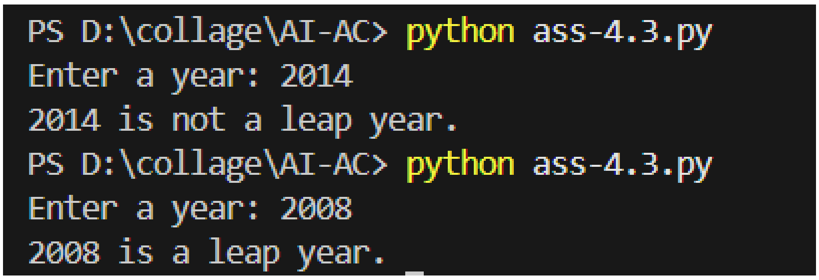
Task 1: Zero-Shot Prompting – Leap Year Check

Prompt and code:



```python
#Task-1
"""Write a Python code that takes a year as input and determines whether it is a leap year.
The function should return a clear result indicating whether the given year is a leap year or not.
Use correct leap year rules and ensure the logic is accurate."""
def is_leap_year(year: int) -> bool:
    if (year % 4 == 0 and year % 100 != 0) or (year % 400 == 0):
        return True
    return False
year = int(input("Enter a year: "))

if is_leap_year(year):
    print(f"{year} is a leap year.")
else:
    print(f"{year} is not a leap year.")
```

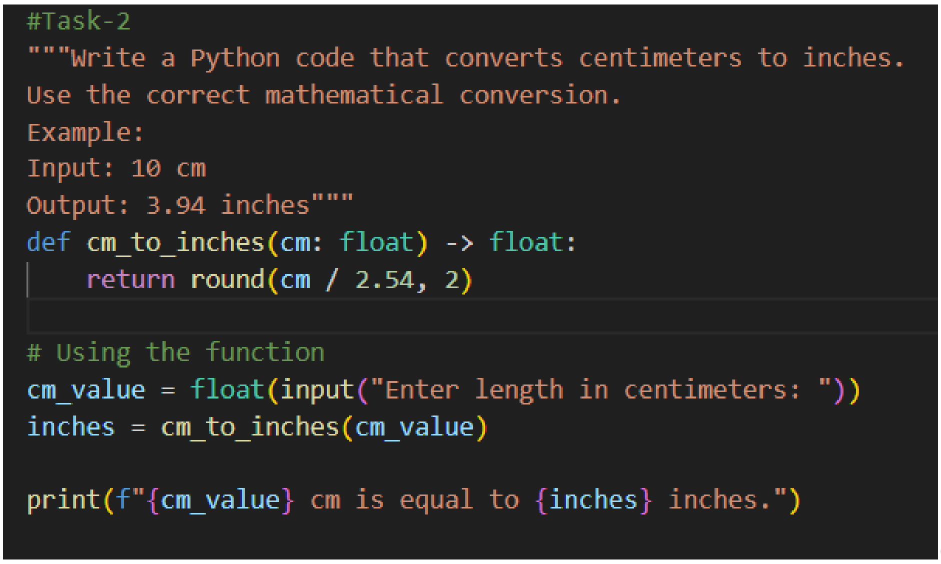
Output:



```
PS D:\collage\AI-AC> python ass-4.3.py
Enter a year: 2014
2014 is not a leap year.
PS D:\collage\AI-AC> python ass-4.3.py
Enter a year: 2008
2008 is a leap year.
```

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

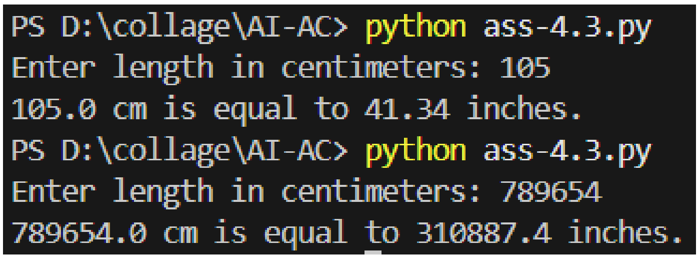
Prompt and code:



```
#Task-2
"""Write a Python code that converts centimeters to inches.
Use the correct mathematical conversion.
Example:
Input: 10 cm
Output: 3.94 inches"""
def cm_to_inches(cm: float) -> float:
    return round(cm / 2.54, 2)

# Using the function
cm_value = float(input("Enter length in centimeters: "))
inches = cm_to_inches(cm_value)

print(f"{cm_value} cm is equal to {inches} inches.")
```

0

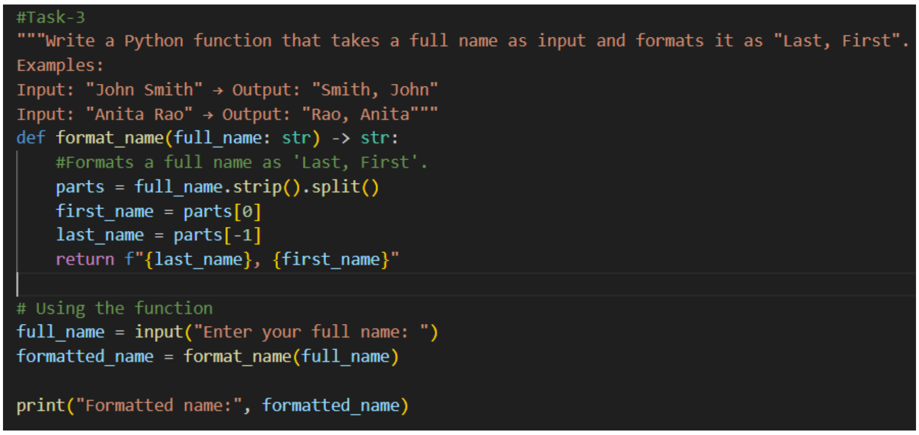
Output:



```
PS D:\collage\AI-AC> python ass-4.3.py
Enter length in centimeters: 105
105.0 cm is equal to 41.34 inches.
PS D:\collage\AI-AC> python ass-4.3.py
Enter length in centimeters: 789654
789654.0 cm is equal to 310887.4 inches.
```

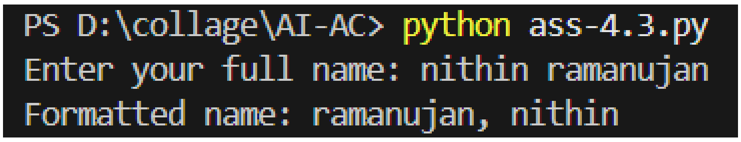
Task-3: Few-Shot Prompting – Name Formatting

Prompt and code:



```
#Task-3
"""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"""
def format_name(full_name: str) -> str:
    #Formats a full name as 'Last, First'.
    parts = full_name.strip().split()
    first_name = parts[0]
    last_name = parts[-1]
    return f"{last_name}, {first_name}"

# Using the function
full_name = input("Enter your full name: ")
formatted_name = format_name(full_name)

print("Formatted name:", formatted_name)
```

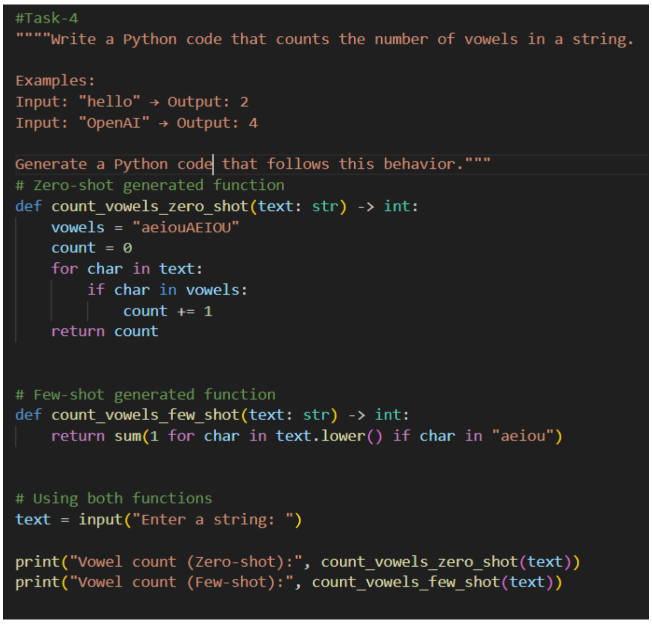
Output:



```
PS D:\collage\AI-AC> python ass-4.3.py
Enter your full name: nithin ramanujan
Formatted name: ramanujan, nithin
```

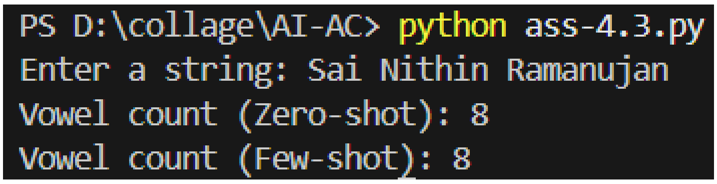
Task-4: Comparative Analysis – Zero-Shot vs Few-Shot

Prompt and code:



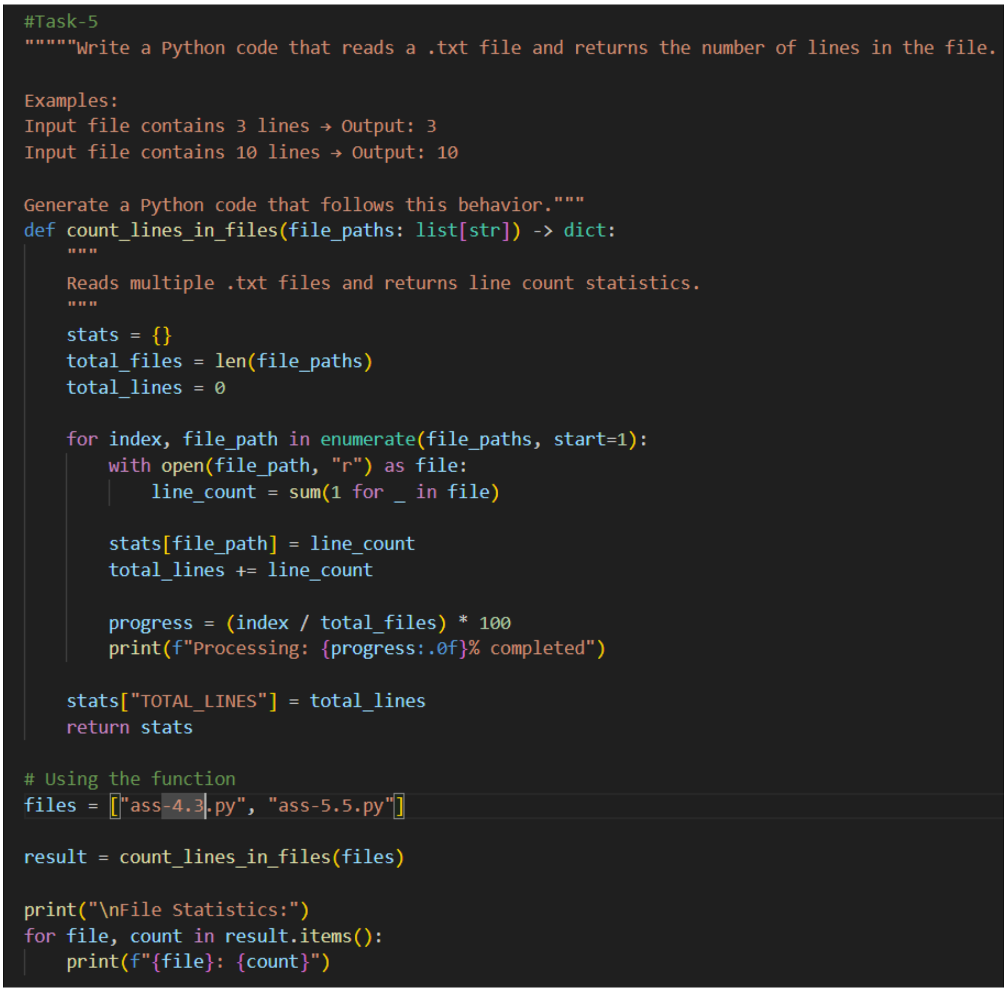
```python
#Task-4
""""Write a Python code that counts the number of vowels in a string.

Examples:
Input: "hello" → Output: 2
Input: "OpenAI" → Output: 4

Generate a Python code that follows this behavior."""
# Zero-shot generated function
def count_vowels_zero_shot(text: str) -> int:
    vowels = "aeiouAEIOU"
    count = 0
    for char in text:
        if char in vowels:
            count += 1
    return count


# Few-shot generated function
def count_vowels_few_shot(text: str) -> int:
    return sum(1 for char in text.lower() if char in "aeiou")


# Using both functions
text = input("Enter a string: ")

print("Vowel count (Zero-shot):", count_vowels_zero_shot(text))
print("Vowel count (Few-shot):", count_vowels_few_shot(text))
```

Output:



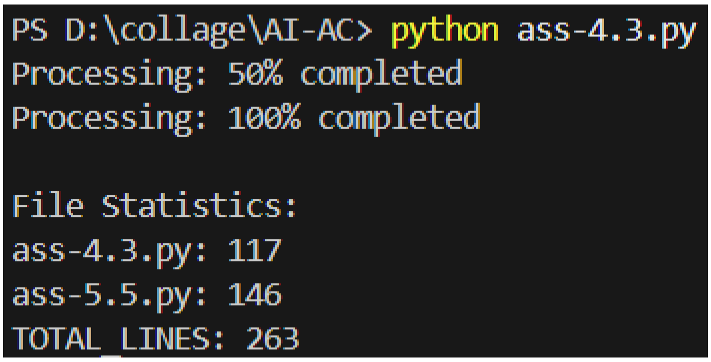
```
PS D:\collage\AI-AC> python ass-4.3.py
Enter a string: Sai Nithin Ramanujan
Vowel count (Zero-shot): 8
Vowel count (Few-shot): 8
```

Task-5: Few-Shot Prompting – File Handling



```
#Task-5
""""Write a Python code that reads a .txt file and returns the number of lines in the file.

Examples:
Input file contains 3 lines → Output: 3
Input file contains 10 lines → Output: 10

Generate a Python code that follows this behavior."""
def count_lines_in_files(file_paths: list[str]) -> dict:
    """
    Reads multiple .txt files and returns line count statistics.
    """
    stats = {}
    total_files = len(file_paths)
    total_lines = 0

    for index, file_path in enumerate(file_paths, start=1):
        with open(file_path, "r") as file:
            line_count = sum(1 for _ in file)

        stats[file_path] = line_count
        total_lines += line_count

        progress = (index / total_files) * 100
        print(f"Processing: {progress:.0f}% completed")

    stats["TOTAL_LINES"] = total_lines
    return stats

# Using the function
files = ["ass-4.3.py", "ass-5.5.py"]

result = count_lines_in_files(files)

print("\nFile Statistics:")
for file, count in result.items():
    print(f"{file}: {count}")
```

Output:



```
PS D:\collage\AI-AC> python ass-4.3.py
Processing: 50% completed
Processing: 100% completed

File Statistics:
ass-4.3.py: 117
ass-5.5.py: 146
TOTAL_LINES: 263
```