**AI ASSISTED CODING LAB-10.5**

**R. YASHRITHA REDDY  
2303A51785**

**Batch-12**

Task Description #1 – Variable Naming Issues Task: Use AI to improve unclear variable names.

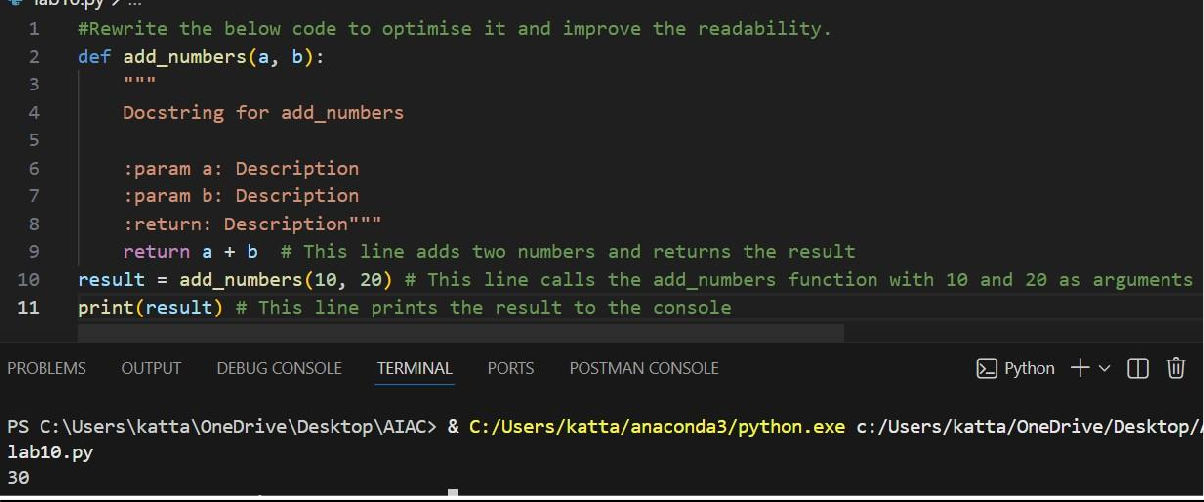
Sample Input Code:

def f(a, b): return a + b print(f(10, 20))

Expected Output:

* **Code rewritten with meaningful function and variable names. Prompt:**

#Rewrite the below code to optimise it and improve the readability. def f(a, b): return a + b print(f(10, 20))

Given Code and Output:

Task Description #2 – Missing Error Handling Task: Use AI to add proper error handling.

Sample Input Code:

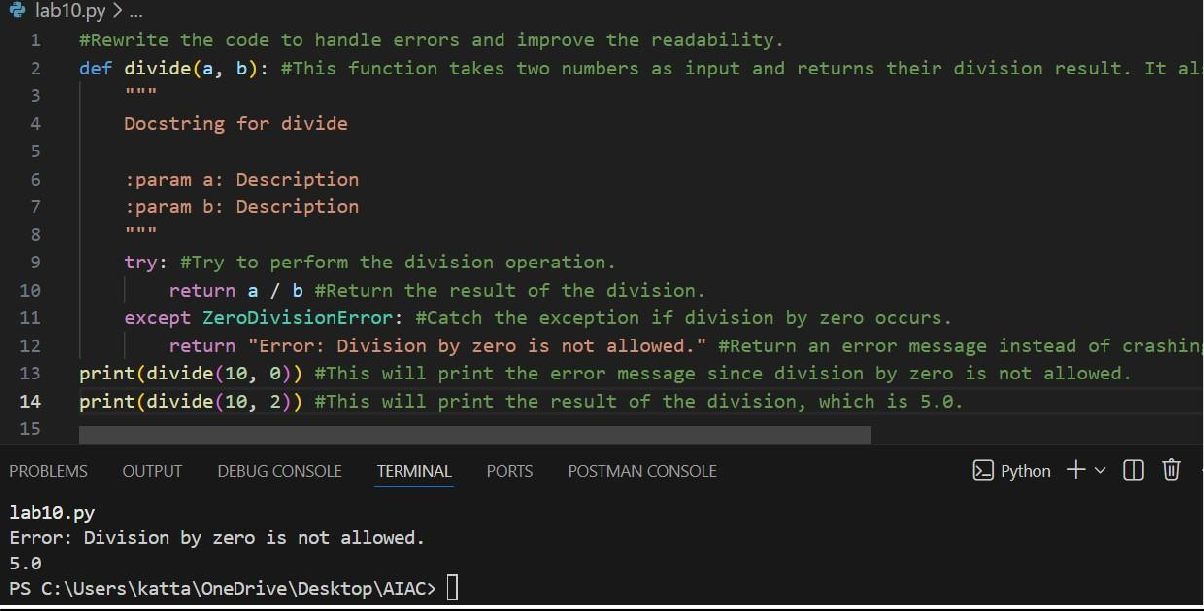
def divide(a, b): return a / b

print(divide(10, 0)) Expected Output:

* **Code with exception handling and clear error messages Prompt:**

#Rewrite the code to handle errors and improve the readability.

Given Code and Output:



Task Description #3: Student Marks Processing System

The following program calculates total, average, and grade of a student, but it has poor readability, style issues, and no error handling.

marks=[78,85,90,66,88]

t=0 for i in marks: t=t+i a=t/len(marks)

if a>=90: print("A") elif a>=75: print("B") elif

a>=60:

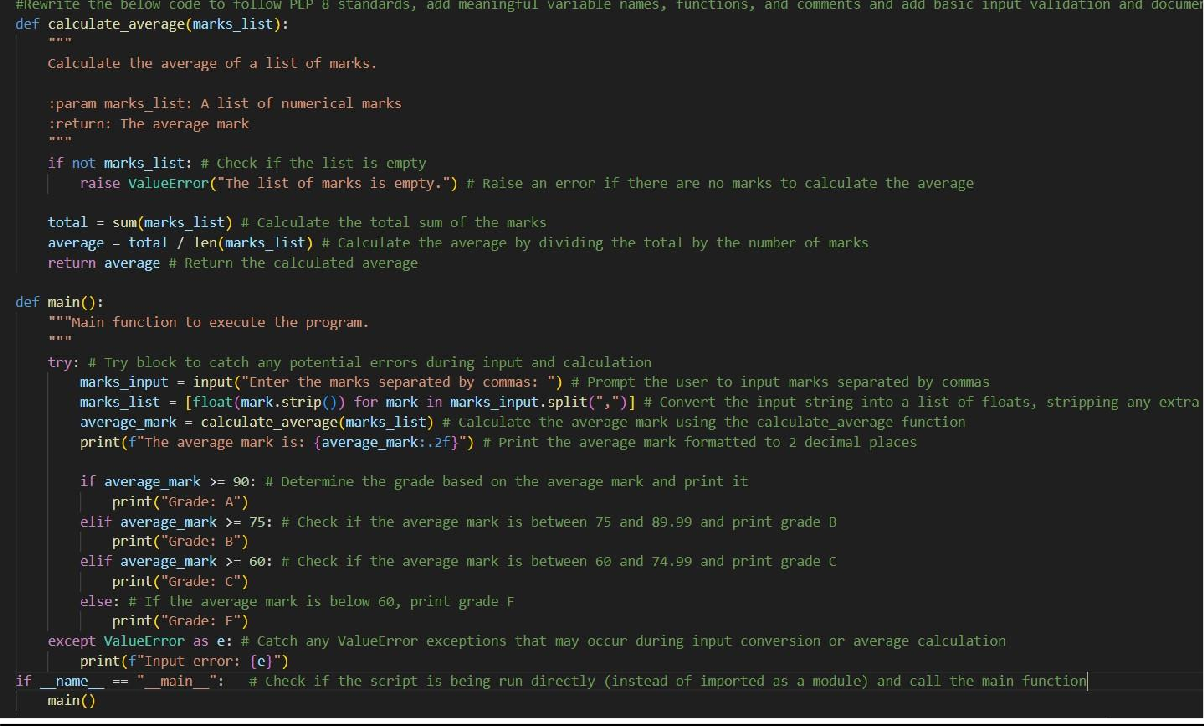
print("C") else:

print("F") Task:

* **Use AI to refactor the code to follow PEP 8 standards.**
* **Add meaningful variable names, functions, and comments.**
* **Add basic input validation and documentation. Prompt:**

#Rewrite the below code to follow PEP 8 standards, add meaningful variable names, functions, and comments and add basic input validation and documentation.

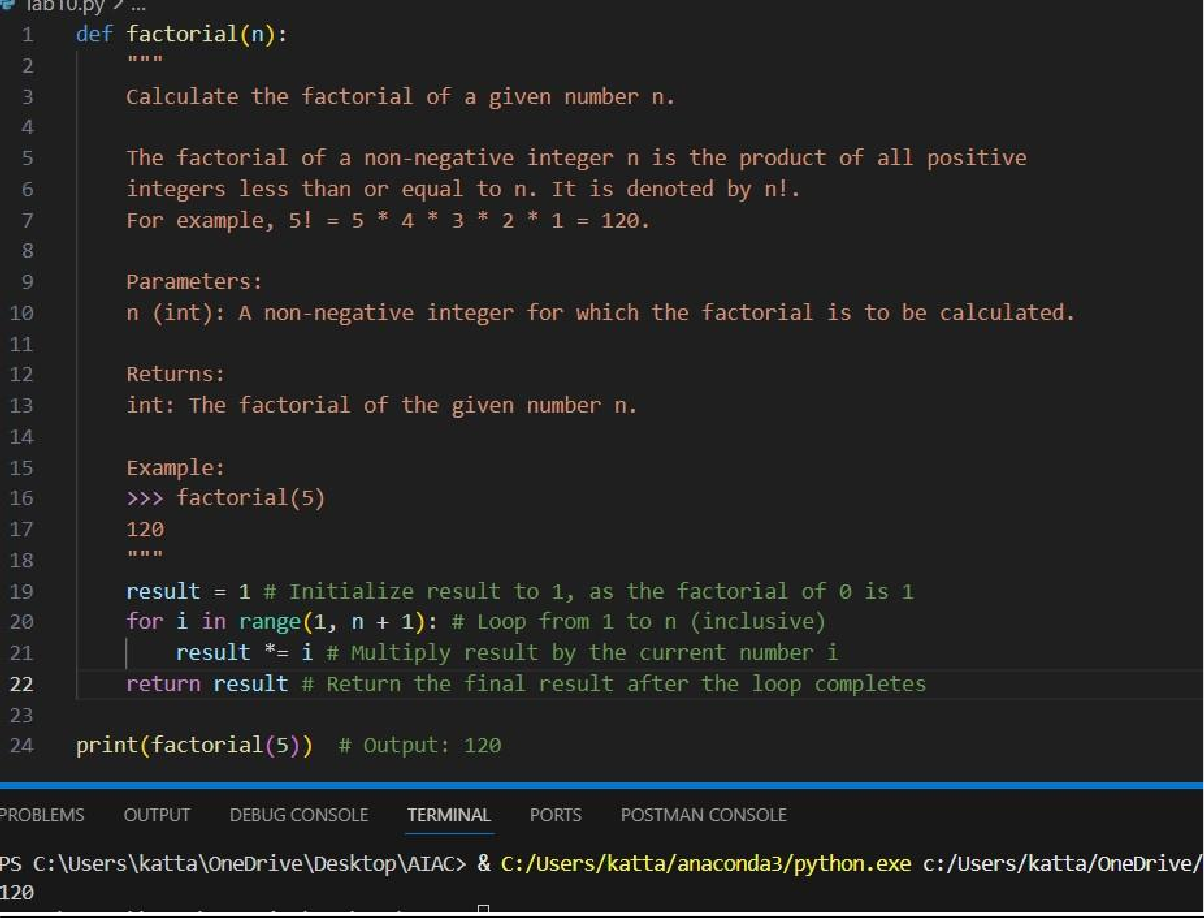
Given Code and Output:



Task Description #4: Use AI to add docstrings and inline comments to the following function. def factorial(n):

result = 1 for i in range(1,n+1):

result \*= i return result

Given Code and Output:

Task Description #5: Password Validation System (Enhanced) The following Python program validates a password using only a minimum length check, which is insufficient for real-world security requirements.

pwd = input("Enter password: ") if len(pwd) >= 8: print("Strong") else: print("Weak")

Task:

1. **Enhance password validation using AI assistance to include multiple security rules such as: o Minimum length requirement**
   * **Presence of at least one uppercase letter**
   * **Presence of at least one lowercase letter**
   * **Presence of at least one digit o Presence of at least one special character**
2. **Refactor the program to: o Use meaningful variable and function names o Follow PEP 8 coding standards o Include inline comments and a docstring**
3. **Analyze the improvements by comparing the original and AI- enhanced versions in terms of: o Code readability and structure o Maintainability and**

reusability o Security strength and robustness Prompt:

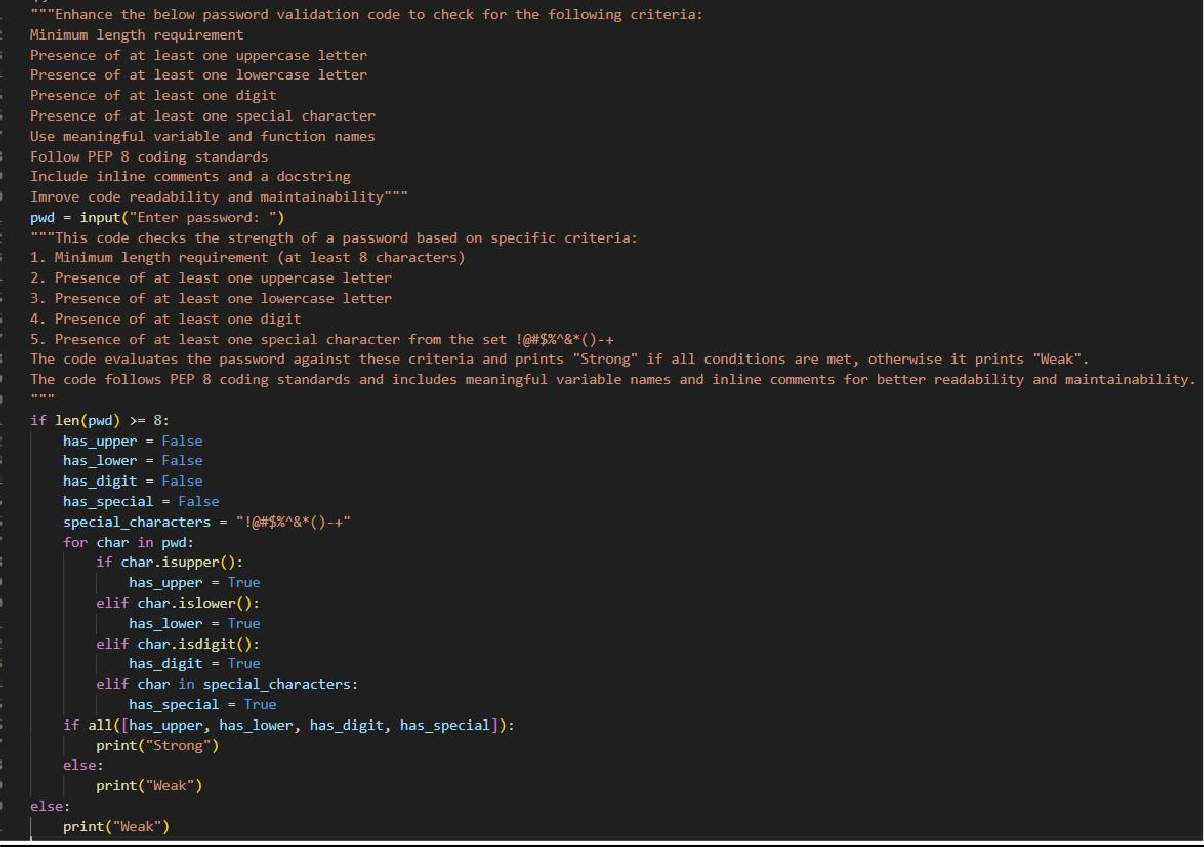
"""Enhance the below password validation code to check for the following criteria:

Minimum length requirement

Presence of at least one uppercase letter Presence of at least one lowercase letter Presence of at least one digit

Presence of at least one special character Use meaningful variable and function names Follow PEP 8 coding standards

Include inline comments and a docstring Imrove code readability and maintainability"""

Given Code and Output:

