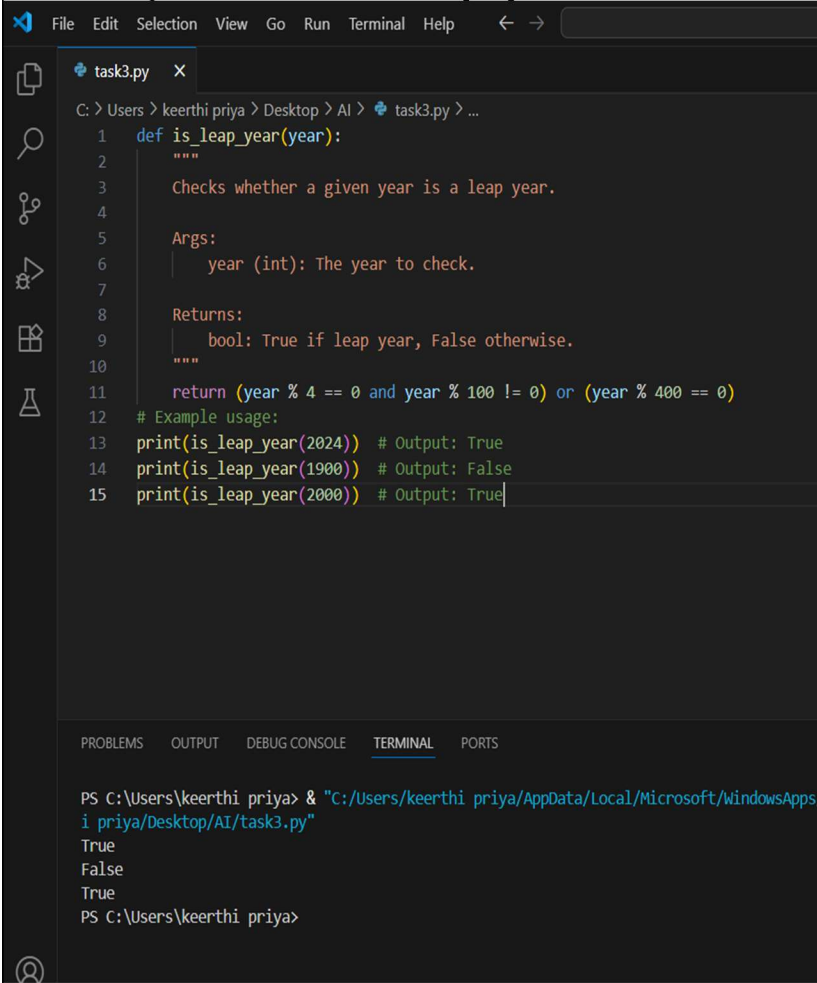



	<div>AI ASSISTED CODING</div> <div>NAME : J.KEERTHI PRIYA</div> <div>ROLL NO:2403A510G4</div> <div>ASSIGNMENT:4.3</div>	
	<div><div>Task Description#1</div><div><ul style="list-style-type: none"><li>Zero-shot: Prompt AI to write a function that checks whether a given year is a leap year.</li></ul></div><div><div>Expected Output#1</div><div><ul style="list-style-type: none"><li>AI-generated function with no examples provided</li></ul></div></div><div><pre>File Edit Selection View Go Run Terminal Help task3.py x C: &gt; Users &gt; keerthi priya &gt; Desktop &gt; AI &gt; task3.py &gt; ... 1 def is_leap_year(year): 2     """ 3     Checks whether a given year is a leap year. 4 5     Args: 6     year (int): The year to check. 7 8     Returns: 9     bool: True if leap year, False otherwise. 10    """ 11    return (year % 4 == 0 and year % 100 != 0) or (year % 400 == 0) 12 # Example usage: 13 print(is_leap_year(2024)) # Output: True 14 print(is_leap_year(1900)) # Output: False 15 print(is_leap_year(2000)) # Output: True  PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS PS C:\Users\keerthi priya&gt; &amp; "C:/Users/keerthi priya/AppData/Local/Microsoft/WindowsApps/ i priya/Desktop/AI/task3.py" True False True PS C:\Users\keerthi priya&gt;</pre></div><div><div>Task Description#2</div><div><ul style="list-style-type: none"><li>One-shot: Give one input-output example to guide AI in writing a function that converts centimeters to inches.</li></ul></div><div><div>Expected Output#2</div><div><ul style="list-style-type: none"><li>Function with correct conversion logic</li></ul></div></div></div></div>	

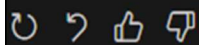


writing a function that converts centimeters to inches. ONLY THIS CODE

 task3.py

> Used 1 reference

```
def cm_to_inches(cm):  
    """  
    Converts centimeters to inches.  
  
    Args:  
        cm (float): Length in centimeters.  
  
    Returns:  
        float: Length in inches.  
    """  
    return cm / 2.54
```



### 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

🔖 Restore Checkpoint

generate a PYTHON function that formats full names as "Last, First".

📎 task3.py

> Used 1 reference

```
def format_full_name(first_name, last_name):  
    """  
    Formats full names as 'Last, First'.  
  
    Args:  
        first_name (str): First name.  
        last_name (str): Last name.  
  
    Returns:  
        str: Formatted name.  
    """  
    return f"{last_name}, {first_name}"
```

#### 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



Restore Checkpoint

write a function that counts the number of vowels in a string.

 task3.py

> Used 1 reference

```
def count_vowels(s):  
    """  
    Counts the number of vowels in a string.  
  
    Args:  
        s (str): Input string.  
  
    Returns:  
        int: Number of vowels in the string.  
    """  
    vowels = 'aeiouAEIOU'  
    return sum(1 for char in s if char in vowels)
```

#### 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



generate a function that reads a .txt file and returns the number of lines.

 task3.py

> Used 1 reference

```
def count_lines_in_file(filename):  
    """  
    Reads a .txt file and returns the number of lines  
  
    Args:  
        filename (str): Path to the .txt file.  
  
    Returns:  
        int: Number of lines in the file.  
    """  
    with open(filename, 'r', encoding='utf-8') as fil  
        return sum(1 for _ in file)
```

**Note:** Report should be submitted a word document for all tasks in a single document with prompts, comments & code explanation, and output and if required, screenshots

#### Evaluation Criteria:

Criteria	Max Marks
Zero Shot (Task #1)	0.5
One Shot (Task#2)	0.5
Few Shot (Task#3 & Task #5)	1.0
Comparison (Task#4)	0.5
<b>Total</b>	<b>2.5 Marks</b>