AI ASSISTED CODING

LAB-7.4

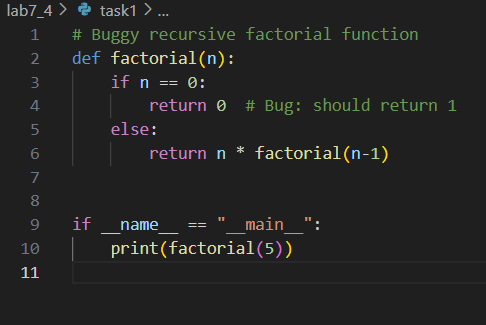
BATCH:04

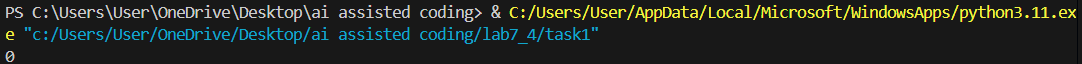
ROLLNO:2403A52096

TASK1:

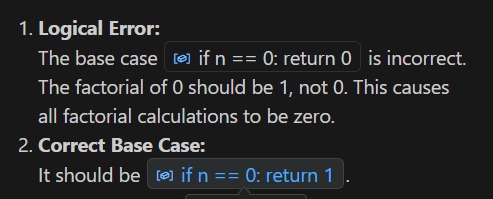
PROMPT: Introduce a buggy Python function that calculates the factorial of a number using recursion.

CODE:



OUTPUT: 

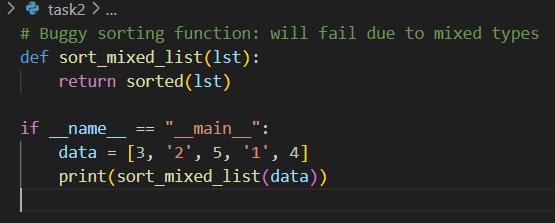
ERRORS:



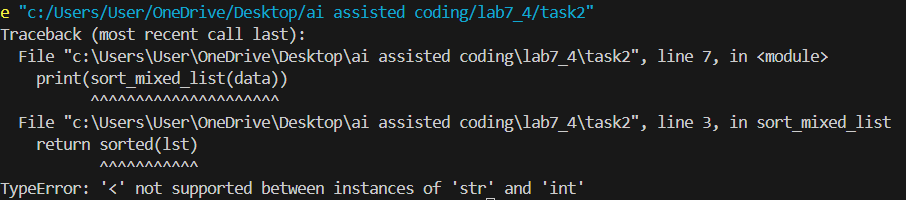
TASK2:

PROMPT: Provide a list sorting function that fails due to a type error (e.g., sorting list with mixed integers and strings). Prompt AI to detect the issue and fix the code for consistent sorting.

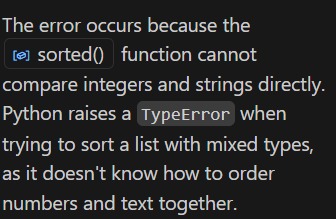
CODE:



OUTPUT:



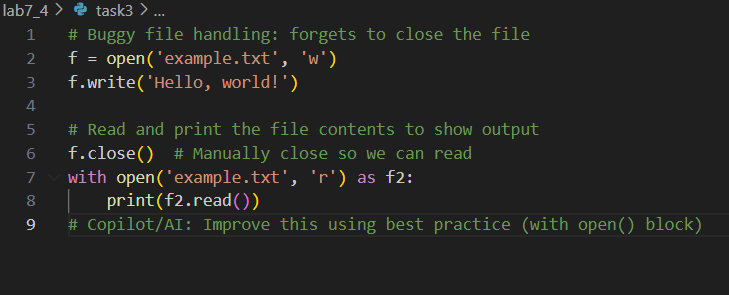
ERRORS:



TASK3:

PROMPT:Write a Python snippet for file handling that opens a file but forgets to close it.

CODE:

  
OUTPUT:



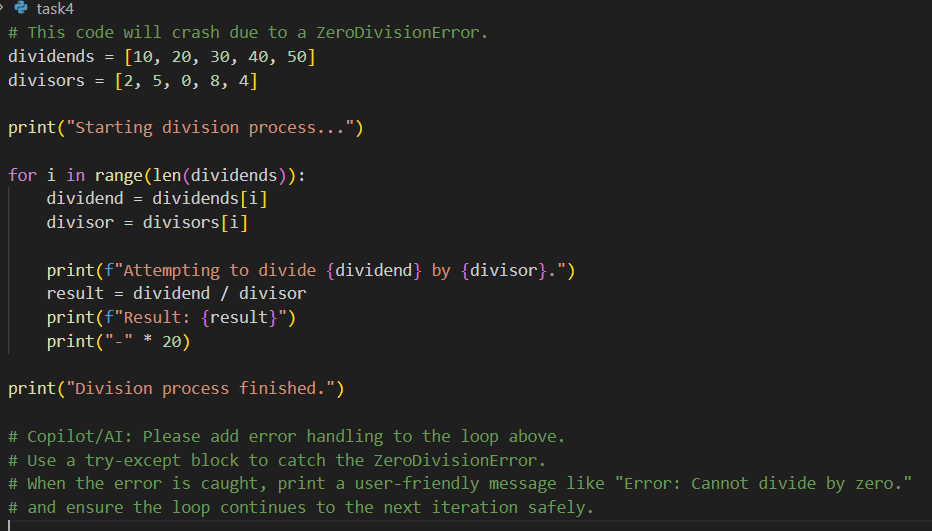
EXPLANATION:

Using the with open() statement is the standard best practice for file handling in Python. It automatically manages closing the file for you, even if an error occurs inside the block. This prevents potential resource leaks and makes the code safer and more readable.

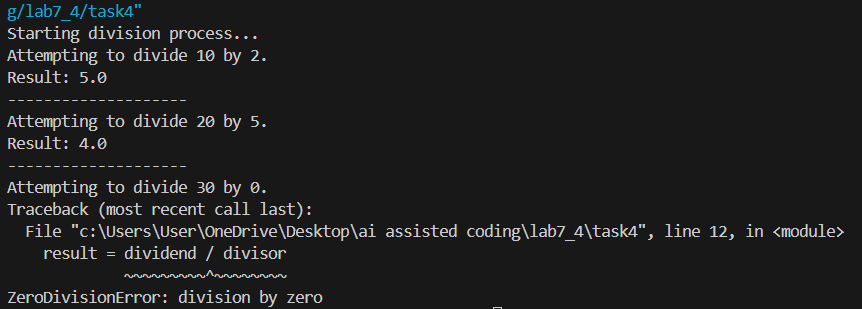
TASK4:

PROMPT:Provide a piece of code with a ZeroDivisionError inside a loop.

CODE:



OUTPUT:



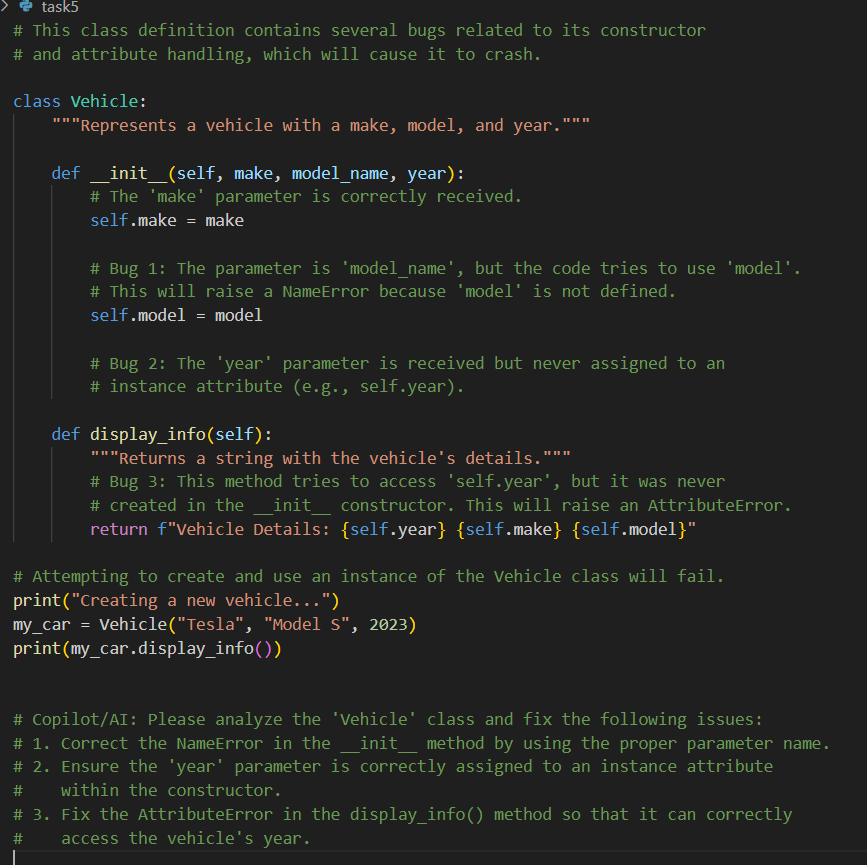
EXPLANATION:

This script performs element-wise division on numbers from two lists. It will crash with a ZeroDivisionError because it attempts to divide by zero. The comments suggest adding try-except error handling to catch the error, allowing the program to continue running safely.

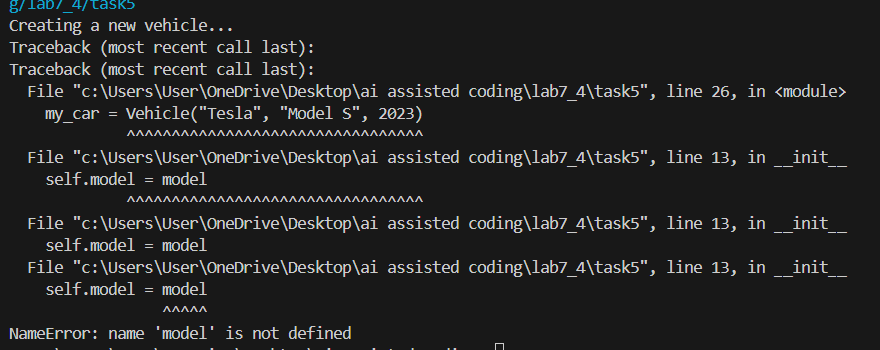
TASK5:

PROMPT:Include a buggy class definition with incorrect \_\_init\_\_ parameters or attribute references.

CODE:



OUTPUT:



EXPLANATION:

This Vehicle class is buggy. Its constructor (\_init\_) fails to correctly assign the model and year attributes. This causes the program to crash with an error when you try to create a Vehicle object and display its information.