Assignment-8.1

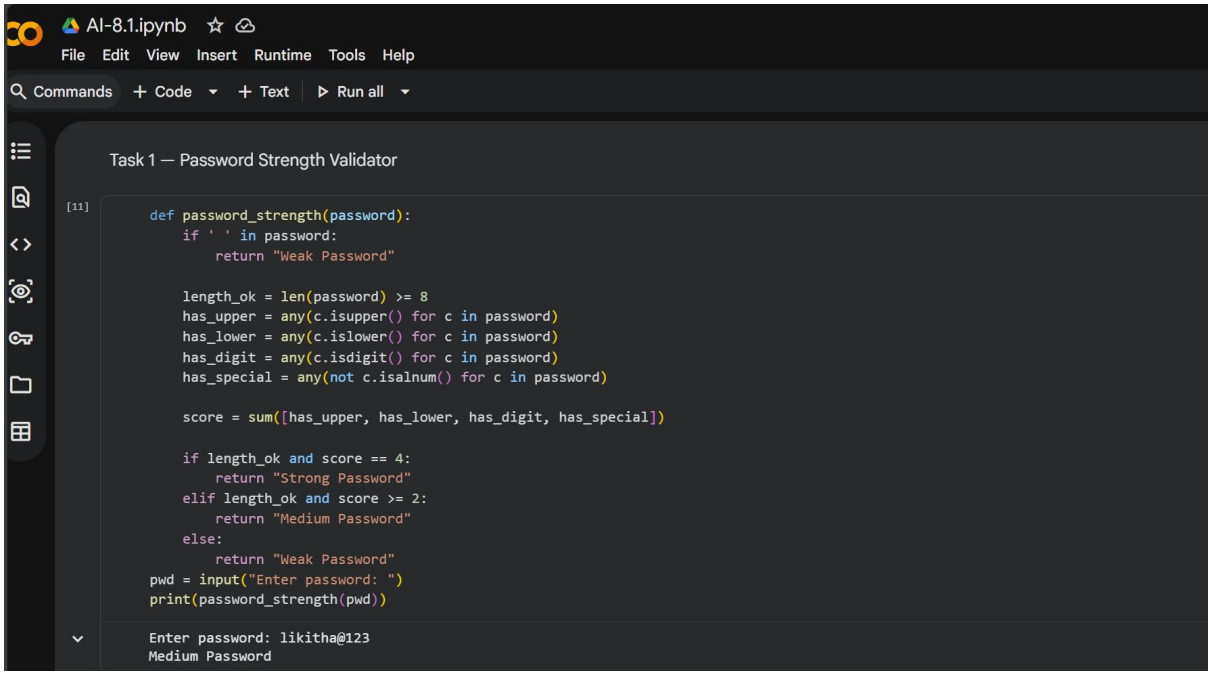
NAME:-pendli Likitha

HT NO:-2303A52393

BATCH:-43

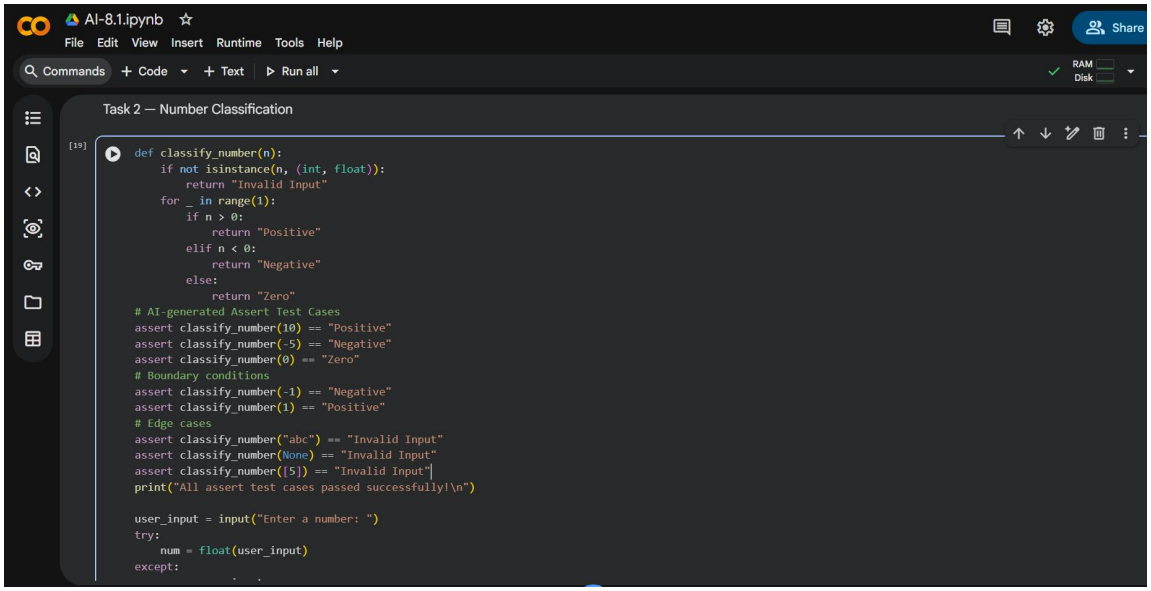
Prompt for Task 1 — Password Strength Validator

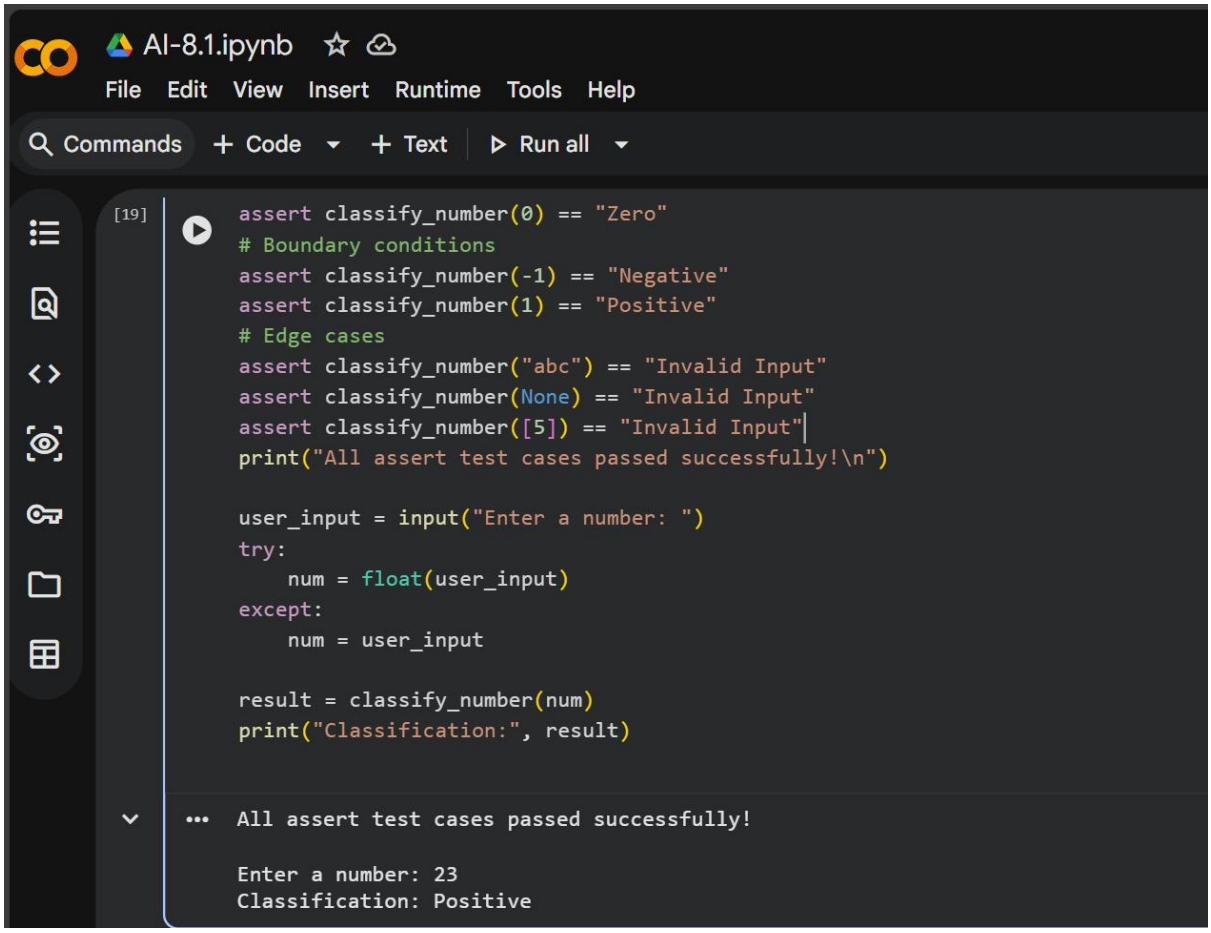
Create a Python function to classify passwords as Weak, Medium, or Strong. A strong password must be at least 8 characters long and include uppercase, lowercase, digit, and special character with no spaces. A medium password meets length and any 2–3 conditions.
Otherwise, it is weak. Generate at least 3 assert test cases and show input–output examples.



Prompt for Task 2 — Number Classification

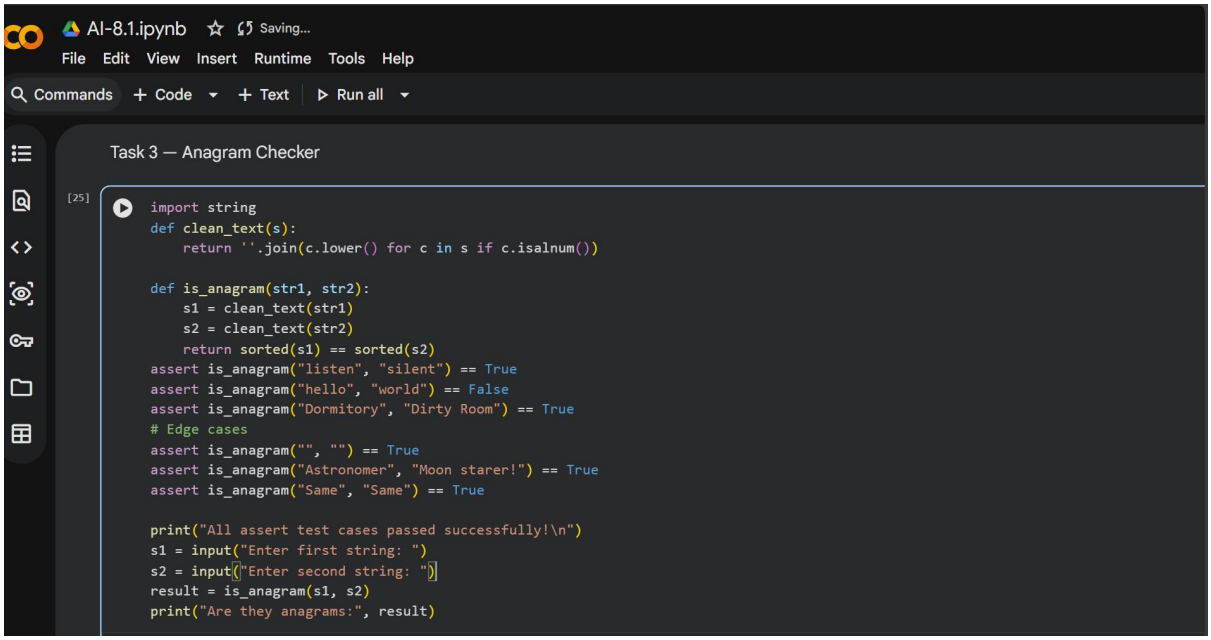
Write a Python function classify _number(n) using loops to classify a number as Positive, Negative, or Zero. Handle invalid inputs like strings and None by returning "Invalid Input". Include boundary conditions such as -1, 0, and 1.Generate at least 3 assert test cases. Also display output using user input.

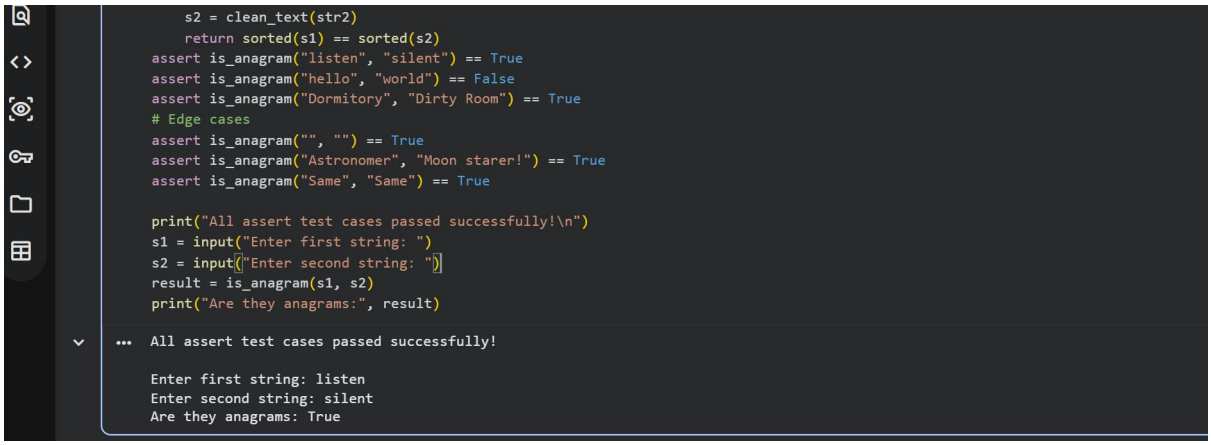

AI-8.1.ipynb
File Edit View Insert Runtime Tools Help
Commands + Code + Text Run all
Share
RAM
Disk
Task 2 — Number Classification
[19]
def classify_number(n):
if not isinstance(n, (int, float)):
return "Invalid Input"
for _ in range(1):
if n > 0:
return "Positive"
elif n < 0:
return "Negative"
else:
return "Zero"
AI-generated Assert Test Cases
assert classify_number(10) == "Positive"
assert classify_number(-5) == "Negative"
assert classify_number(0) == "Zero"
Boundary conditions
assert classify_number(-1) == "Negative"
assert classify_number(1) == "Positive"
Edge cases
assert classify_number("abc") == "Invalid Input"
assert classify_number(None) == "Invalid Input"
assert classify_number([5]) == "Invalid Input"
print("All assert test cases passed successfully!\n")
user_input = input("Enter a number: ")
try:
num = float(user_input)
except:


AI-8.1.ipynb
File Edit View Insert Runtime Tools Help
Commands + Code + Text Run all
[19]
assert classify_number(0) == "Zero"
Boundary conditions
assert classify_number(-1) == "Negative"
assert classify_number(1) == "Positive"
Edge cases
assert classify_number("abc") == "Invalid Input"
assert classify_number(None) == "Invalid Input"
assert classify_number([5]) == "Invalid Input"
print("All assert test cases passed successfully!\n")
user_input = input("Enter a number: ")
try:
num = float(user_input)
except:
num = user_input
result = classify_number(num)
print("Classification:", result)
All assert test cases passed successfully!
Enter a number: 23
Classification: Positive

Prompt for Task 3 — Anagram Checker

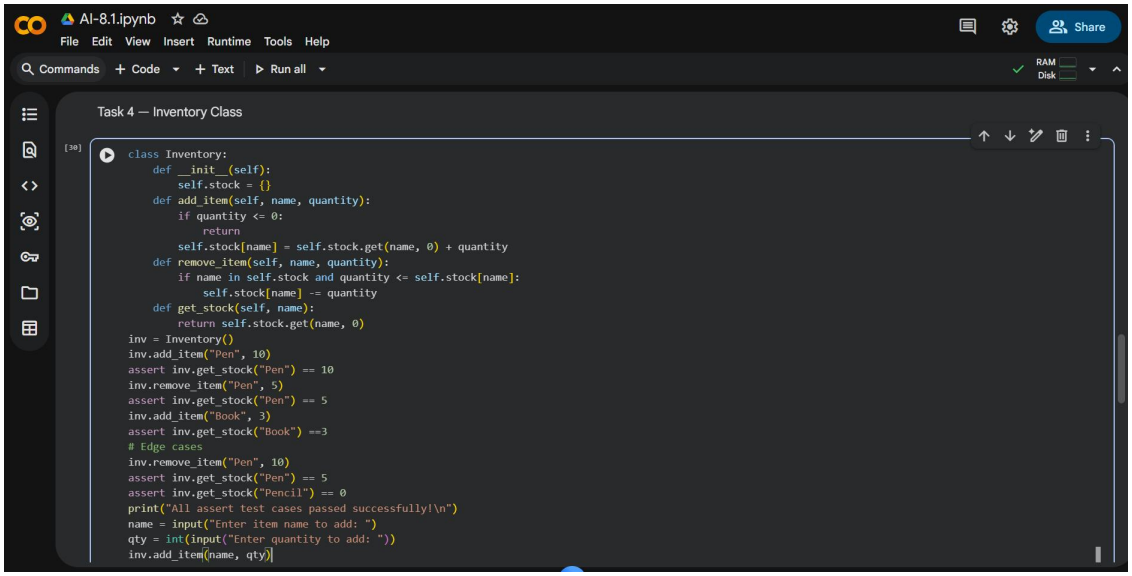
Create a Python function is_anagram(str1, str2) to check if two strings are anagrams. Ignore case, spaces, and punctuation while comparing. Handle edge cases like empty strings and identical words. Generate at least 3 assert test cases. Show input and output examples.





Prompt for Task 4 — Inventory Class

Design a Python class Inventory to manage stock items. Include methods: add_item(name, quantity), remove_item(name, quantity), and get_stock(name).Ensure stock never becomes negative and handle missing items. Generate at least 3 assert-based test cases .Demonstrate functionality with sample input and output.



Prompt for Task 5 — Date Validation & Formatting

Write a Python function validate_and_format_date(date_str) to validate dates in "MM/DD/YYYY" format. Convert valid dates to "YYYY-MM-DD" and return "Invalid Date" for wrong inputs. Handle invalid dates, formats, and leap year cases. Generate at least 3 assert test cases. Display results using user input.