AIAssistedCoding

Assignment–4.3

Name :

M.Anu Chandra

RollNo:2303A54070

Batch:48

Lab4:AdvancedPromptEngineering–Zero-shot,One-shot,andFew-shot Techniques

Task1:Zero-ShotPrompting–LeapYearCheck Scenario: Zero-shotpromptinginvolvesgivinginstructionswithoutproviding examples.

Promptused:#WriteaPythonfunctionthattakesayearasinput andcheckswhether it isaleap year. #Thefunctionshouldreturnanappropriatemessageindicatingwhetherthe year isaleapyearor not. #Donot includeanyinput-output examples.

- **Generatedcode:**

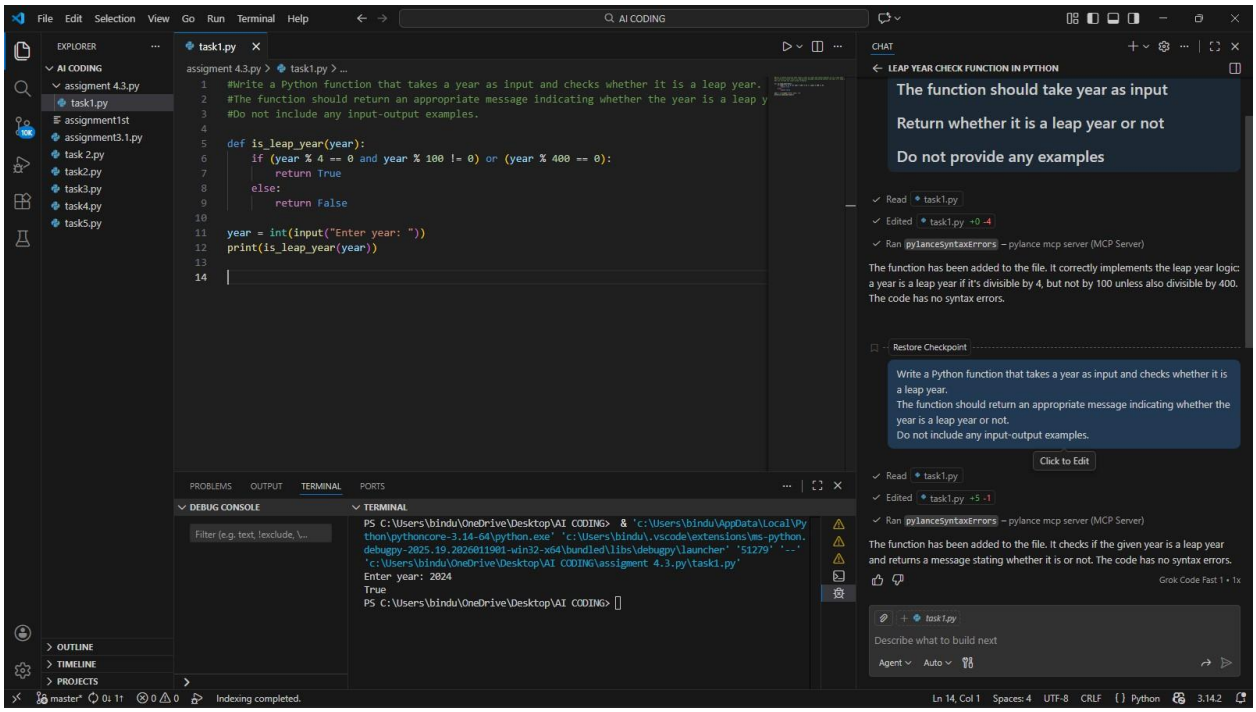
```
defis_leap_year(year):
    if(year%4==0andyear%100!=0)or(year%400==0):
        returnTrue
    else:
        returnFalse

year=int(input("Enteryear:")) print(is_leap_year(year)
```

- **SampleInput:**
  Enteryear:2024
- **SampleOutput:**

True

- **ShortExplanationofLogic:**

theprogramchecksthe leap yearconditionsusinglogicaloperators.ayear isa leap year if the it is divisible by 400 or divisibe by 4 but not divisible by100. the logic is implemented directly without providing any prior .



Task2:One-ShotPrompting –CentimeterstoInchesConversion

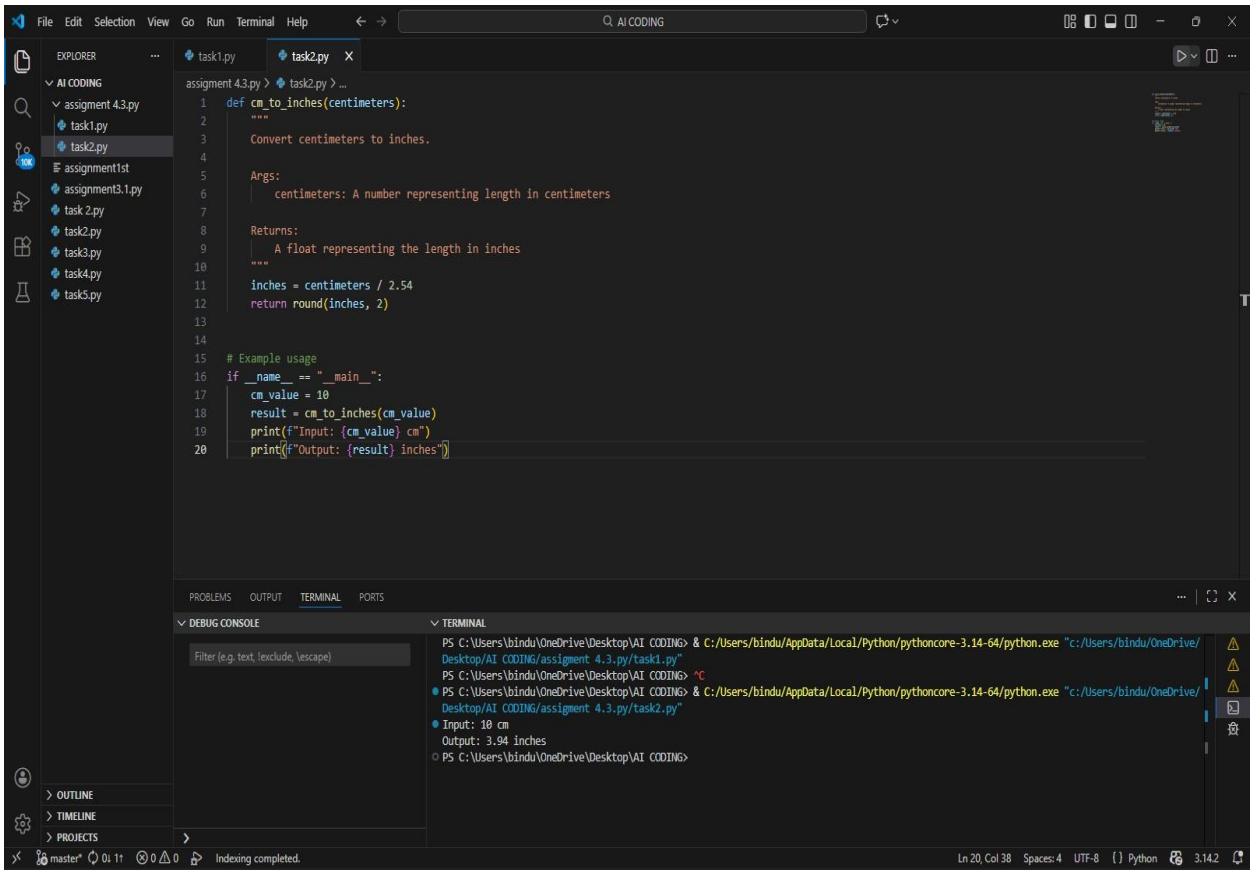
❖ **Scenario:One-shotpromptingguidesAIusingasingle example.**

- **Promptused:**
- #WriteaPythonfunctiontoconvertcentimetersto inches
- #Usetheformulainches=centimeters/2.54
- #Example:
- #Input:10cm
- # Output:3.94inches
- **Generatedcode:**

```
defcm_to_inches(centimeters):
    """
    Convertcentimeterstoinches.

    Args:
        centimeters:Anumberrepresentinglengthincentimeters

    Returns:
        Afloatrepresentingthelengthin inches
    """
    inches=centimeters/2.54
    returnround(inches, 2)


#Exampleusage
ifname=="main": cm_value=10
    result=cm_to_inches(cm_value)
    print(f"Input:{cm_value}cm")
    print(f"Output:{result}inches")
```

- **SampleInput:**
- Entervalueincm:25.4
- **SampleOutput:**
  10.0inches
  ShortExplanationofLogic:

The functiontakes centimeters as input and converts it into inches using the standard formula Cm%2.54oneexample intheprompt helpstheaitogeneratethe logicaccuratelyinoneshort



Task3:Few-shotpromting-NameFormatting

Scenario:

Few-shotpromptingimprovesaccuracybyprovidingmultipleexamples.

❖

- **Promptused:**
- #WriteaPythonfunctiontoformatnamesas"Last,First"
- #Examples:
- #"JohnSmith"->"Smith,John"
- # "Anita Rao" ->"Rao, Anita"

- **Generatedcode:**

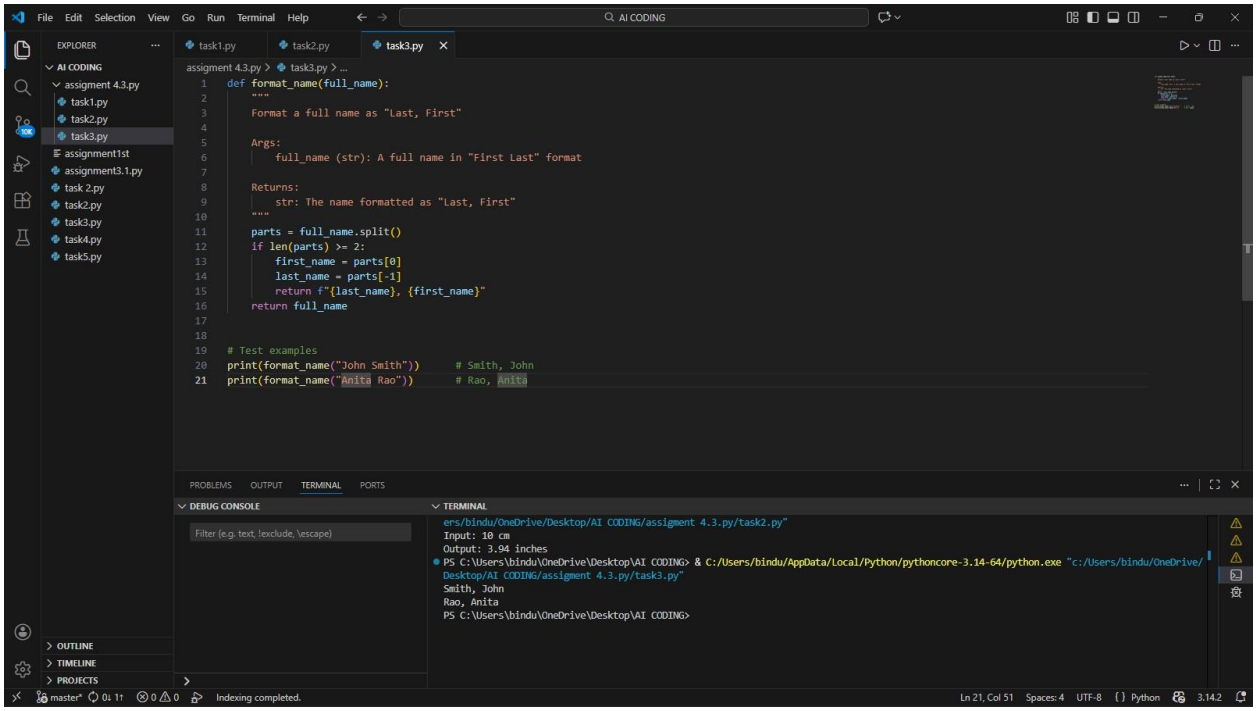
```
"""


defformat_name(full_name):
    """
    Formatafullnameas"Last,First"

    Args:
        full_name(str):Afullnamein"FirstLast"format

    Returns:
        str:Thenameformattedas"Last, First"
    """
    parts=full_name.split()
    iflen(parts) >=2:
        first_name=parts[0]
        last_name=parts[-1]
        returnf"{last_name},{first_name}"
    returnfull_name


#Testexamples
print(format_name("JohnSmith"))        #Smith, John
print(format_name("AnitaRao"))         #Rao,Anita
```

- **SampleInput:**

Enterfullname:JohnSmith

- **SampleOutput:Smith,John**

ShortExplanationofLogic: The functionsplitsthefullname into first and last namesand rearrangesthemintherequiredformat itprovidingmultipleexamplesimprovesfewshort



Task4:ComparativeAnalysis–Zero-ShotvsFew-Shot

- **Scenario: Differentpromptstrategiesmayproducedifferentcodequality.**
- **Prompt1:Zero-ShotPrompting**

WriteaPythonfunctionthat countsthenumberofvowelsinagivenstring. The function should return the total count.
Donotprovideanyexamples.

Generatedcode:

```
def count_vowels(string):
    vowels="aeiouAEIOU" count
    = 0
    forcharinstring:
        ifcharin vowels:
```

- **SampleInput:**

Enterstring:HelloWorld

- **SampleOutput:**

Numberofvowels:3

- **Prompt2:Few-ShotPrompting**

  - WriteaPythonfunctiontocount vowelsinastring.
  - Examples:
  - Input:"hello"→Output:2
  - Input:"AI AssistedCoding"→Output:7

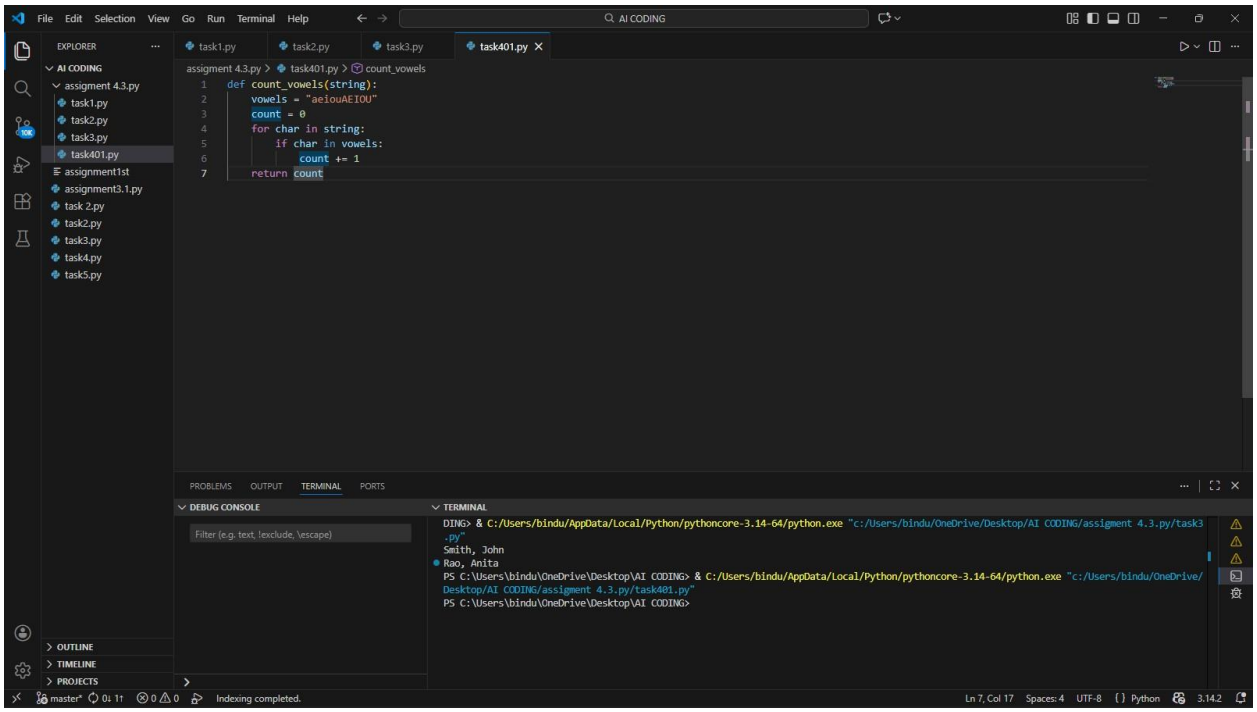
❖ **Generatedcode:**

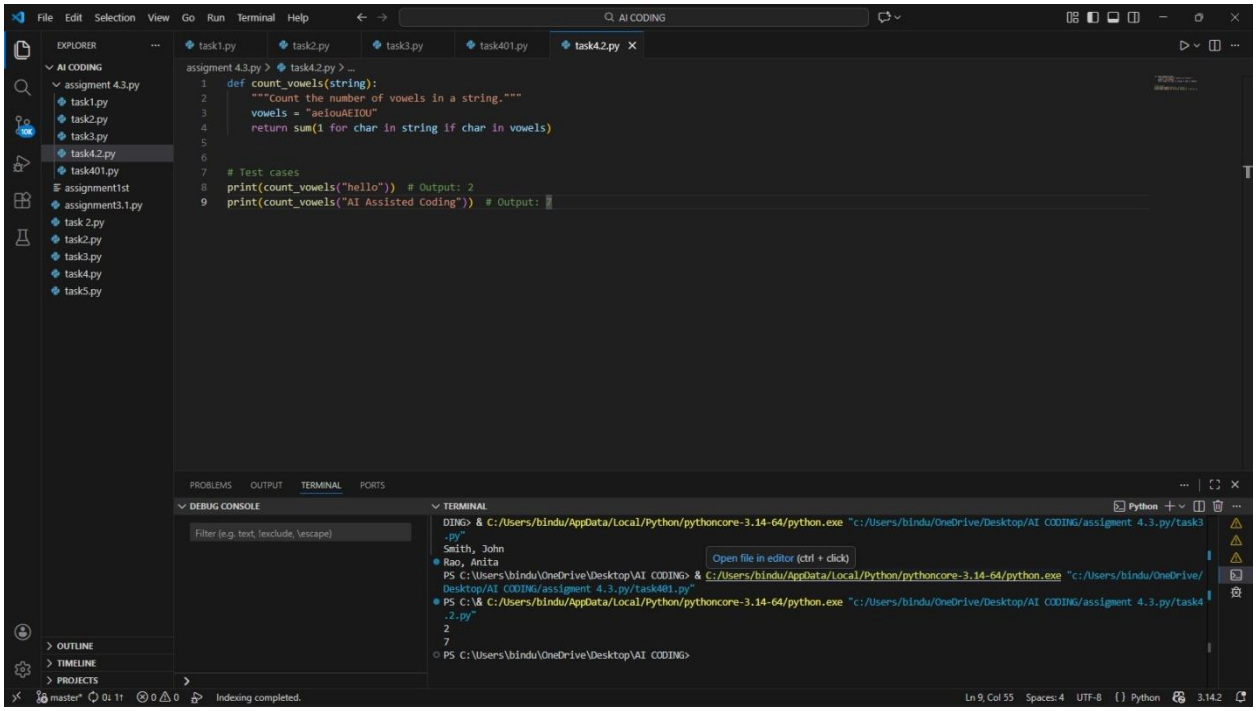
```
❖ efcount_vowels(string):
❖     """Countthenumberofvowelsinastring."""
❖     vowels="aeiouAEIOU"
❖     returnsum(1forcharinstringifcharinvowels)
❖
❖ #Testcases
❖ print(count_vowels("hello"))#Output: 2
❖ print(count_vowels("AIAssistedCoding"))#Output:7
```

  - **SampleInput:**
    Enterstring:HelloWorld
  - **SampleOutput:3**

Explanation(Few-Shot);

The functionusesapredefined vowelset andpython'ssumwiththegeneratorexpressionto count vowels efficiently the logic is compact and easier to understand due to example .





ComparisonTable

Criteria	**Zero-Shot**	**Few-Shot**
Accuracy	Correct	Correct
Readability	Medium	High
LogicalClarity	Simple	Veryclear
CodeLength	Longer	Shorter

Task5:Few-ShotPrompting –FileHandling(LineCount)

Scenario:Youarebuildingatextfileanalyzerthatcountsthenumberoflinesinafile. Clear examples help ensure correct file handling.

Promptused:

 # Write a Python function to read a text file#Thefunctionshouldtakethefilenameasinput #Openthefilein readmode#Countthetotalnumberoflinesinthefile #Return the line count as an integer

 # Example:

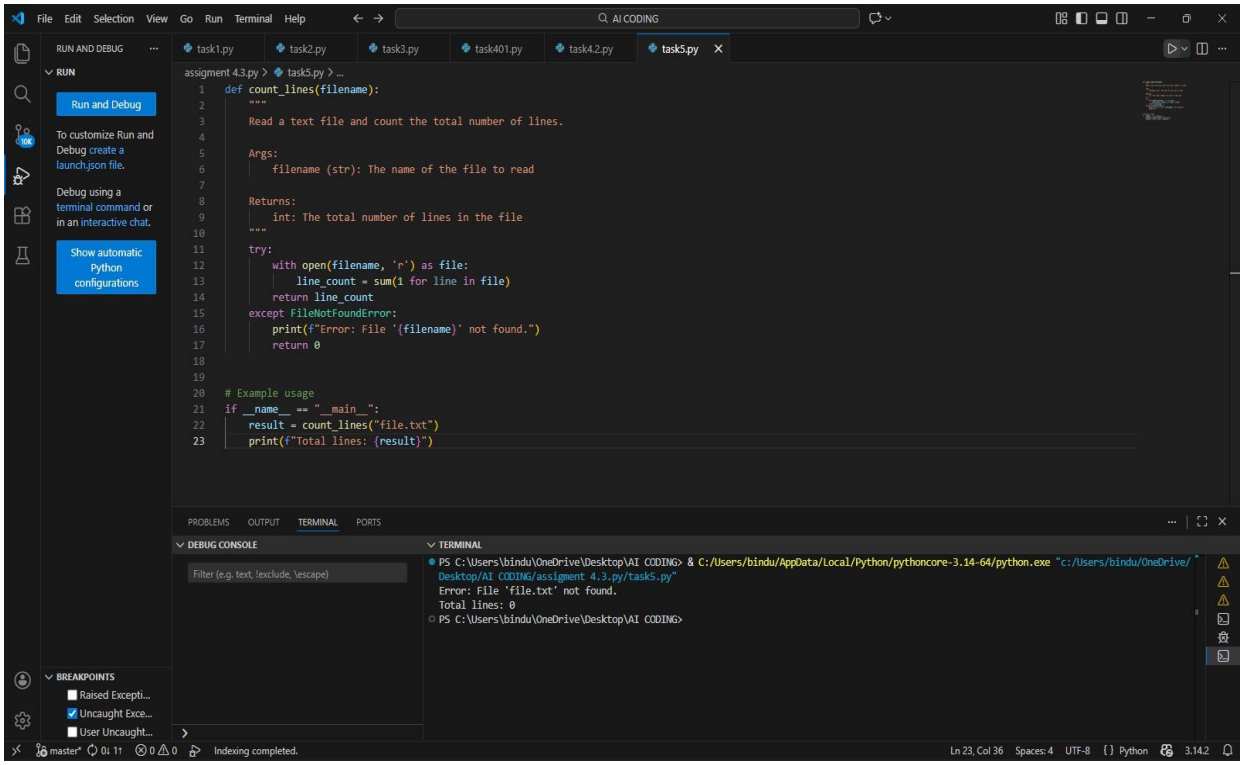
#If"file.txt"contains3lines,theoutput shouldbe3

Generatedcode:

```
defcount_lines(filename):
    """
    Readatextfileandcountthetotalnumberoflines.

    Args:
        filename(str):Thenameofthefileto read

    Returns:
        int:Thetotalnumberoflinesin the file
    """
    try:
        withopen(filename, 'r') asfile:
            line_count=sum(1forlineinfile)
        returnline_count
    exceptFileNotFoundError:
        print(f"Error:File'{filename}'notfound.") return0


#Exampleusage
ifname=="main":result=count_lines("file.txt")
    print(f"Totallines:{result}")
```

- **SampleInput:**
- print(count_lines("file.txt"))
- **SampleOutput:**
  3

ShortExplanationofLogic:
Theprogramreadsa textfileinreadmodeandcounts thenumber of lines presentinit a counter variable is usedtokeep trackofthenumber oflines. Eachtimea lineis readfromthefilethecounter increases by one. After reading the entire file, the function returns the total number of lines. This method ensures accurate line counting and is easy to understand


File Edit Selection View Go Run Terminal Help
AI CODING
RUN AND DEBUG
RUN
Run and Debug
To customize Run and Debug create a launch.json file.
Debug using a terminal command or in an interactive chat.
Show automatic Python configurations
task1.py
task2.py
task3.py
task401.py
task4.2.py
task5.py
assigment 4.3.py > task5.py > ...
def count_lines(filename):
"""
Read a text file and count the total number of lines.
Args:
filename (str): The name of the file to read
Returns:
int: The total number of lines in the file
"""
try:
with open(filename, 'r') as file:
line_count = sum(1 for line in file)
return line_count
except FileNotFoundError:
print(f"Error: File '{filename}' not found.")
return 0
# Example usage
if __name__ == "__main__":
result = count_lines("file.txt")
print(f"Total lines: {result}")
PROBLEMS OUTPUT TERMINAL PORTS
DEBUG CONSOLE
Filter (e.g. text, !exclude, \escape)
TERMINAL
PS C:\Users\bindu\OneDrive\Desktop\AI CODING> & C:/Users/bindu/AppData/Local/Python/pythoncore-3.14-64/python.exe "c:/Users/bindu/OneDrive/Desktop/AI CODING/assigment 4.3.py/task5.py"
Error: File 'file.txt' not found.
Total lines: 0
PS C:\Users\bindu\OneDrive\Desktop\AI CODING>
BREAKPOINTS
Raised Excepti...
Uncaught Exce...
User Uncaught...
master* 0↓ 1↑ 0 0
Indexing completed.
Ln 23, Col 36 Spaces: 4 UTF-8 Python 3.14.2