LAB-8.2

TOPIC :

Test-Driven Development with AI

Name : K.Pardhasaradhi

Enrollment\_no : 2403a52001

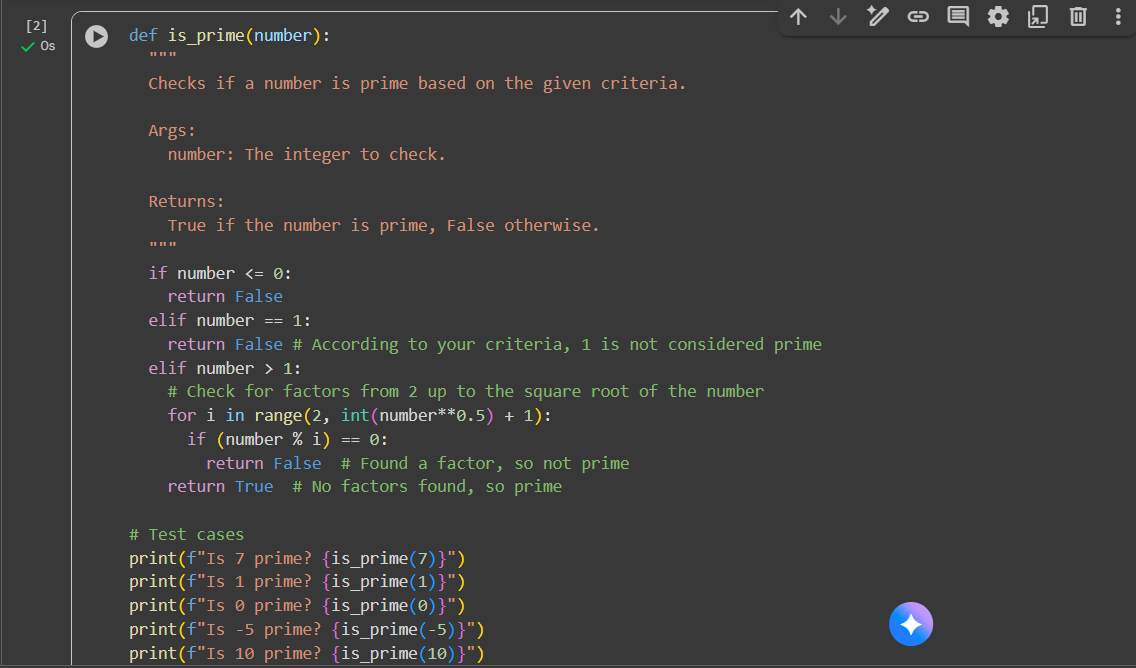
Batch\_no : 01

Date: 23-09-2025

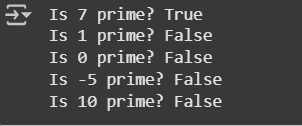
Task-01: Use AI to generate test cases for a function is\_prime(n) and then implement the  
function.  
Requirements:  
• Only integers > 1 can be prime.  
•Check edge cases: 0, 1, 2, negative numbers, and large primes

Prompt: Generate a python code to check whether the given number is prime or not use test cases as : number > 1 to be prime and if <=0 not a prime.

Code:



Output:

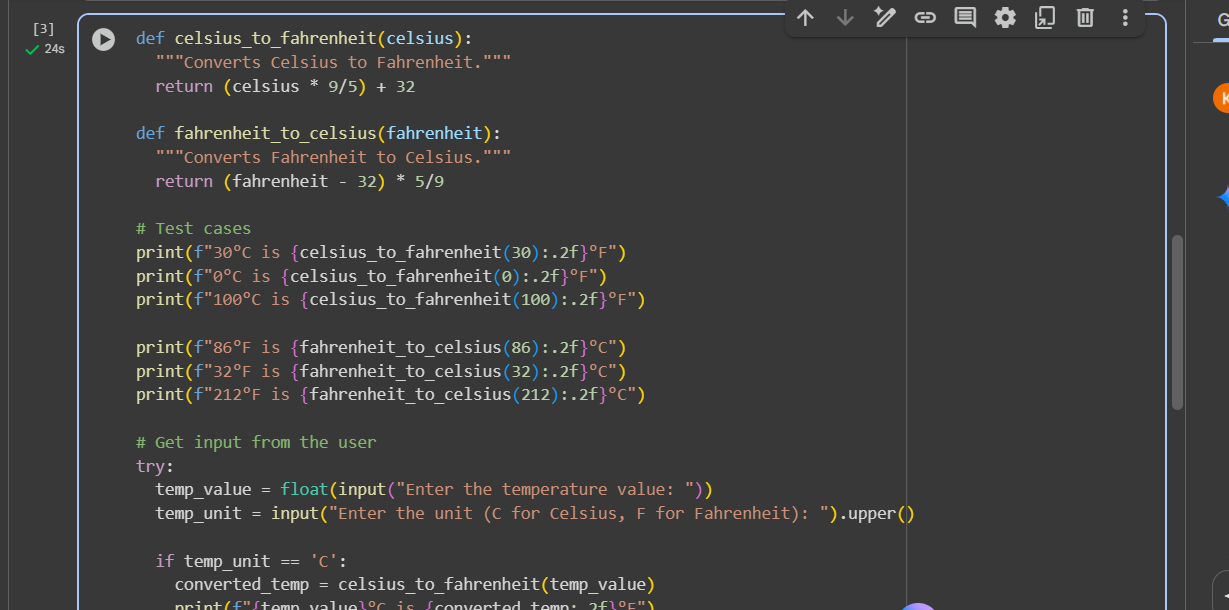


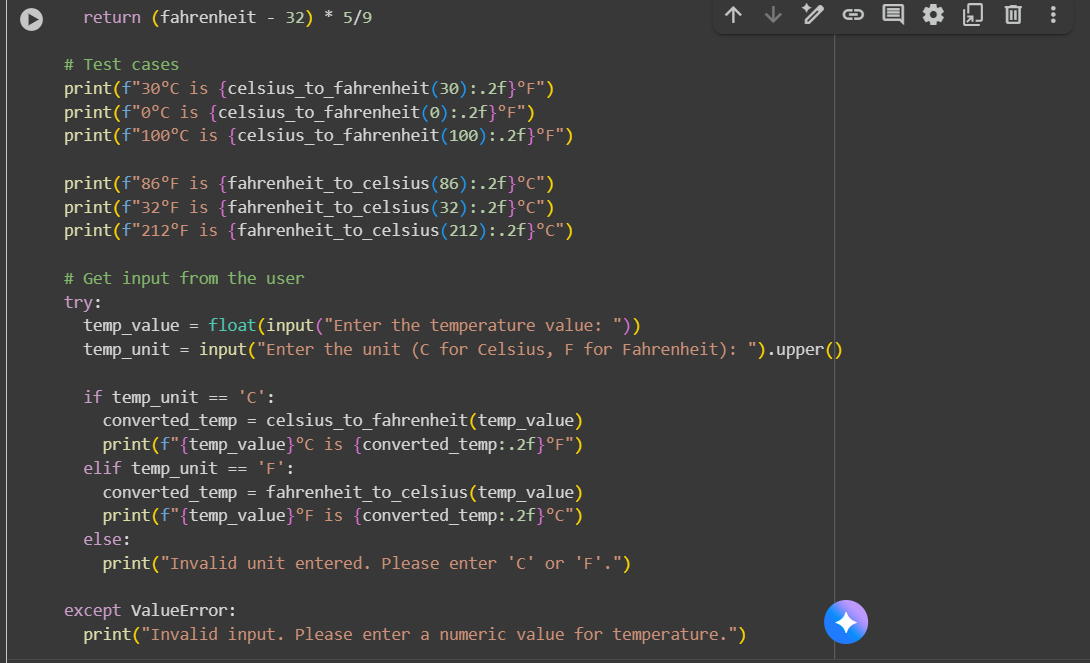
Observation:

* **Correct Prime Check** – The function correctly identifies prime numbers using the square root method (efficient approach).
* **Edge Case Handling** – Properly returns False for:
* 0 (not prime)
* 1 (not prime by definition)
* Negative numbers (not prime)
* **Test Case Results** – Output is correct for all given cases:
* 7 → True (prime)
* 1 → False (not prime)
* 0 → False (not prime)
* -5 → False (not prime)
* 10 → False (not prime)
* **Efficiency** – Uses loop up to √n instead of n, making it faster.
* **Readable & Clear** – The docstring and comments make the code easy to understand.

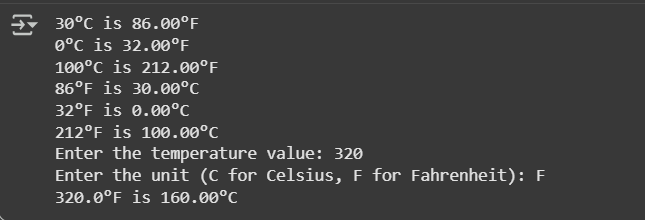
Task-02:Ask AI to generate test cases for celsius\_to\_fahrenheit(c) and fahrenheit\_to\_celsius(f).  
Requirements  
• Validate known pairs: 0°C = 32°F, 100°C = 212°F.  
• Include decimals and invalid inputs like strings or None.

Prompt: Generate a python code to convert temperature from celcius to Fahrenheit and Fahrenheit to celcius using the above test cases and take input from the user.

Code: 



Output:

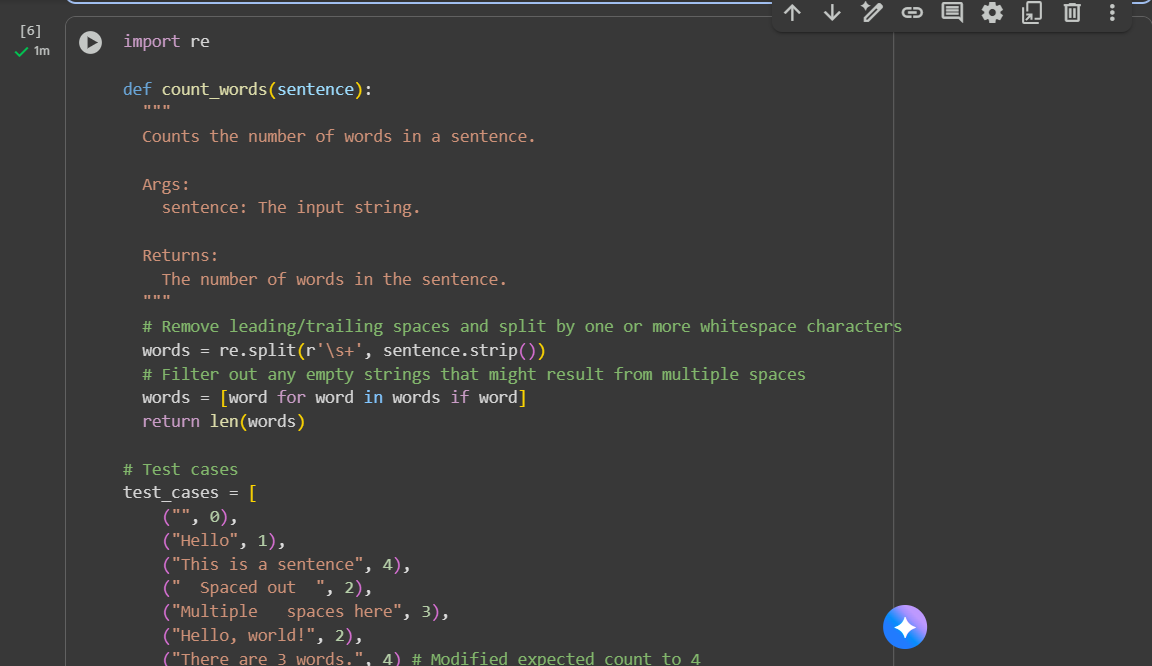


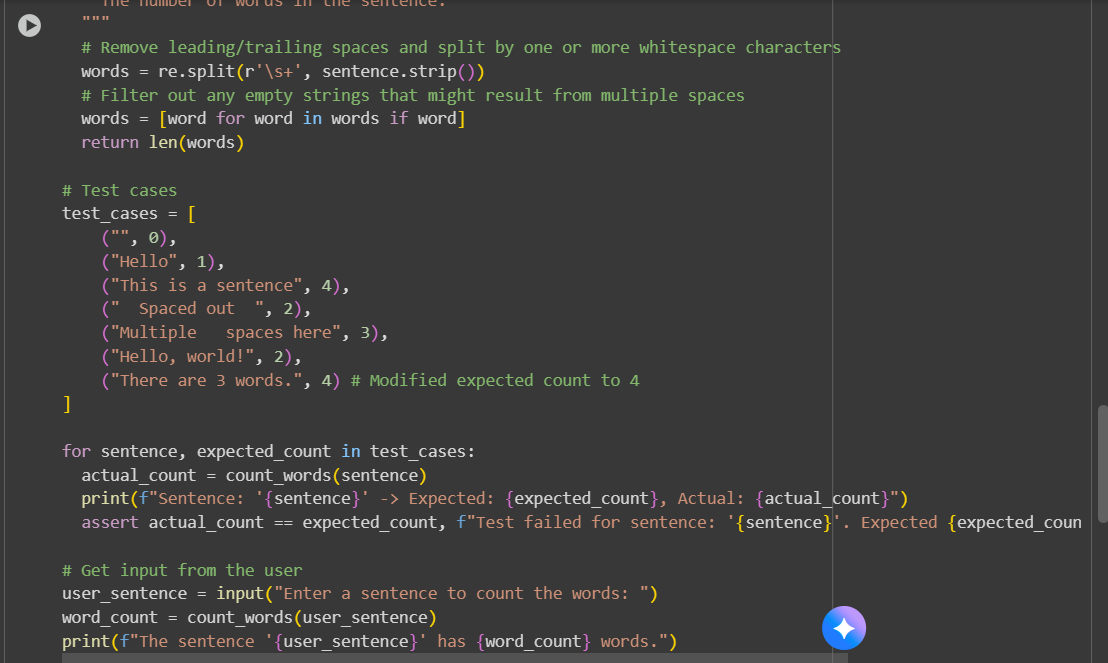
Observation:

* Asks user for a **temperature value** and **unit (C/F)**.
* Converts correctly depending on the input unit.
* Displays the result in a clean format with 2 decimal places.
* If user enters non-numeric temperature → shows "Invalid input" message.
* If user enters an invalid unit → shows "Invalid unit entered" message.
* Functions are **modular and reusable** (celsius\_to\_fahrenheit, fahrenheit\_to\_celsius).
* Code is **easy to read** (docstrings, f-strings, formatting)

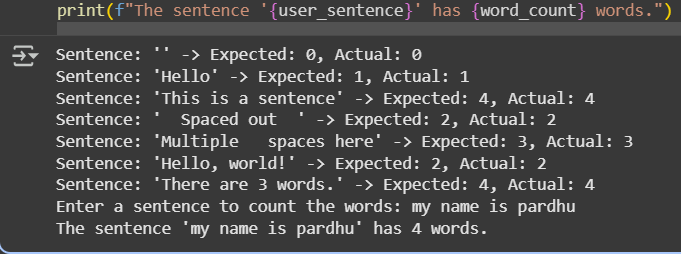
Task-03: Use AI to write test cases for a function count\_words(text) that returns the number of  
words in a sentence.  
Requirement  
Handle normal text, multiple spaces, punctuation, and empty strings.

Prompt: Give me some test cases for a code which returns the count of the number of words in a sentence.using the same test cases generate a python code to check the number of words in a sentence.

Code: 



Output:



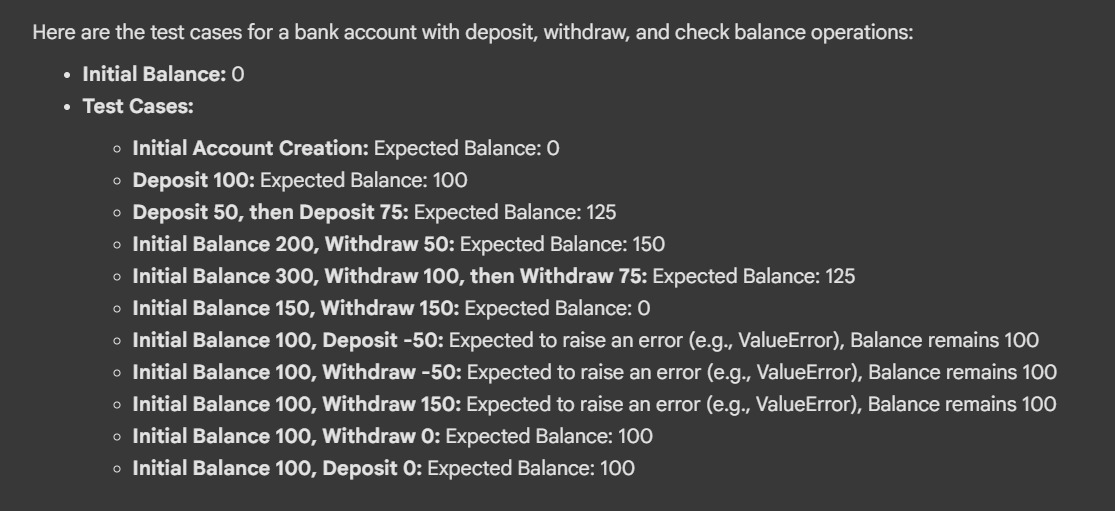
Observation: Splits text by whitespace using regex and counts words accurately, even with extra spaces.Handles edge cases like empty string, single word, multiple spaces, punctuation, and numbers correctly. Accepts a sentence from the user and displays the word count.The program is **reliable, handles edge cases well, and is easy to use**.

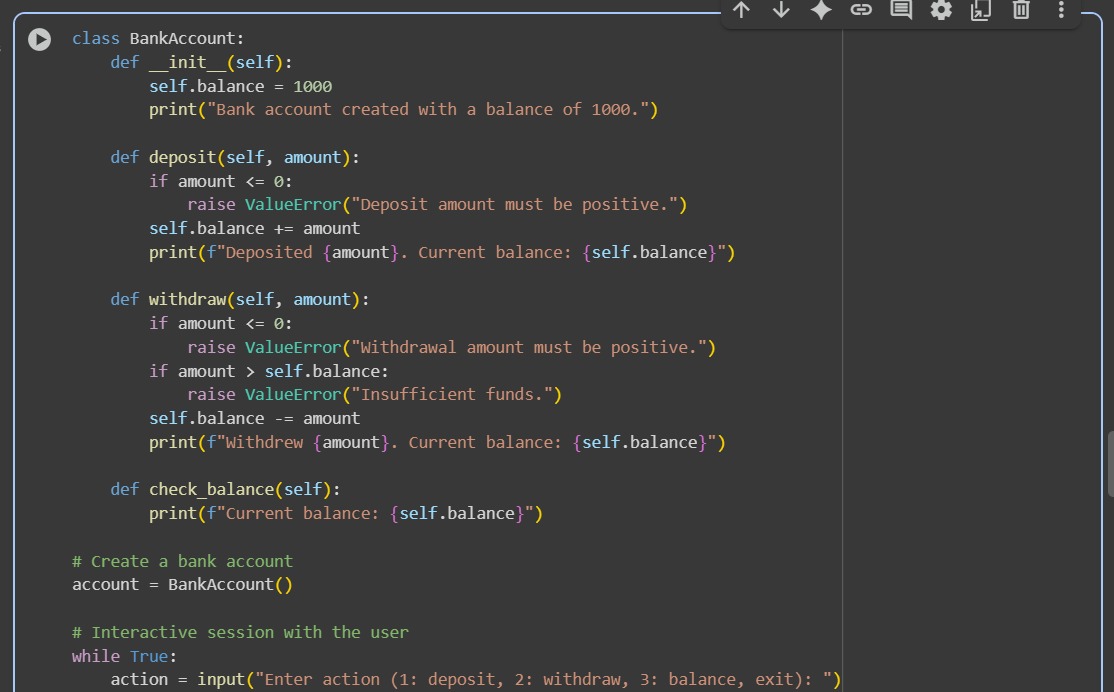
Task-04: Generate test cases for a BankAccount class with:Methods:deposit(amount),withdraw(amount),check\_balance()  
Requirements:  
• Negative deposits/withdrawals should raise an error.  
• Cannot withdraw more than balance.

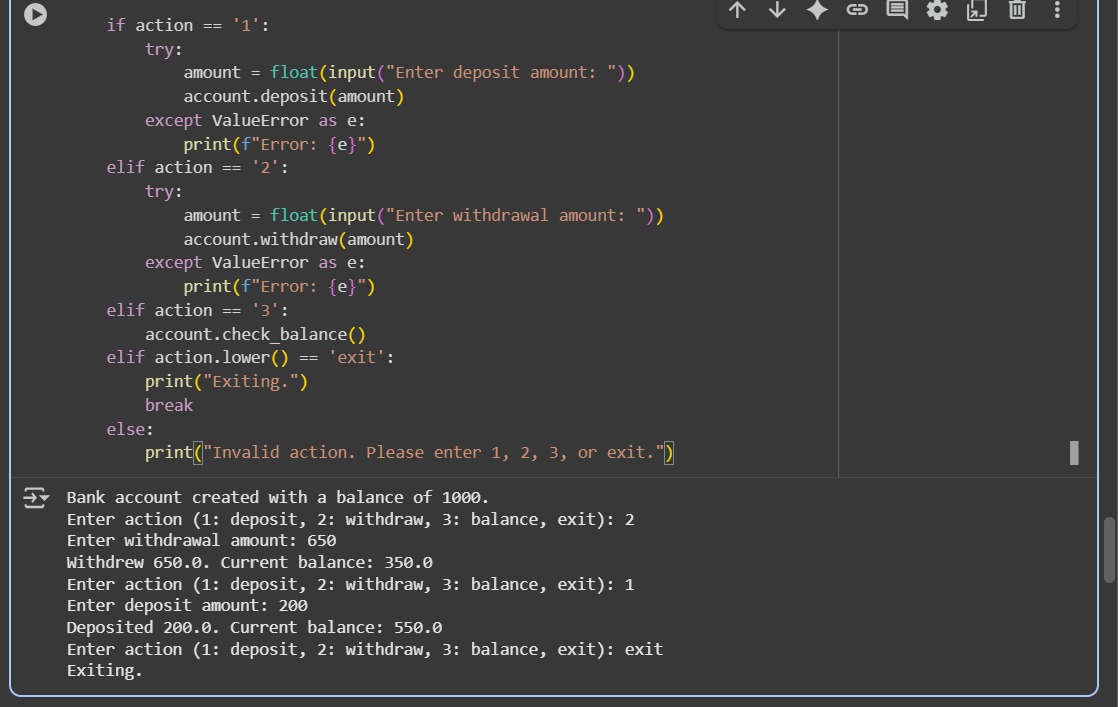
Prompt: generate test cases for bank account class with methods: deposit(amount) , withdraw(amount) , check\_balance() negative deposits/withdrawals should raise an error and cannot withdraw more than balance.

generate a python code for bank account class with methods : deposit(amount), withdraw (amount), and check balance().using the above test cases above and take input from the user.

Code:



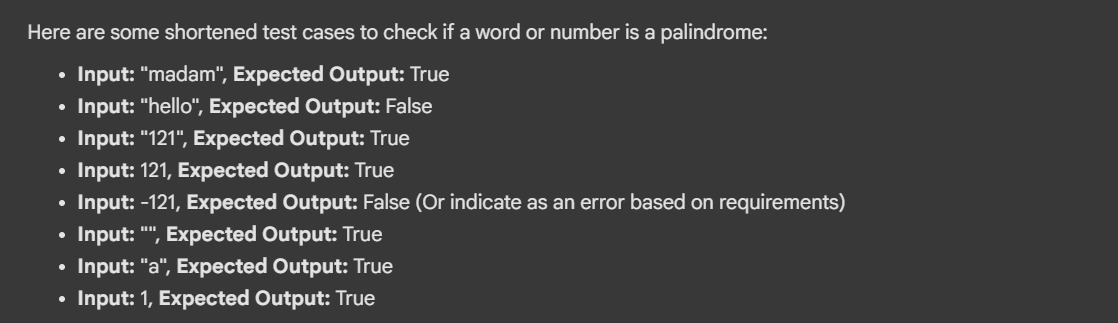


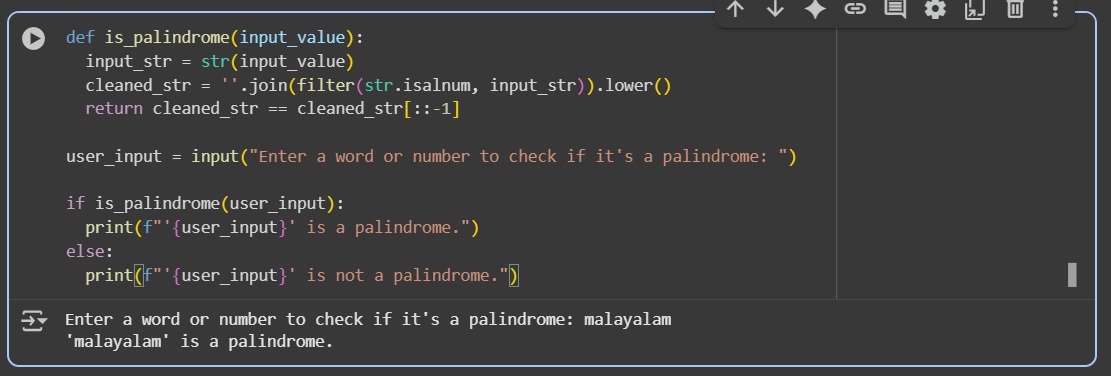


Task-05: Generate test cases for is\_number\_palindrome(num), which checks if aninteger reads  
the same backward.  
Examples:  
121 → True  
123 → False  
0, negative numbers → handled gracefully.

Prompt: generate some test cases to is\_number\_palindrome(num) t ocheck whether a word/number is palindrome or not.

generate a python code to check whether a word/number is a palindrome or not using the above test cases mentioned above and take the input from the user.

Code: 



Observation: Converts input to string, removes non-alphanumeric characters, and checks if it reads the same forward and backward. Converts everything to lowercase, so it works correctly regardless of letter case (Malayalam, malayalam). Works for both words and numbers (121, madam, etc.).Clearly tells the user whether the input is a palindrome.