ASSIGNMENT-7

NAME: A. MANIDEEPIKA

HT.NO: 2403A52052

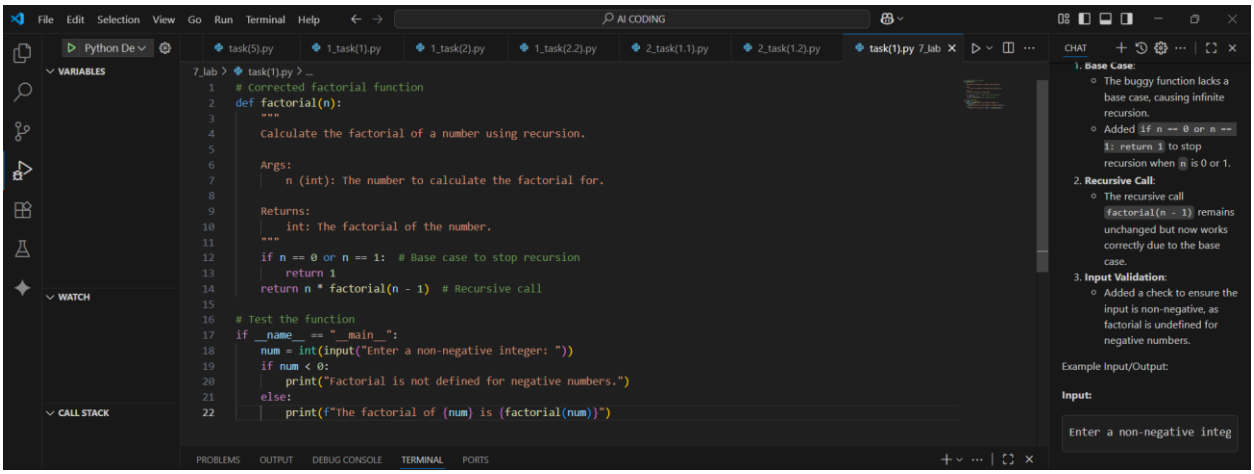
BATCH: AIB03

Task 1:

Introduce a buggy Python function that calculates the factorial of a number using recursion. Use Copilot or Cursor AI to detect and fix the logical or syntax errors.

Prompt: correct my error.

Code:



OP:

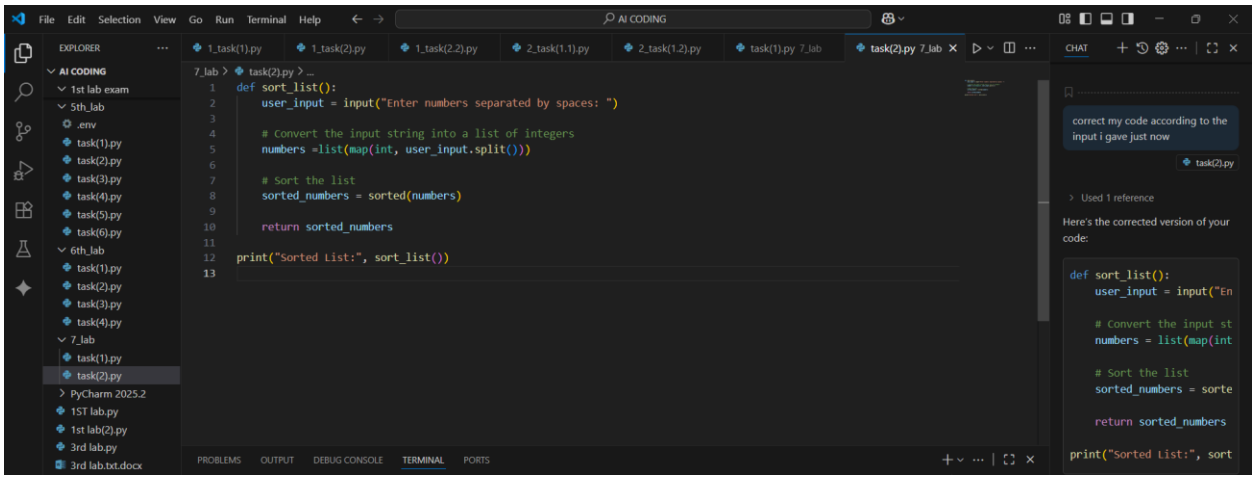
OBSERVATION: The buggy function lacks a base case, causing infinite recursion. Added if n == 0 or n == 1: return 1 to stop recursion when n is 0 or 1. The recursive call factorial (n - 1) remains unchanged but now works correctly due to the base case. Added a check to ensure the input is non-negative, as factorial is undefined for negative numbers.

Task 2:

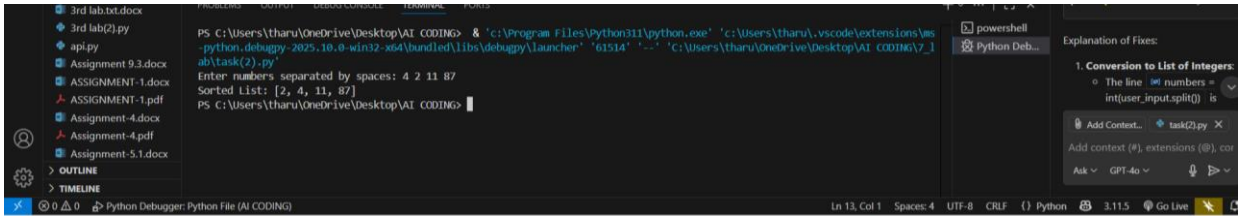
Provide a list sorting function that fails due to a type error (e.g., sorting list with mixed
integers and strings). Prompt AI to detect the issue and fix the code for consistent sorting

Prompt: correct my code according to the input i gave just now

Code:



OP:



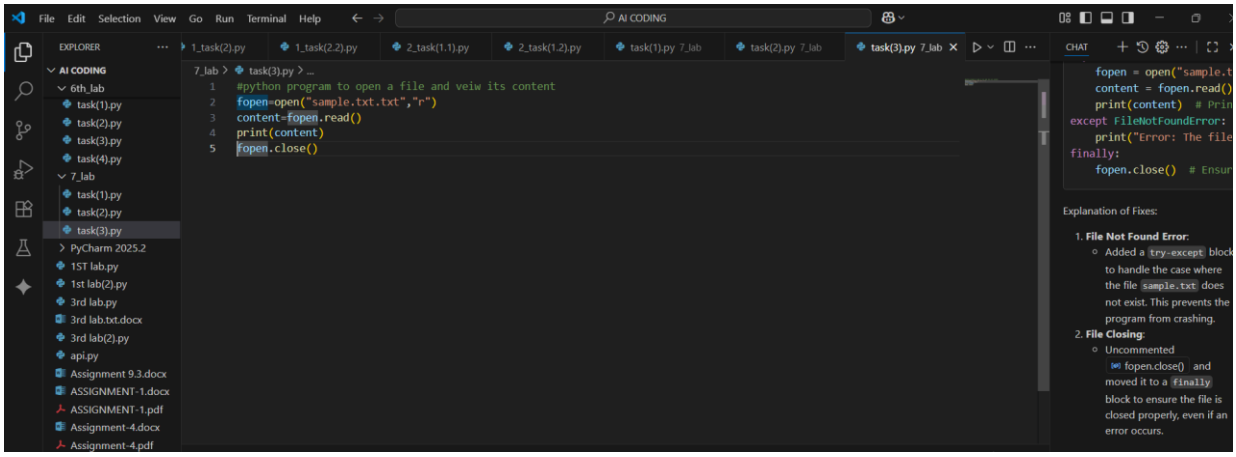
OBSERVATION: The line number=int(user_input.split()) is incorrect because split() returns a list of strings, and int() cannot directly convert a list. Fixed by using list(map(int, user_input.split())), which converts each string in the list to an integer. The sorted() function works correctly once the input is properly converted to a list of integers.

Task 3:

Write a Python snippet for file handling that opens a file but forgets to close it. Ask Copilot or Cursor AI to improve it using the best practice (e.g., with open() block)

Prompt: correct my error

Code:



OP:

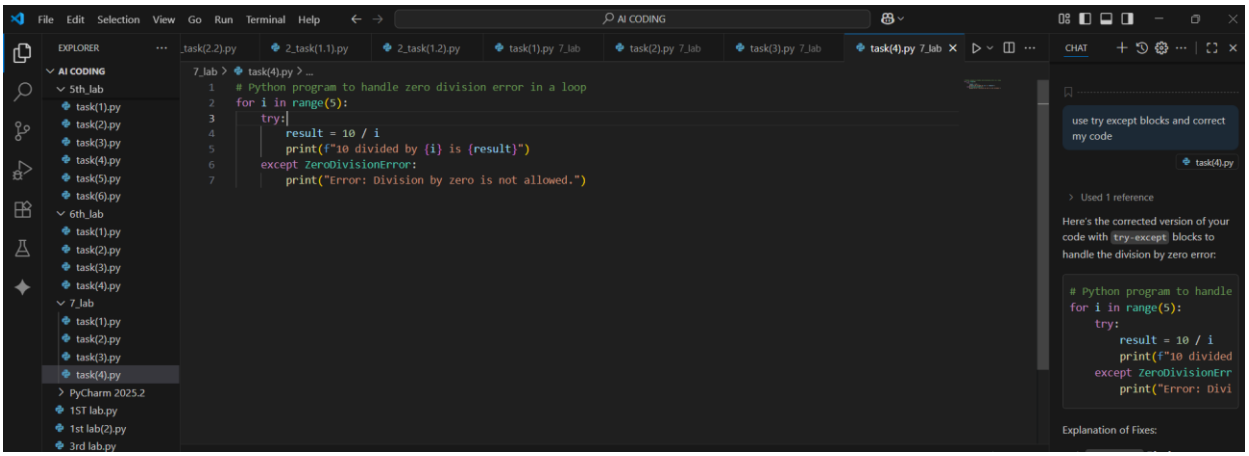
OBSERVATION: file sample.txt does not exist. This prevents the program from crashing. Uncommented fopen.close() and moved it to a finally block to ensure the file is closed properly, even if an error occurs.

Task 4:

Provide a piece of code with a ZeroDivisionError inside a loop. Ask AI to add error handling using try-except and continue execution safely

Prompt: use try except blocks and correct my code

Code:



OP:

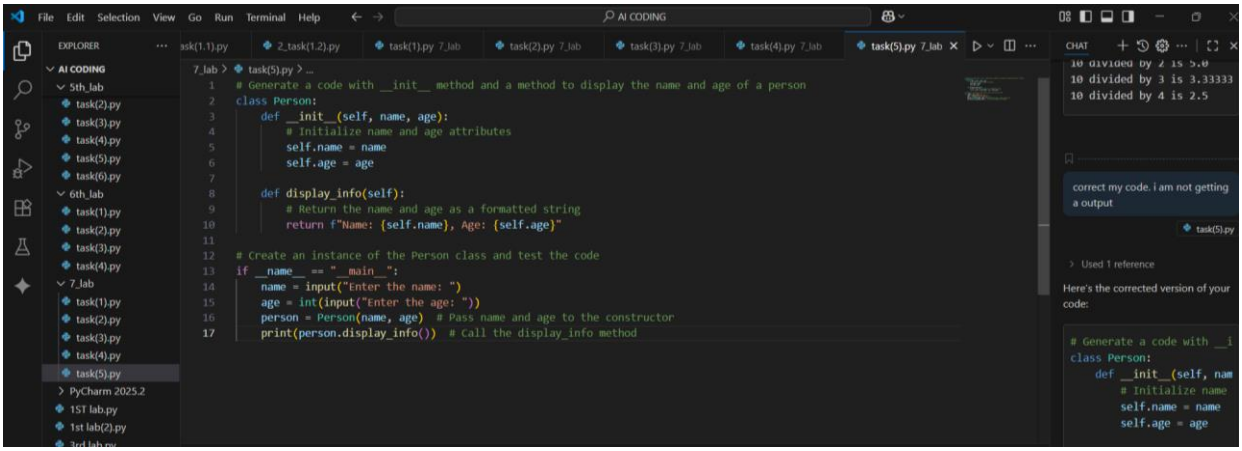
OBSERVATION: Wrapped the division operation 10 / i in a try block to catch any ZeroDivisionError. If i is 0, the except block is executed, and an error message is printed. The program continues execution even if a division by zero occurs, thanks to the try-except block.

Task 5:

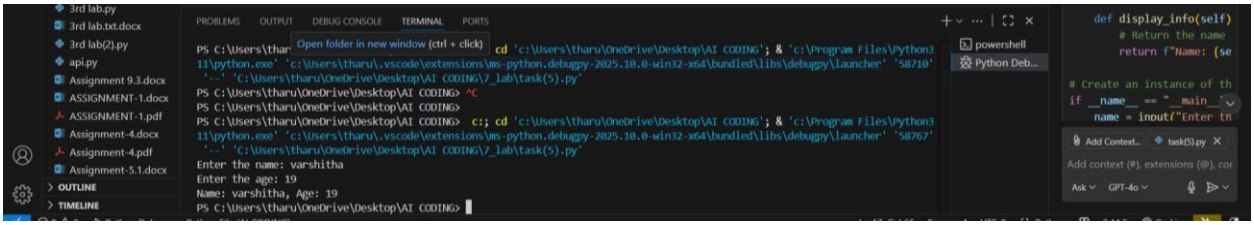
Include a buggy class definition with incorrect __init__ parameters or attribute references. Ask AI to analyze and correct the constructor and attribute usage

Prompt: correct my code. i am not getting a output

Code:



OP:



OBSERVATION: Added name and age as parameters to the __init__ method so that they can be passed when creating an instance of the person class. Added input prompts to allow the user to enter the name and age dynamically. Created an instance of the person class using the user-provided name and age. Called the display_info method to display the name and age.