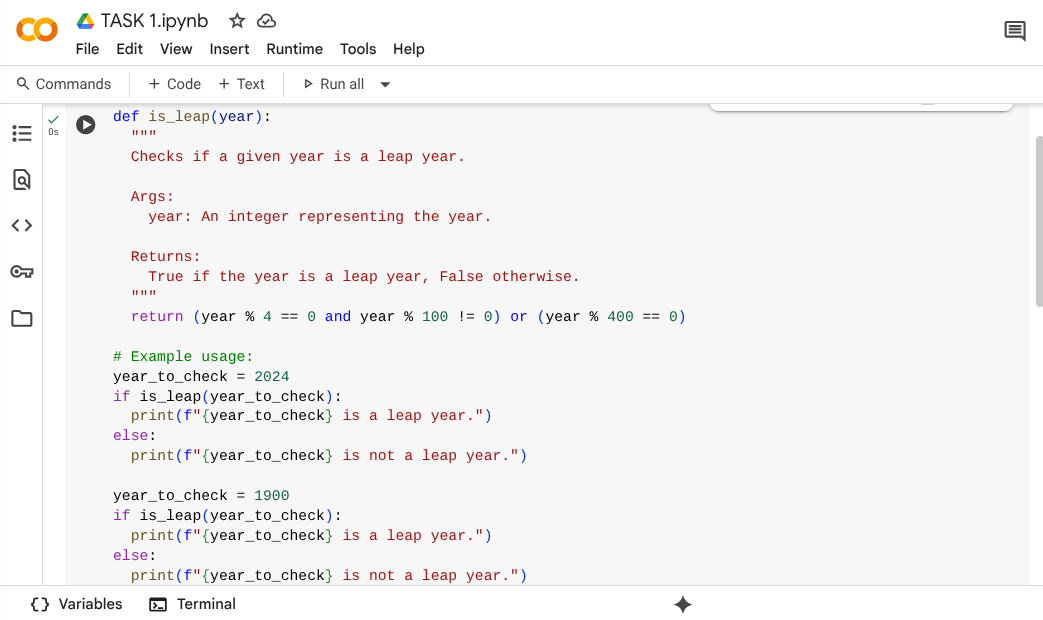
Assignment-4.3

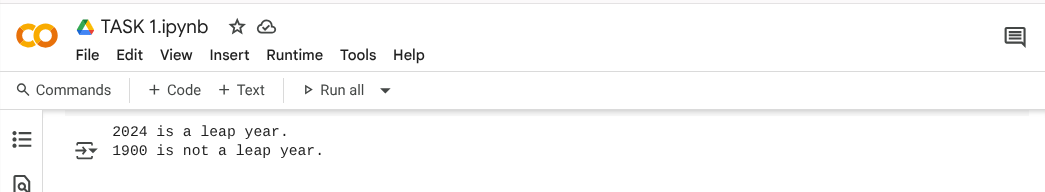
Task-1:

Write a python program that checks the given year is leap year using functions

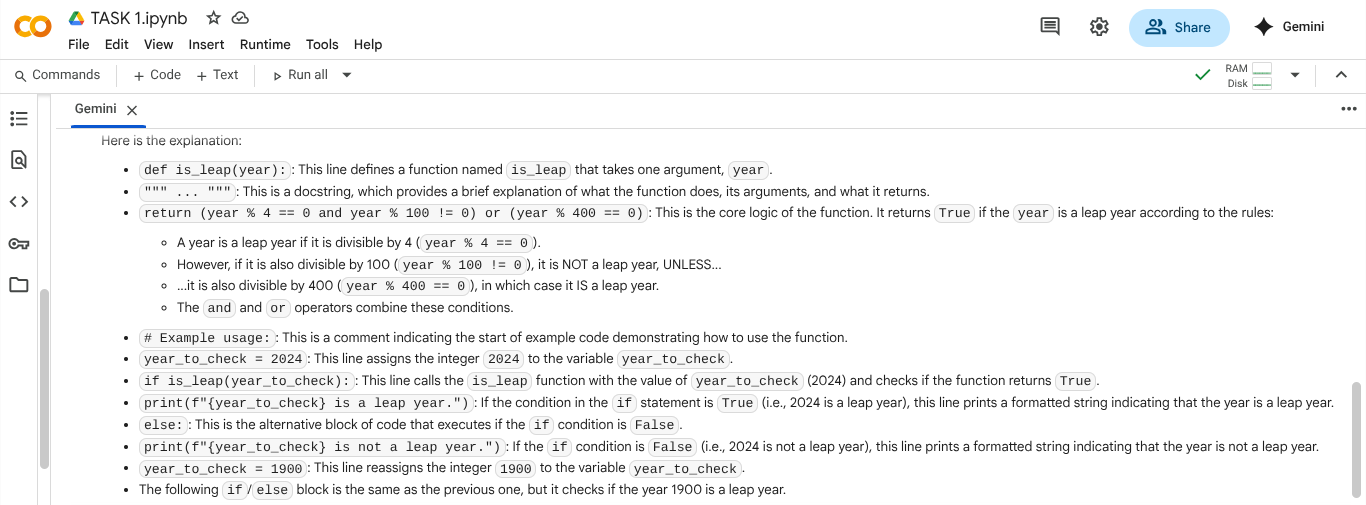
Code:



Output:



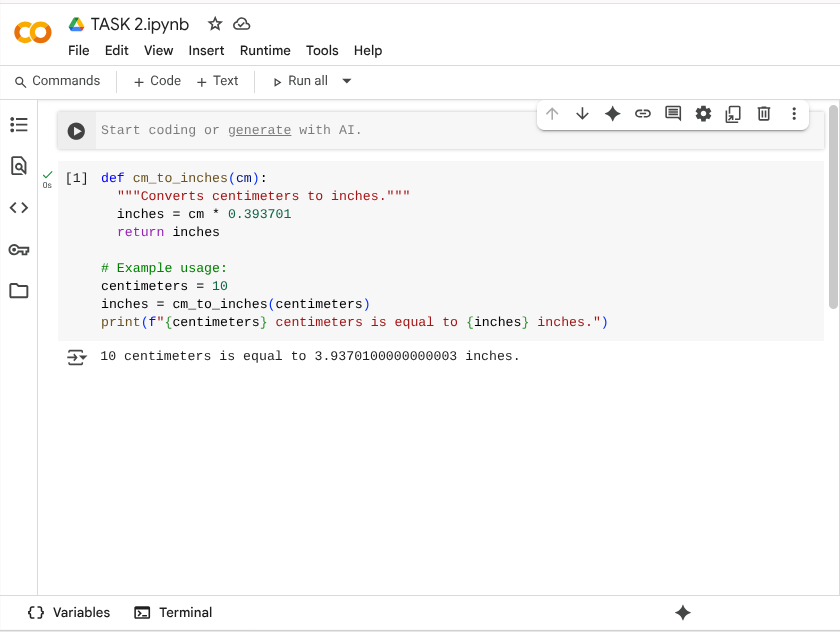
Explanation:



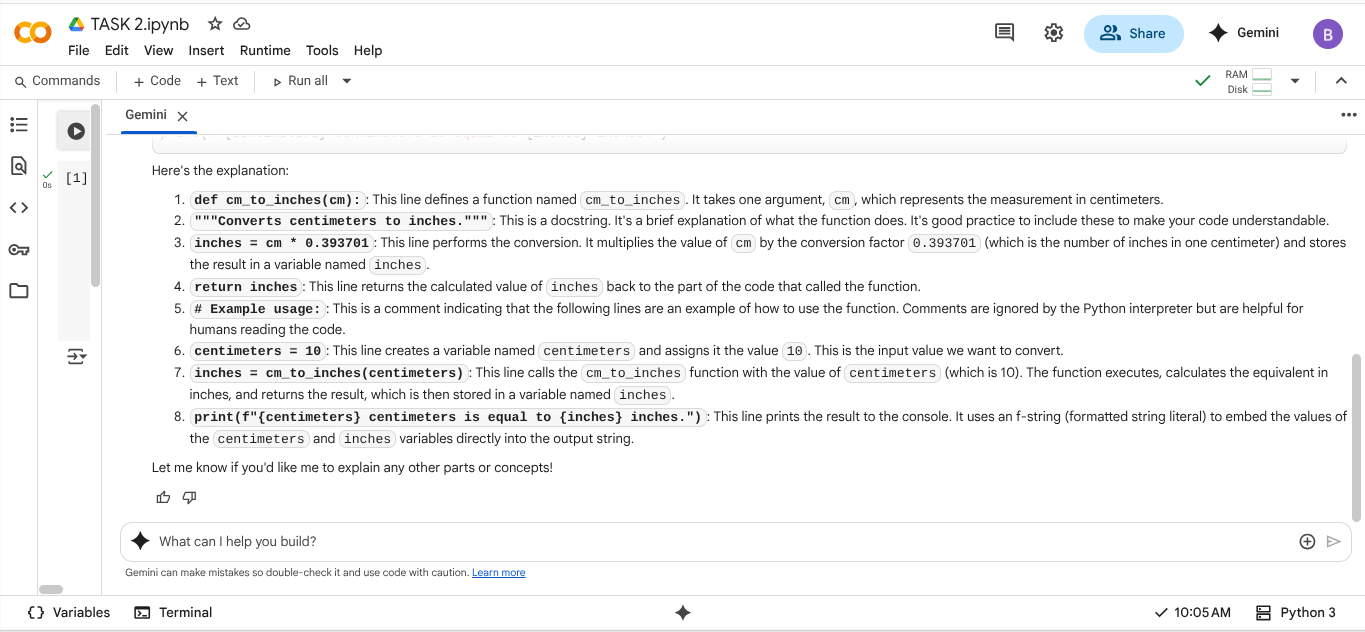
Task-2:

Write a program in python that converts centimeters into inches using one shot.

Code and output:



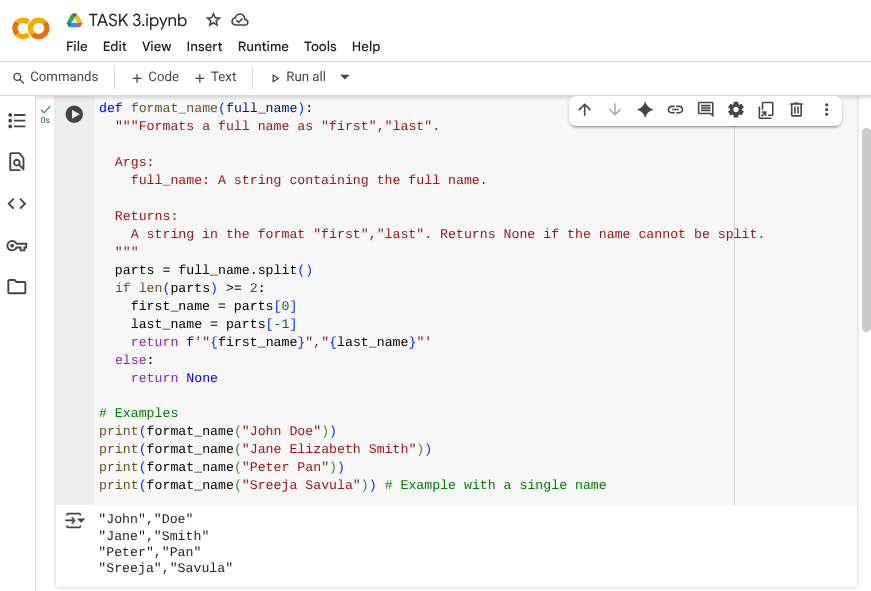
Explanation:



Task-3:

Write a python program that formats full names into “first” and “last” using few shot.

Code and Output:

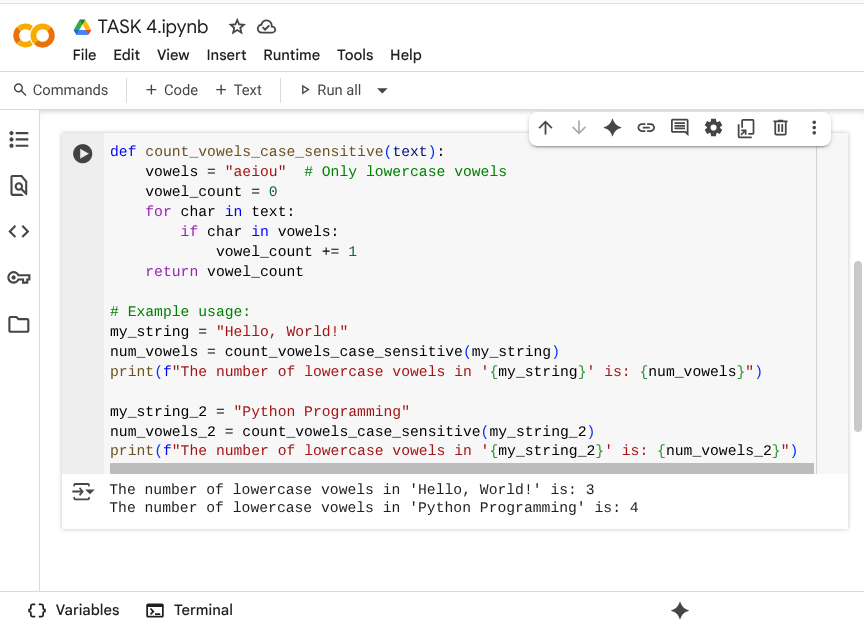


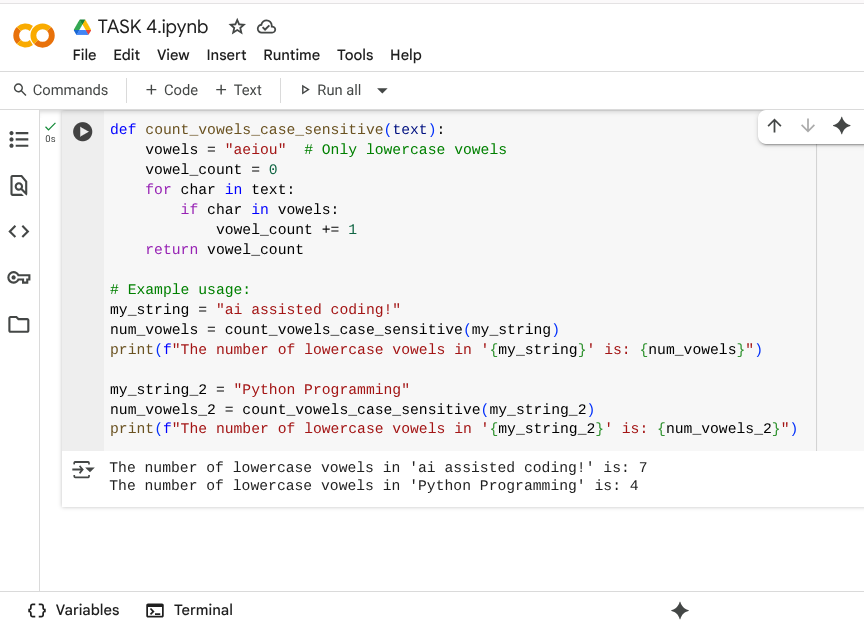
Explanation:



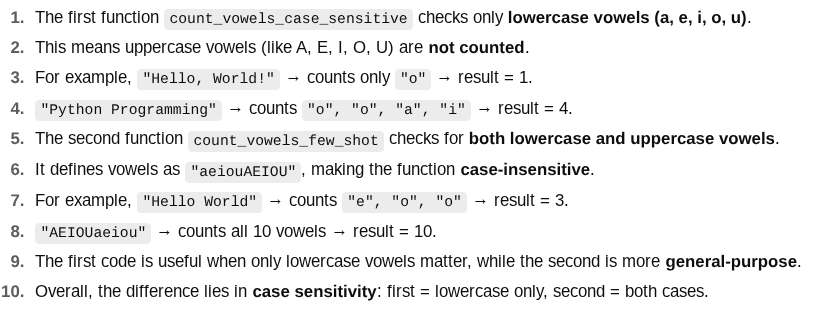
Task-4:

Write a python program that counts the number of vowels in a string.

Zero Shot Code and Output:

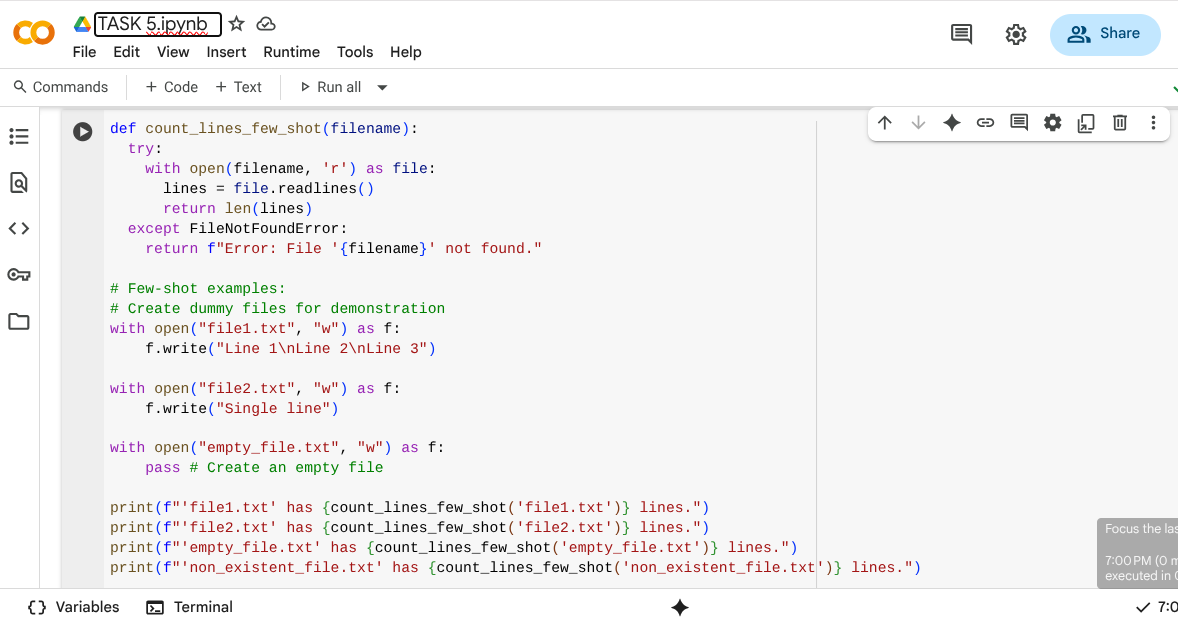
Few Shot Code and Output:

Comparision between Zero shot and Few shot code Explanation:

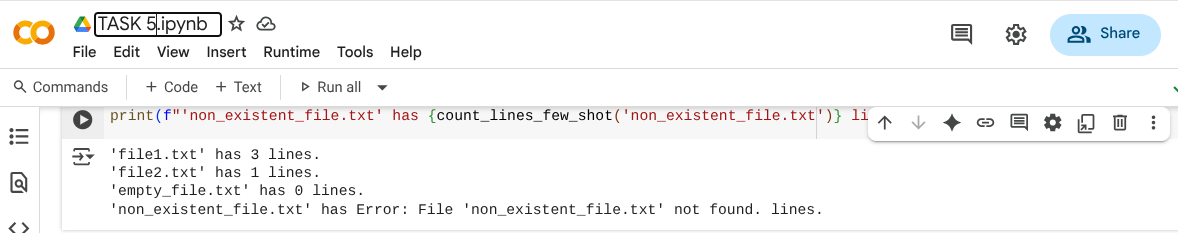


Task-5:

Write a python program that reads a .txt file and returns the number of lines using few short.

Code:

Output:



Explanation:

1. **def count\_lines\_few\_shot(filename):**: Defines a function to count lines in a file.
2. **"""..."""**: A docstring explaining the function's purpose, arguments, and return value.
3. **try:**: Starts a block to handle potential errors, like the file not being found.
4. **with open(filename, 'r') as file:**: Opens the specified file for reading.
5. **lines = file.readlines()**: Reads all lines into a list.
6. **return len(lines)**: Returns the number of lines in the list.
7. **except FileNotFoundError:**: Catches the error if the file doesn't exist.
8. **return f"Error: File '{filename}' not found."**: Returns an error message if the file is not found.
9. **# Few-shot examples:**: Comments indicating the following lines are examples.
10. **# Create dummy files for demonstration**: Comment explaining dummy file creation.
11. **with open("file1.txt", "w") as f: f.write("Line 1\nLine 2\nLine 3")**: Creates "file1.txt" with three lines.
12. **with open("file2.txt", "w") as f: f.write("Single line")**: Creates "file2.txt" with one line.
13. **with open("empty\_file.txt", "w") as f: pass**: Creates an empty "empty\_file.txt".
14. **print(f"'file1.txt' has {count\_lines\_few\_shot('file1.txt')} lines.")**: Prints the line count for "file1.txt".
15. **print(f"'non\_existent\_file.txt'has {count\_lines\_few\_shot('non\_existent\_file.txt')} lines.")**: Prints the result for a non-existent file (showing the error handling).