AI ASSISTED CODING

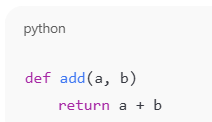
Assignment-7.3 :

**2403A52107**

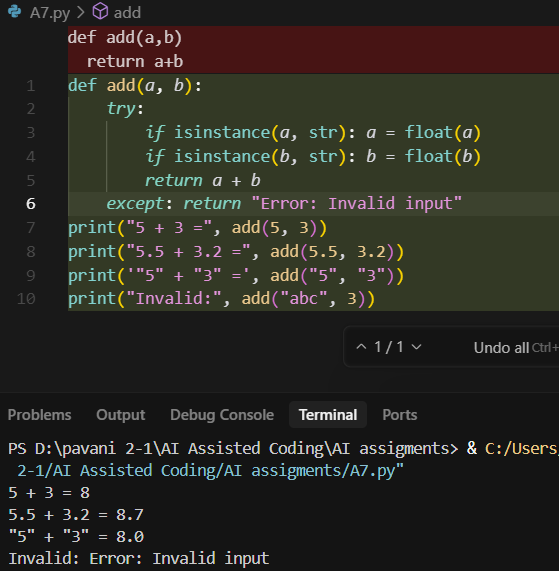
**Pavani Voddepalli**

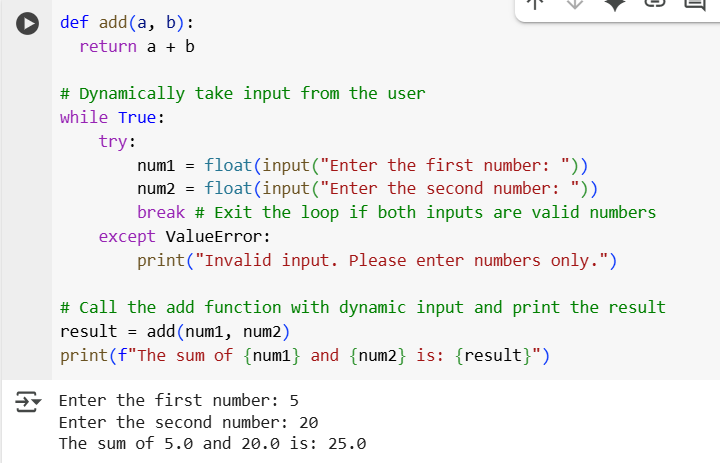
**Batch : 24BTCAIAI05**

**TASK-1 :** Analyze the ERROR and evaluates the addition of two numbers dynamically in Python

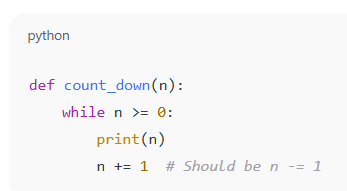


CODE & OUTPUT : FOR GEMINI AND CURSOR

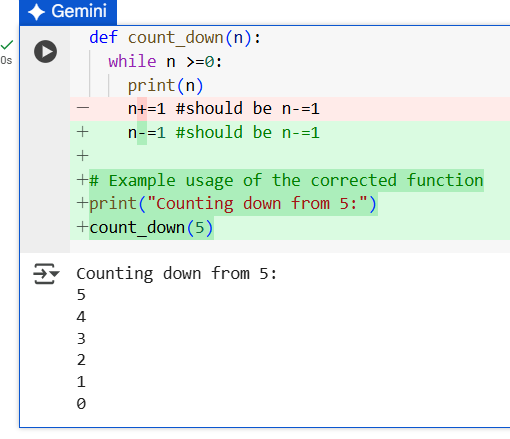


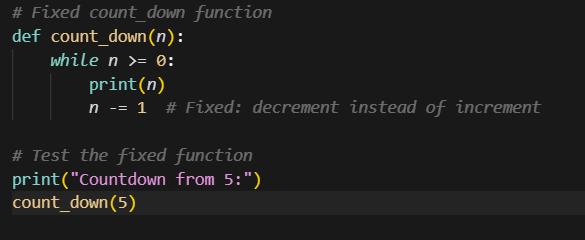


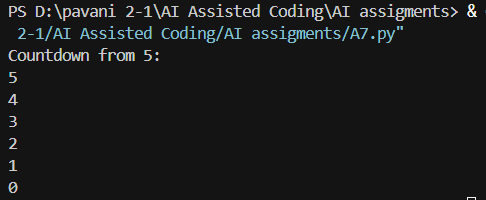
**TASK-2:** Fix a logic error in a loop that causes infinite iteration and evaluates the changes required modifying the code

****

CODE& OUTPUT :







EXPLANATION : This part defines a function called count\_down that takes one argument, n.

* while n >=0:: This is a loop that continues as long as the value of n is greater than or equal to 0.
* print(n): Inside the loop, the current value of n is printed.
* n-=1: This line is the fix for the infinite loop. It decreases the value of n by 1 in each step. This ensures that eventually, n will become less than 0, and the loop will stop. The comment #should be n-=1 indicates the intended correction.

# Example usage of the corrected function  
print("Counting down from 5:")  
count\_down(5)

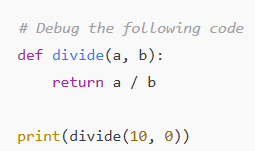
This section shows how to use the count\_down function.

* print("Counting down from 5:"): This prints a message to the console.
* count\_down(5): This calls the count\_down function with n set to 5, starting the countdown from 5.

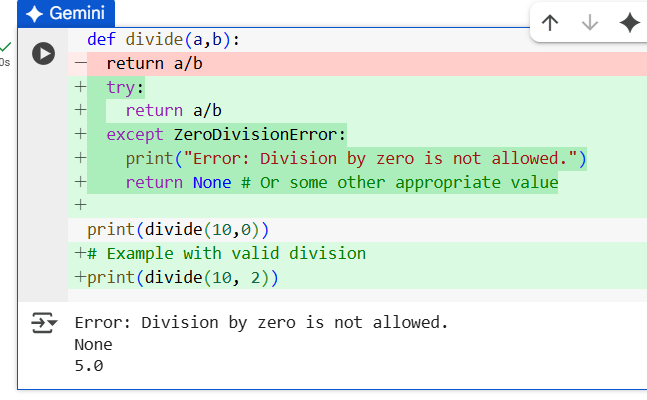
**The Logic Error:**

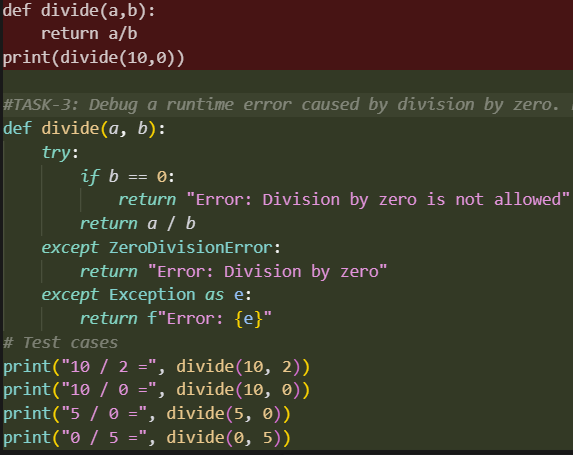
* The function is called count\_down but it's **incrementing** (n+=1) instead of **decrementing**
* Since n keeps getting bigger, the condition n >= 0 will always be true
* This creates an **infinite loop** that never ends

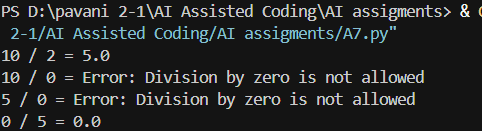
**TASK-3 :** Debug a runtime error caused by division by zero. Let AI insert try-except.

****

CODE & OUTPUT :





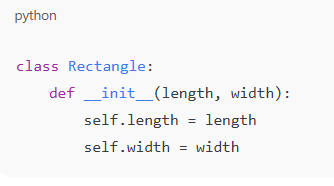


Explanation :

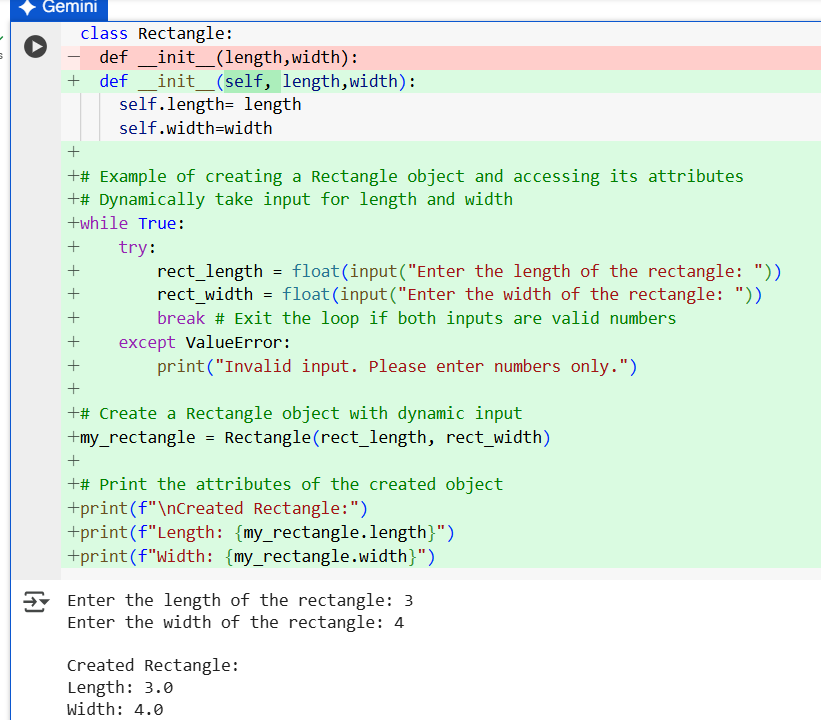
**The Runtime Error:**

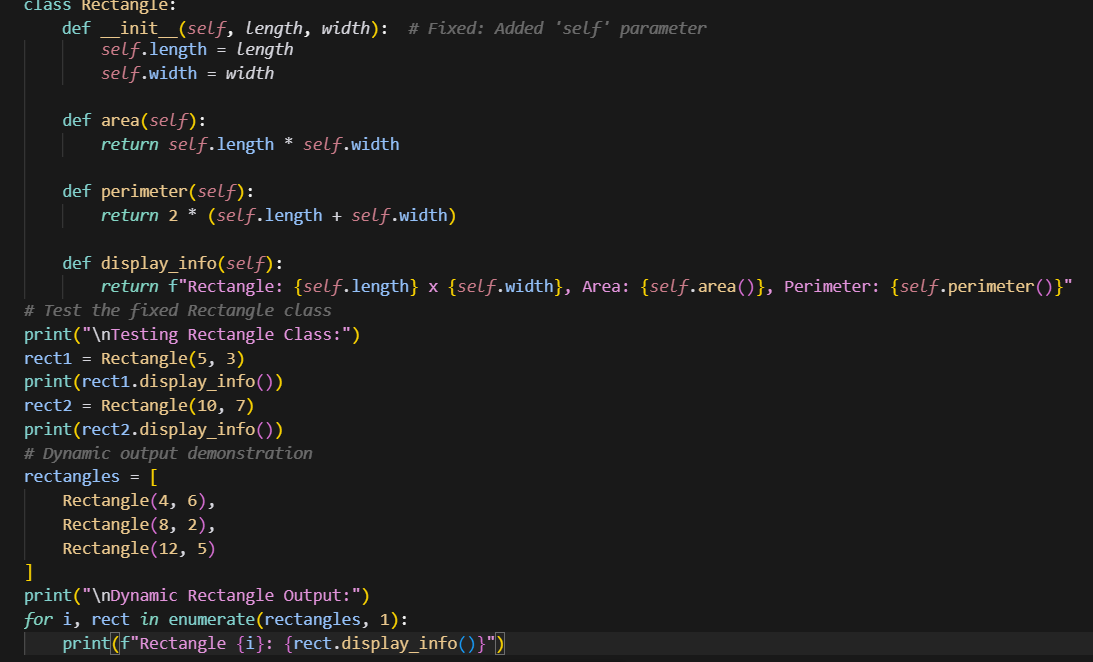
* When b = 0, division by zero occurs
* Python raises ZeroDivisionError: division by zero
* The program crashes without proper error handling

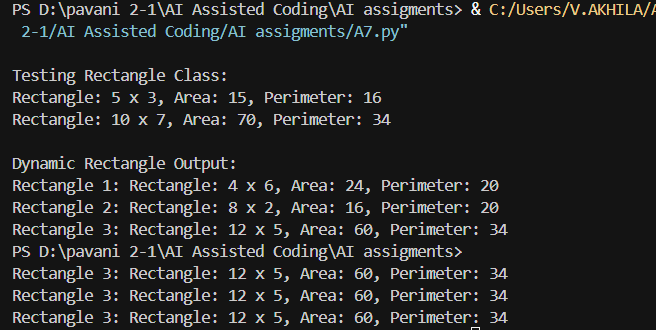
**TASK -4 :** Debug the python program , evaluate the error and modify the code with the appropriate dynamic output

****

CODE & OUTPUT :





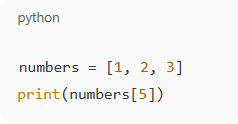


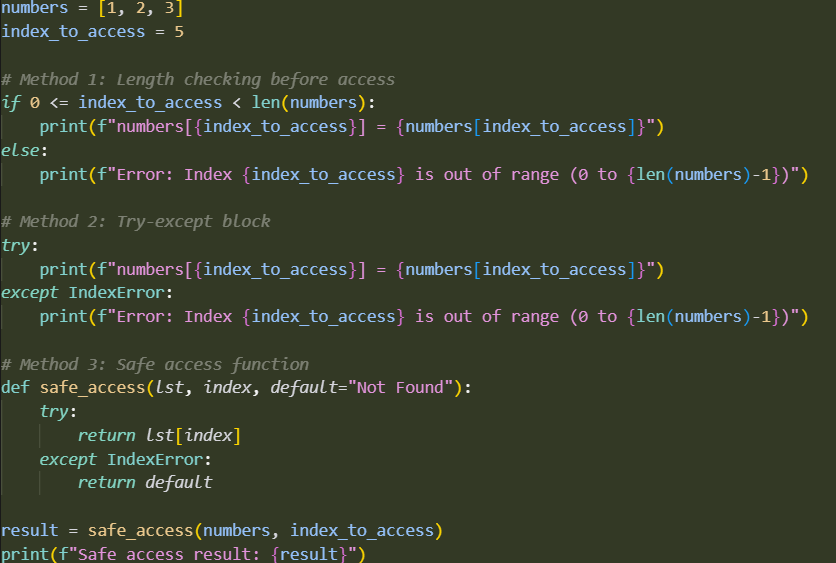
EXPLANATION :

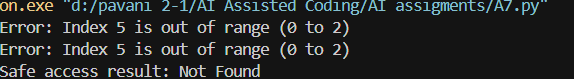
**The Error:**

* Missing self parameter in the \_\_init\_\_ method
* Python will raise TypeError: \_\_init\_\_() takes 2 positional arguments but 3 were given
* The self parameter is required for all instance methods in Python classes

**TASK-5 :**

Generate a python program recorrecting to resolve the access of an invalid list index checking length or using safe access logic





1. **IndexError**: The list numbers has only 3 elements (indices 0, 1, 2)
2. **Invalid Access**: Trying to access numbers[5] attempts to get the 6th element, which doesn't exist
3. **No Error Handling**: The code had no protection against accessing invalid indices