AI ASSISTED CODING

LAB ASSIGNMENT – 4.3

NAVYA MASADI

2403A53020

24BTCAICYB01-2-1

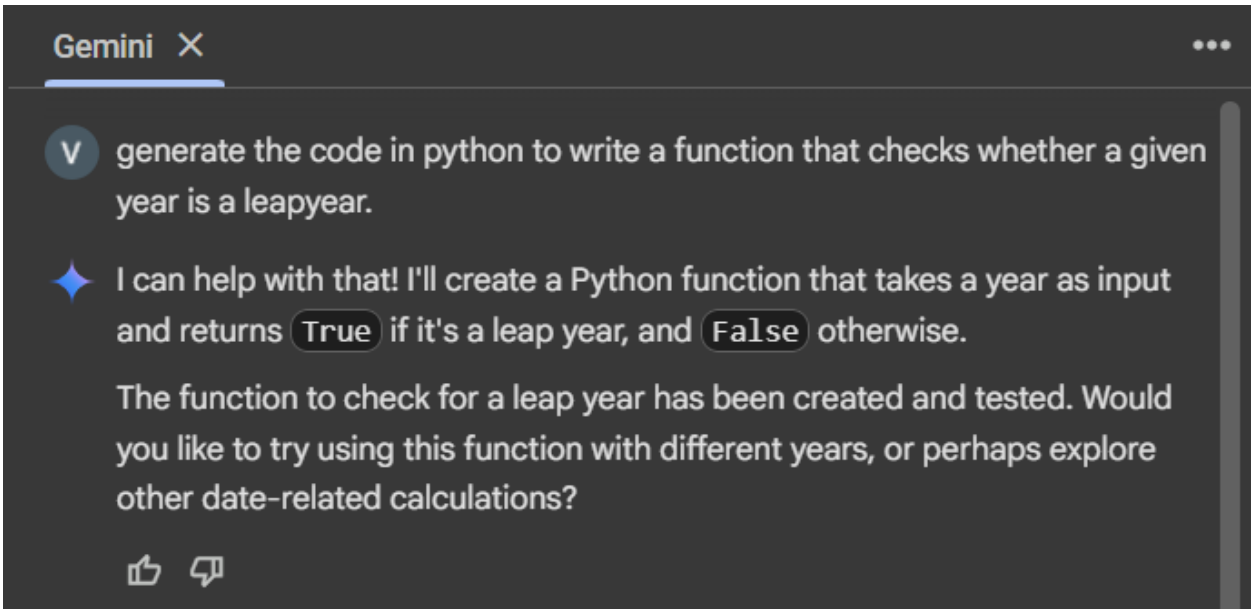
Task Description#1

● Zero-shot: Prompt AI to write a function that checks whether a given year is a leapyear.

Expected Output#1

● AI-generated function with no examples provided

PROMPT:



CODE:

```
Start coding or generate with AI.
```

```
def is_leap_year(year):
  """
  Checks if a given year is a leap year.

  Args:
    year: The year to check.

  Returns:
    True if the year is a leap year, False otherwise.
  """
  return (year % 4 == 0 and year % 100 != 0) or (year % 400 == 0)

# Example usage:
print(f"Is 2000 a leap year? {is_leap_year(2000)}")
print(f"Is 1900 a leap year? {is_leap_year(1900)}")
print(f"Is 2024 a leap year? {is_leap_year(2024)}")
print(f"Is 2023 a leap year? {is_leap_year(2023)}")
```

OUTPUT:

```
Is 2000 a leap year? True
Is 1900 a leap year? False
Is 2024 a leap year? True
Is 2023 a leap year? False
```

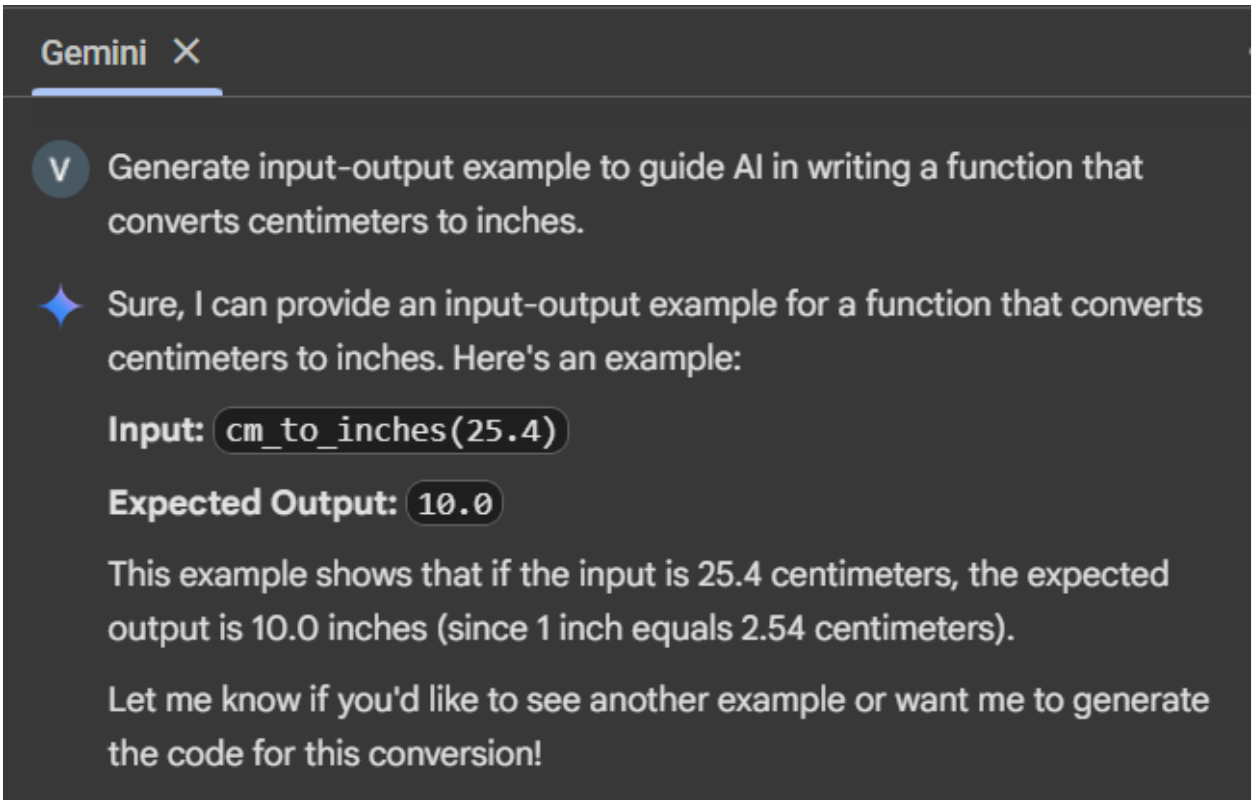
Task Description#2

● One-shot: Give one input-output example to guide AI in writing a function thatconverts centimeters to inches.

Expected Output#2

● Function with correct conversion logic.

PROMPT:



CODE AND OUTPUT:

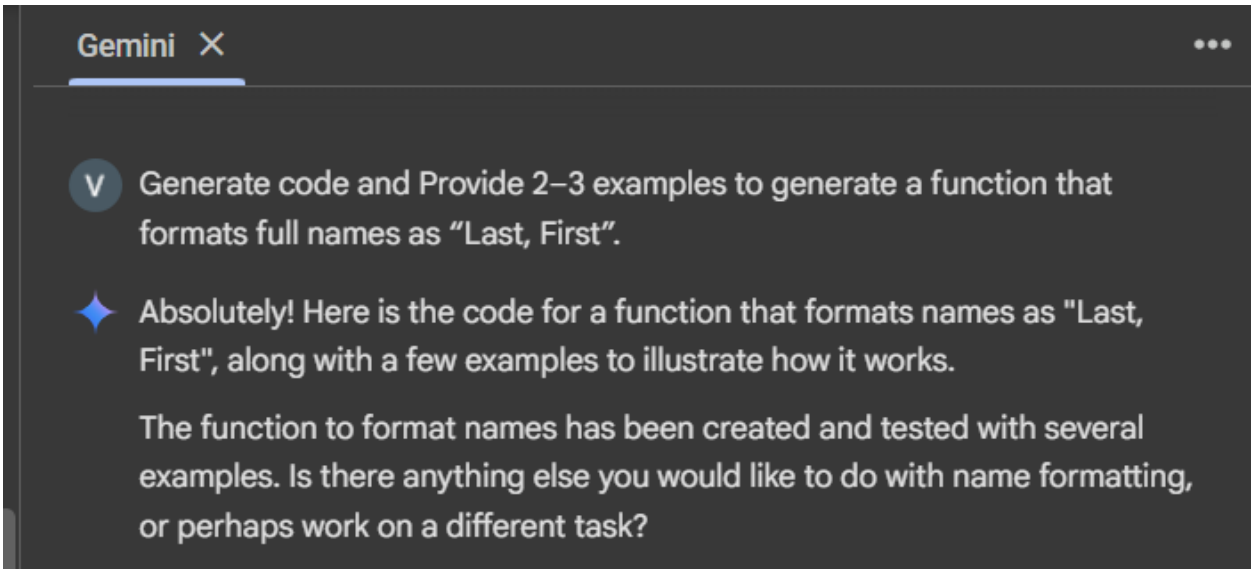
Task Description#3

● Few-shot: Provide 2–3 examples to generate a function that formats full names as“Last, First”.

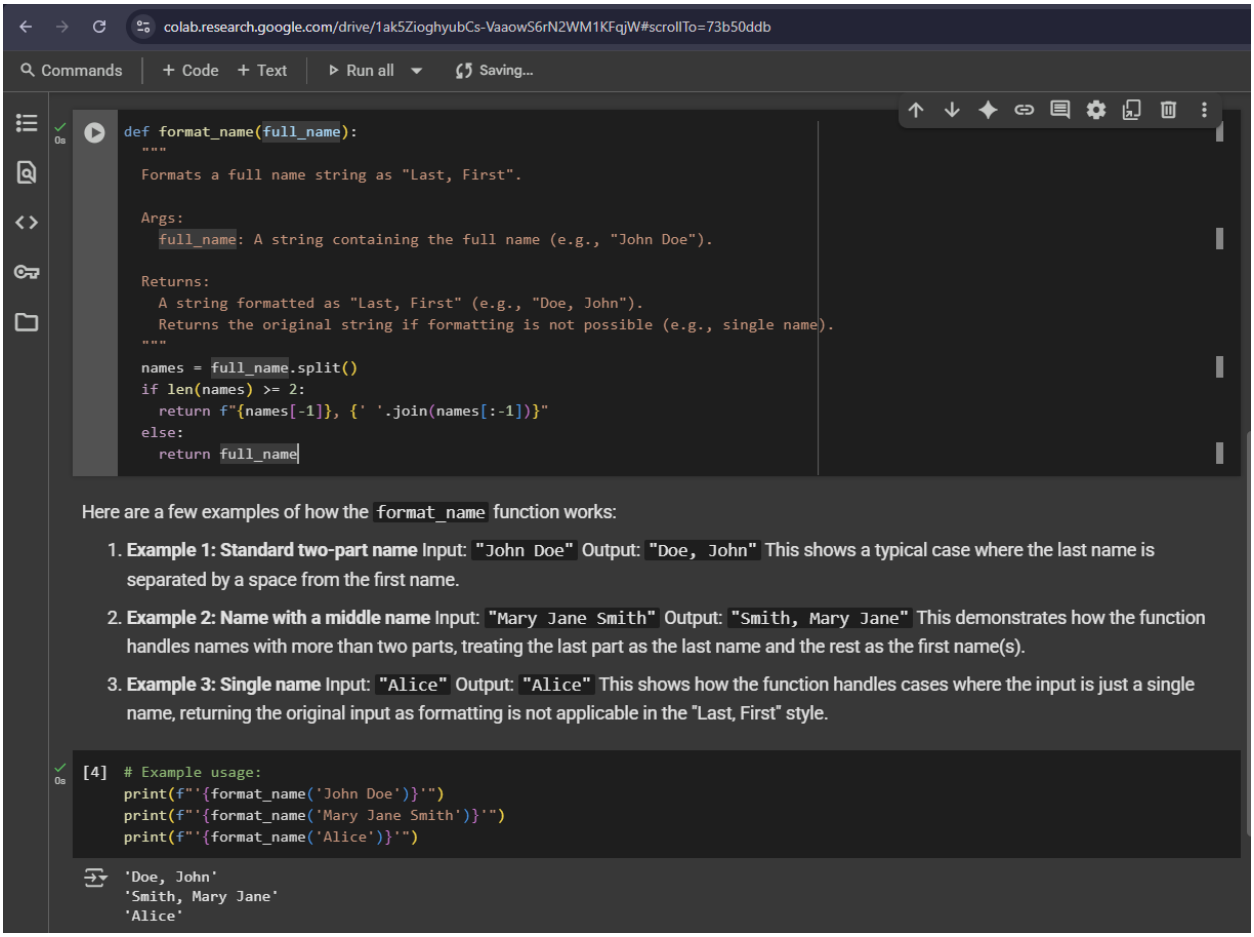
Expected Output#3

● Well-structured function respecting the examples.

PROMPT:



CODE AND OUTPUT:



Task Description#4

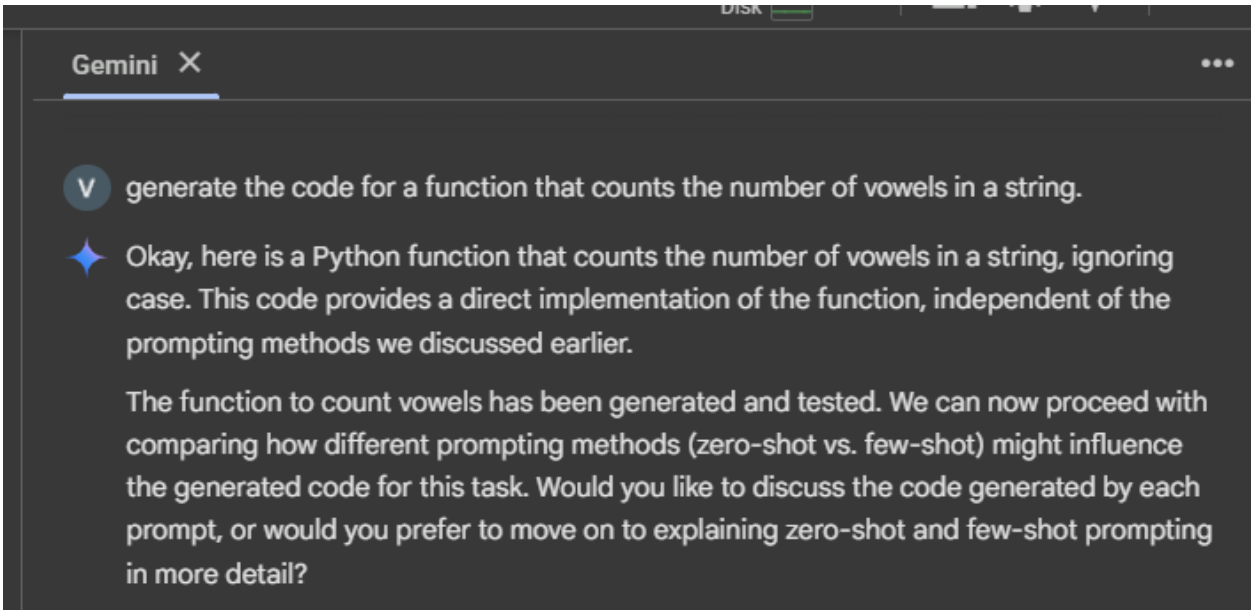
● Compare zero-shot and few-shot prompts for writing a function that counts the
number of vowels in a string

.

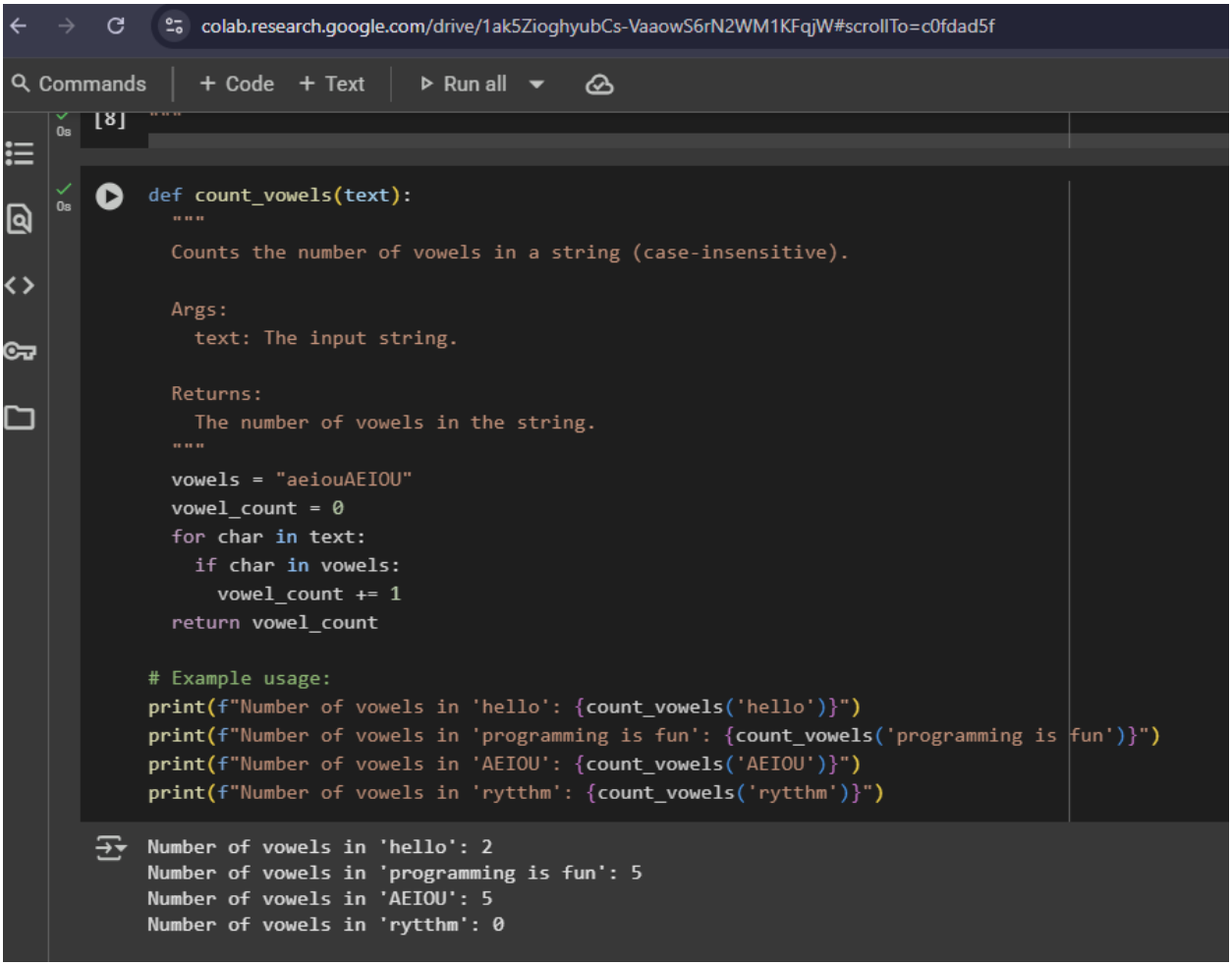
Expected Output#4

● Functional output and comparative reflection.

PROMPT:



CODE AND OUTPUT:



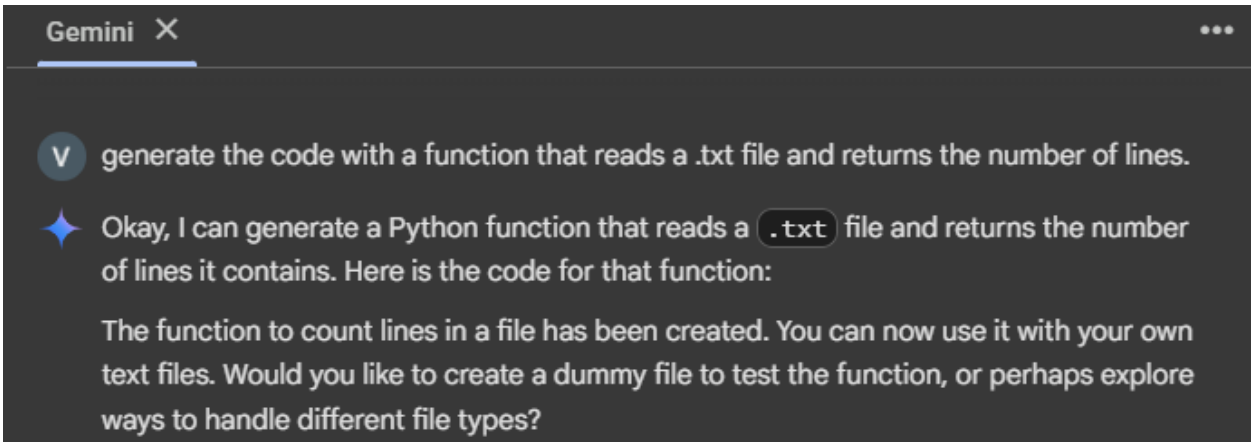
Task Description#5

● Use few-shot prompting to generate a function that reads a .txt file and returns thenumber of lines.

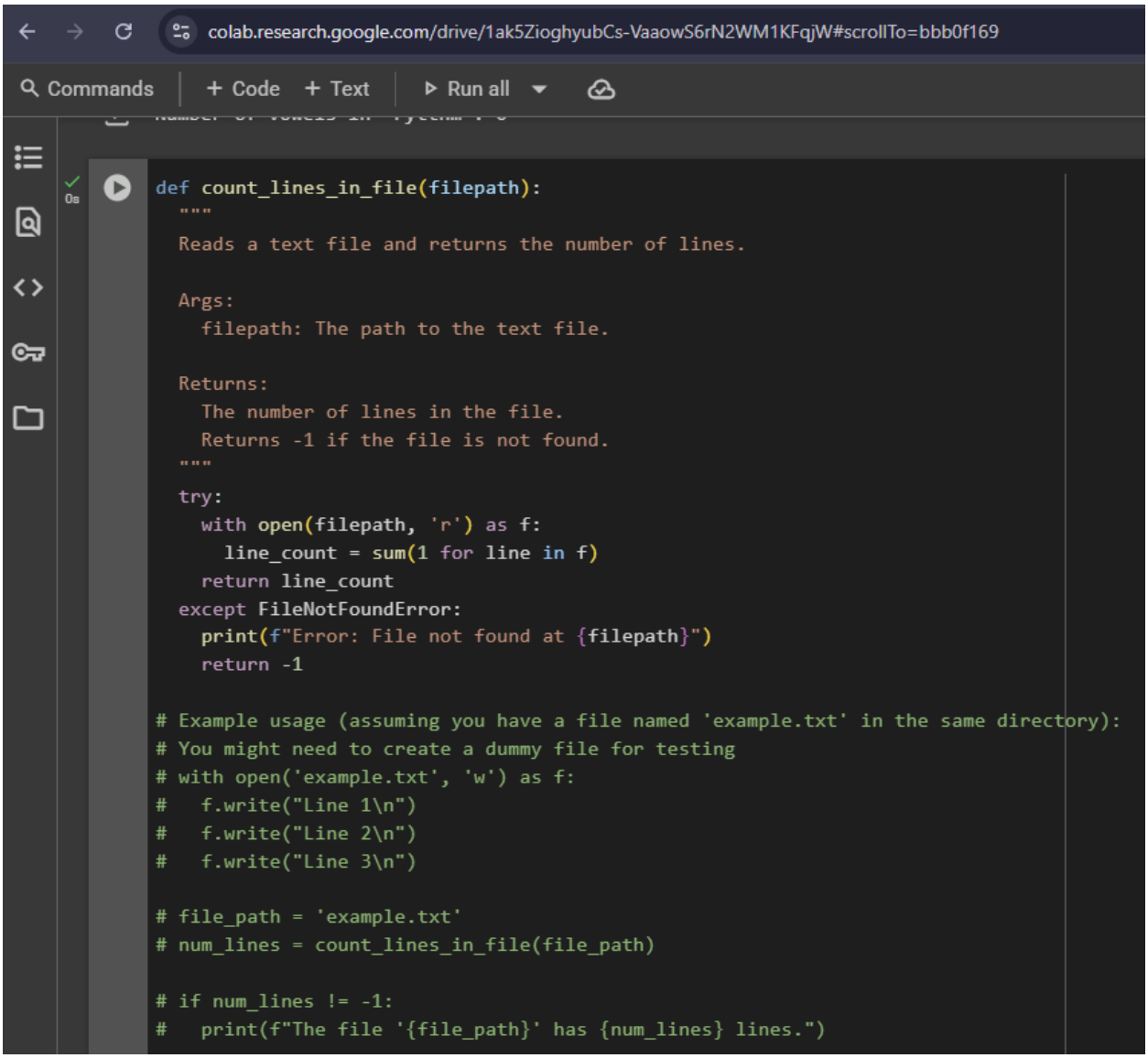
Expected Output#5

● Working file-processing function with AI-guided logic.

PROMPT:



CODE:


colab.research.google.com/drive/1ak5ZioghyubCs-VaaowS6rN2WM1KFqjW#scrollTo=bbb0f169
Commands
+ Code
+ Text
Run all
0s
def count_lines_in_file(filepath):
"""
Reads a text file and returns the number of lines.
Args:
filepath: The path to the text file.
Returns:
The number of lines in the file.
Returns -1 if the file is not found.
"""
try:
with open(filepath, 'r') as f:
line_count = sum(1 for line in f)
return line_count
except FileNotFoundError:
print(f"Error: File not found at {filepath}")
return -1
Example usage (assuming you have a file named 'example.txt' in the same directory):
You might need to create a dummy file for testing
with open('example.txt', 'w') as f:
f.write("Line 1\n")
f.write("Line 2\n")
f.write("Line 3\n")
file_path = 'example.txt'
num_lines = count_lines_in_file(file_path)
if num_lines != -1:
print(f"The file '{file_path}' has {num_lines} lines.")