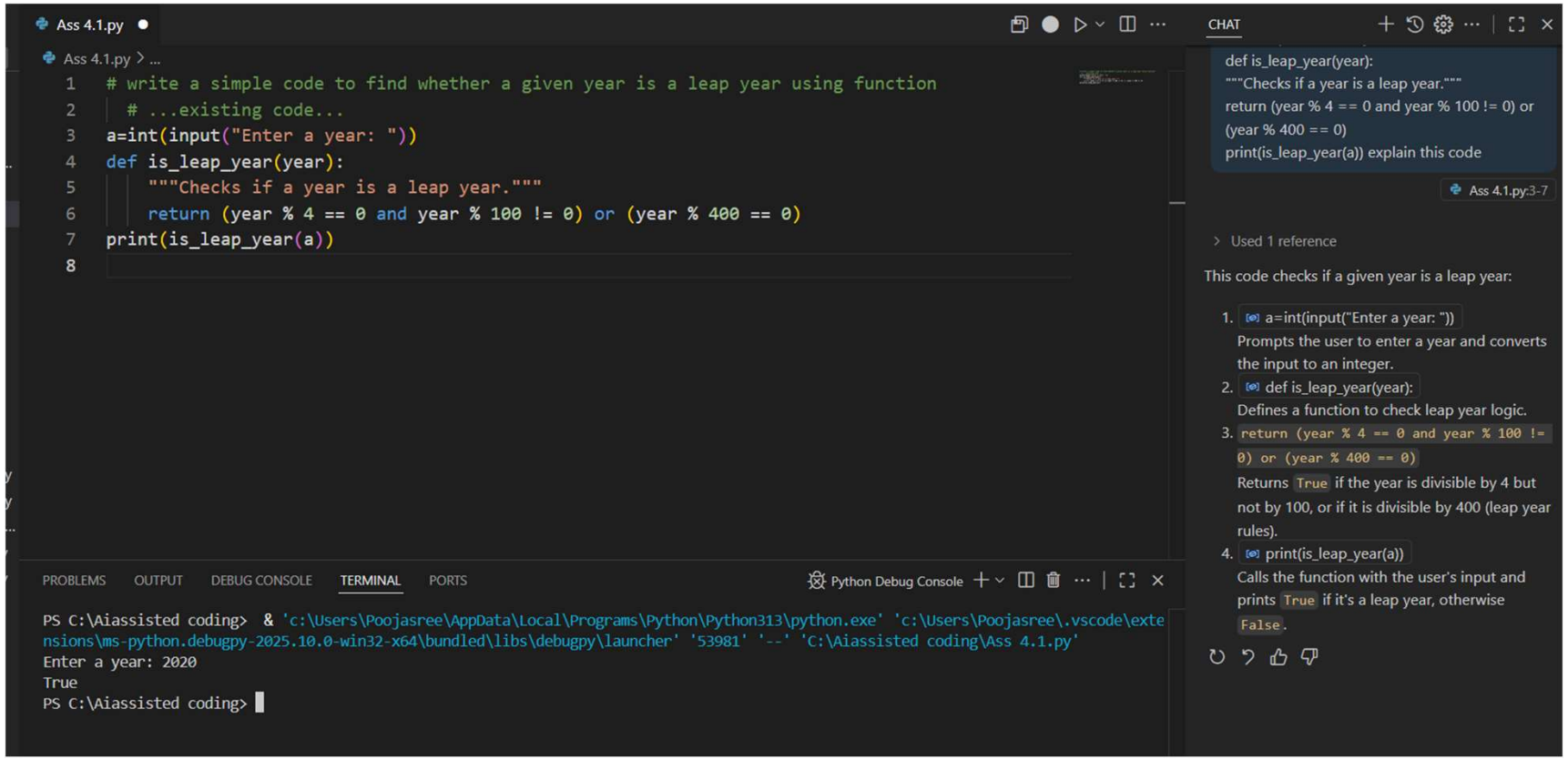
Assignment-4

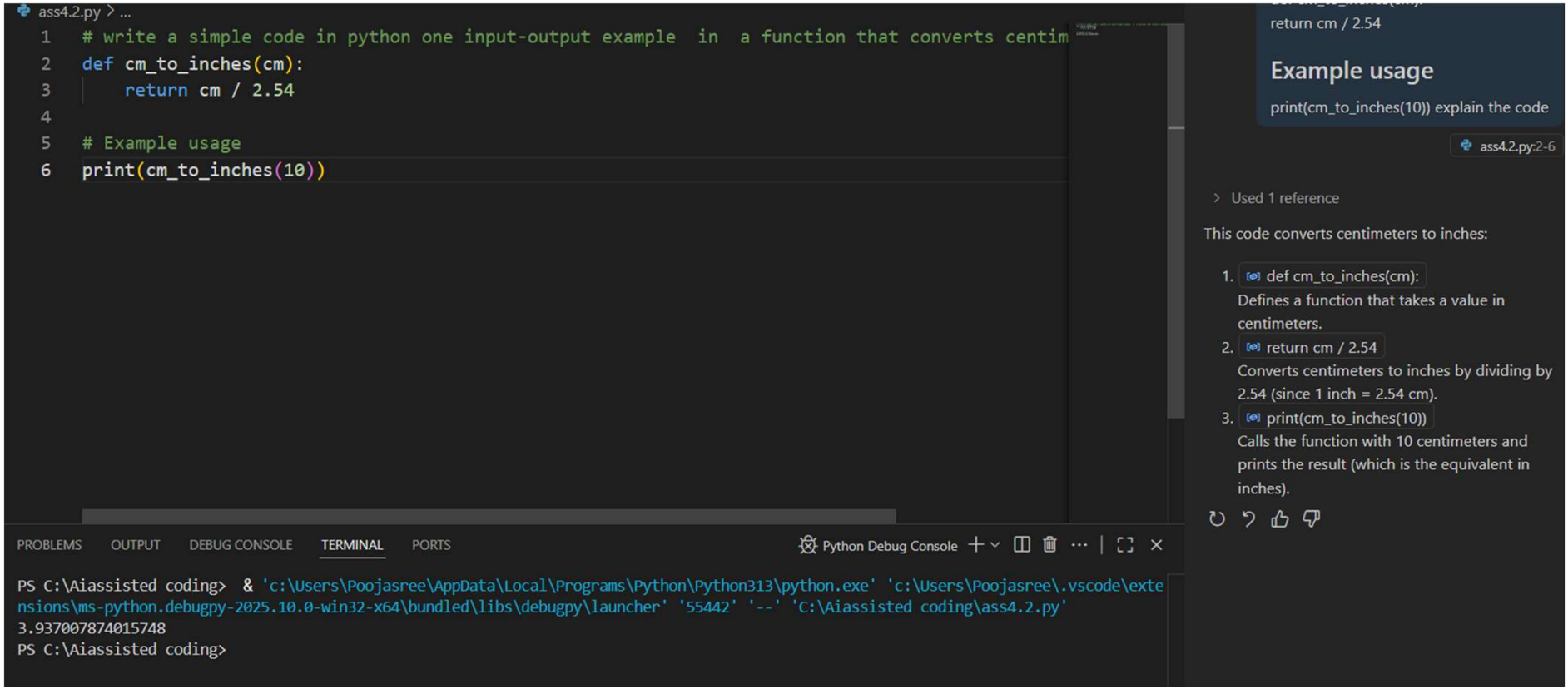
Task-1:

Function that checks whether a given year is a leap year

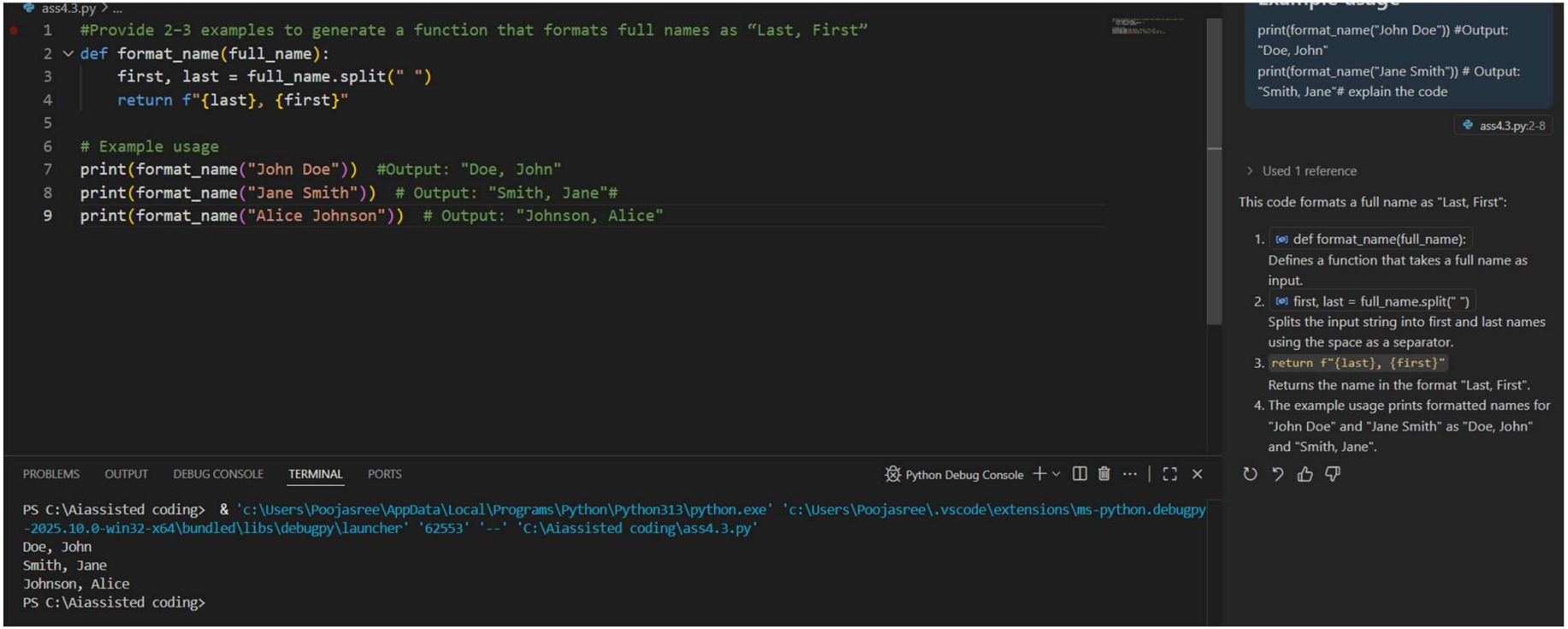


Task-2:

write a simple code in python one input-output example in a function that converts centimeters to inches.



Task-3:
#Provide 2–3 examples to generate a function that formats full names as “Last, First”



Task:4

counts the number of vowels in a string.

1.Zero shot

```python
a=input("enter a string")
def count_vowels(input_string):


    vowels = "aeiouAEIOU"
    vowel_count = 0
    for char in input_string:
        if char in vowels:
            vowel_count += 1
    return vowel_count
print(count_vowels(a))
```

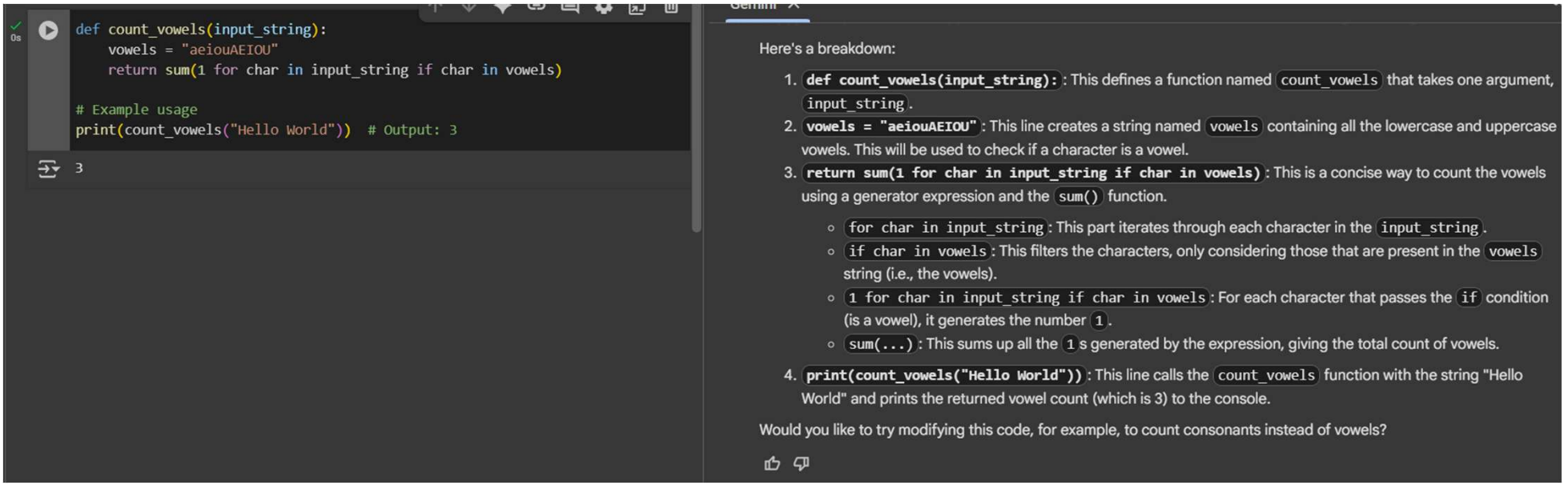
Here's a breakdown:

1. `a=input("enter a string")`: This line prompts the user to enter a string using the message "enter a string" and stores the entered string in the variable `a`.
2. `def count_vowels(input_string):`: This defines a function named `count_vowels` that takes one argument, `input_string`.
3. `vowels = "aeiouAEIOU"`: This line creates a string named `vowels` containing all the lowercase and uppercase vowels. This will be used to check if a character is a vowel.
4. `vowel_count = 0`: This initializes a variable `vowel_count` to 0. This variable will keep track of the number of vowels found in the input string.
5. `for char in input_string:`: This starts a loop that iterates through each character in the `input_string`. In each iteration, the current character is assigned to the variable `char`.
6. `if char in vowels:`: Inside the loop, this line checks if the current character (`char`) is present in the `vowels` string.
7. `vowel_count += 1`: If the character is a vowel (the condition in the `if` statement is true), this line increments the `vowel_count` by 1.
8. `return vowel_count`: After the loop finishes iterating through all the characters in the input string, this line returns the final value of `vowel_count`.
9. `print(count_vowels(a))`: This line calls the `count_vowels` function with the user's input string stored in variable `a` and prints the returned vowel count to the console.

Output:

```
enter a string poojasree
5
PS C:\Aiassisted coding>
```

One shot:



```python
def count_vowels(input_string):
    vowels = "aeiouAEIOU"
    return sum(1 for char in input_string if char in vowels)

# Example usage
print(count_vowels("Hello World"))  # Output: 3
```

```
3
```

Here's a breakdown:

1. `def count_vowels(input_string):` : This defines a function named `count_vowels` that takes one argument, `input_string`.
2. `vowels = "aeiouAEIOU"` : This line creates a string named `vowels` containing all the lowercase and uppercase vowels. This will be used to check if a character is a vowel.
3. `return sum(1 for char in input_string if char in vowels)` : This is a concise way to count the vowels using a generator expression and the `sum()` function.
    - `for char in input_string` : This part iterates through each character in the `input_string`.
    - `if char in vowels` : This filters the characters, only considering those that are present in the `vowels` string (i.e., the vowels).
    - `1 for char in input_string if char in vowels` : For each character that passes the `if` condition (is a vowel), it generates the number `1`.
    - `sum(...)` : This sums up all the `1`s generated by the expression, giving the total count of vowels.
4. `print(count_vowels("Hello World"))` : This line calls the `count_vowels` function with the string "Hello World" and prints the returned vowel count (which is 3) to the console.

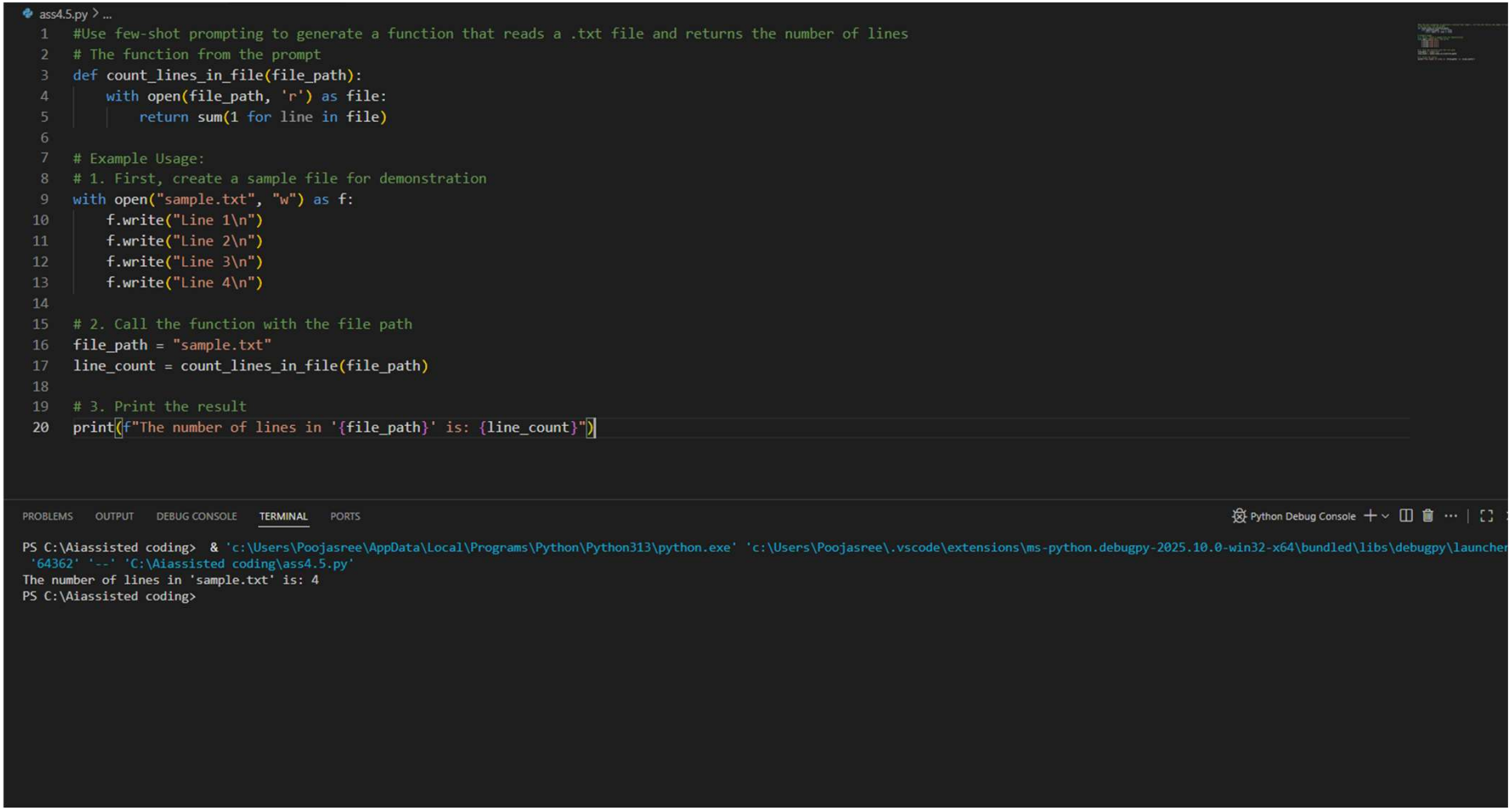
Would you like to try modifying this code, for example, to count consonants instead of vowels?

Zero-Shot Analysis:
In a zero-shot context, which means evaluating the code without any prior examples or context about the user's intent, the first function is superior. It uses a generator expression (1 for char in input_string if char in vowels) combined with the built-in sum() function. This approach is highly efficient because it doesn't create a temporary list in memory, as it generates the values on the fly.

A one-shot analysis:
where we consider the code's functionality based on a single example, reveals that both snippets are functionally correct. Both functions,count vowels (input_string), take a string as input and correctly return the number of vowels.

Task-5:

#Use few-shot prompting to generate a function that reads a .txt file and returns the number of lines



Explanation:

Here's a breakdown:

1. `def count_lines_in_file(file_path):` : This defines a function named `count_lines_in_file` that takes one argument, `file_path`.
2. `with open(file_path, 'r') as file:` : This opens the file specified by `file_path` in read mode (`'r'`). The `with` statement ensures the file is automatically closed even if errors occur. The opened file object is assigned to the variable `file`.
3. `return sum(1 for line in file)` : This part iterates through each `line` in the `file` object. For each line, it generates the number `1`. The `sum()` function then adds up all these `1`s, effectively counting the number of lines.
4. `with open("sample.txt", "w") as f:` : This creates a sample file named `sample.txt` in write mode (`"w"`). If the file already exists, it will be overwritten. The file object is assigned to the variable `f`.
5. `f.write("Line 1\n")` (and similar lines): These lines write the specified strings to the `sample.txt` file. The `\n` at the end of each string creates a new line.
6. `file_path = "sample.txt"` : This sets the `file_path` variable to the name of the sample file.
7. `line_count = count_lines_in_file(file_path)` : This calls the `count_lines_in_file` function with the `file_path` and stores the returned line count in the `line_count` variable.
8. `print(f"The number of lines in '{file_path}' is: {line_count}")` : This prints a formatted string to the console showing the name of the file and the calculated number of lines.

Sample.txt file

```
I am poojasree
i am currently persuing cse aiml from sru
i am  in 2nd year now
i  am from hanamkonda
```