COURSE   :   AI ASSISTED CODING

NAME   : M. VARSHITH REDDY

HT.NO   :   2403A51289

BATCH   :   01

Task Description#1

• Zero-shot: Prompt AI with only the instruction — Write a Python function to

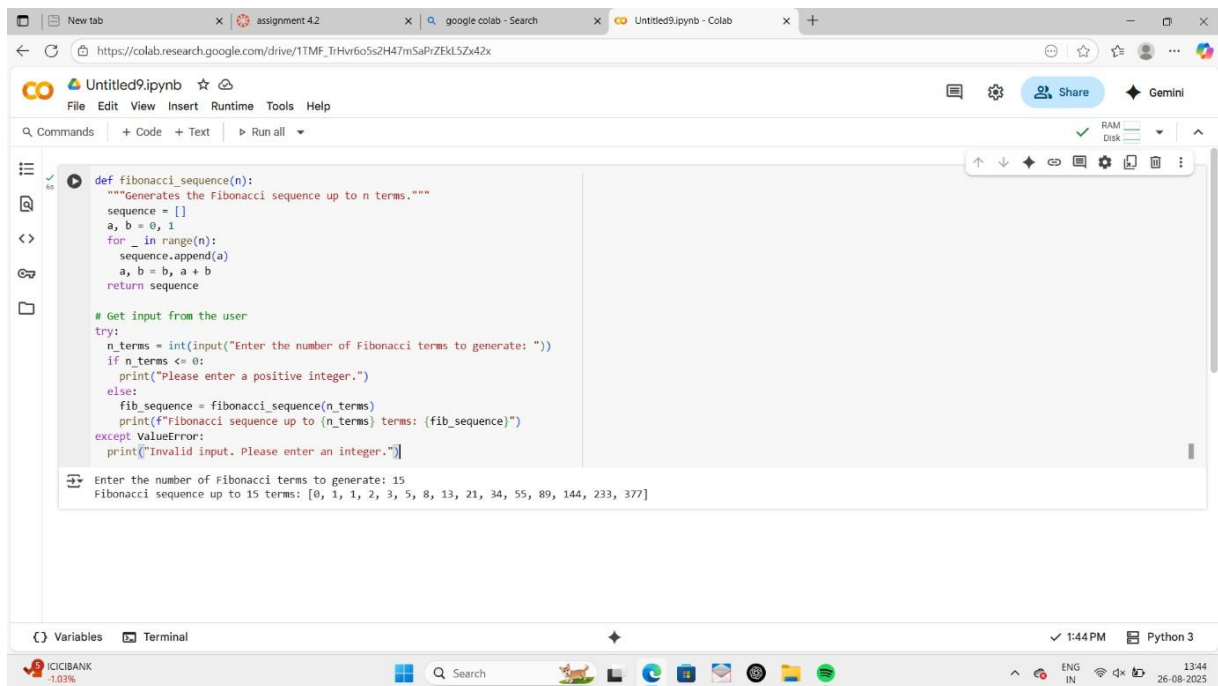generate the Fibonacci sequence up to n terms

PROMPT:

write a code in Python function to generate the Fibonacci sequence up to n term with input is given by user.

Expected Output#1

• A working function without using any sample inputs/outputs

CODE :


```python
def fibonacci_sequence(n):
    """Generates the Fibonacci sequence up to n terms."""
    sequence = []
    a, b = 0, 1
    for _ in range(n):
        sequence.append(a)
        a, b = b, a + b
    return sequence

# Get input from the user
try:
    n_terms = int(input("Enter the number of Fibonacci terms to generate: "))
    if n_terms <= 0:
        print("Please enter a positive integer.")
    else:
        fib_sequence = fibonacci_sequence(n_terms)
        print(f"Fibonacci sequence up to {n_terms} terms: {fib_sequence}")
except ValueError:
    print("Invalid input. Please enter an integer.")
```

```
Enter the number of Fibonacci terms to generate: 15
Fibonacci sequence up to 15 terms: [0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377]
```

OBSERVATION:

generate_fibonacci_sequence that generates the Fibonacci sequence up to a specified number of terms. It then prompts the user to enter the desired number of terms, handles potential invalid input (non-integer or negative numbers), and prints the resulting Fibonacci sequence

Task Description#2

• One-shot: Provide one example: Input: 100, Output: 37.78 to help AI generate a function that converts Fahrenheit to Celsius.
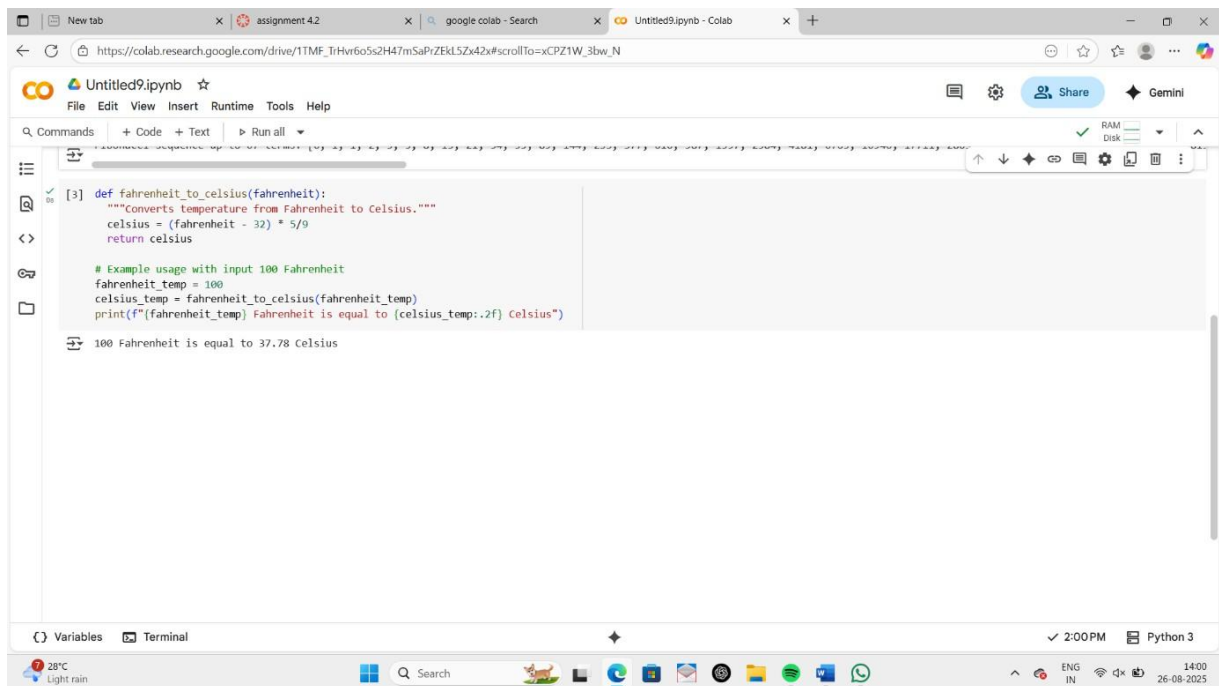
Expected Output#2

• A correct conversion function guided by the single example

PROMPT:

Generate a function that converts Fahrenheit to Celsius take input as 100 and output as 37.78

CODE:

OBSERVATION:

This code defines a function to convert Fahrenheit to Celsius and prompts the user for the temperature in Fahrenheit.

Task Description#3

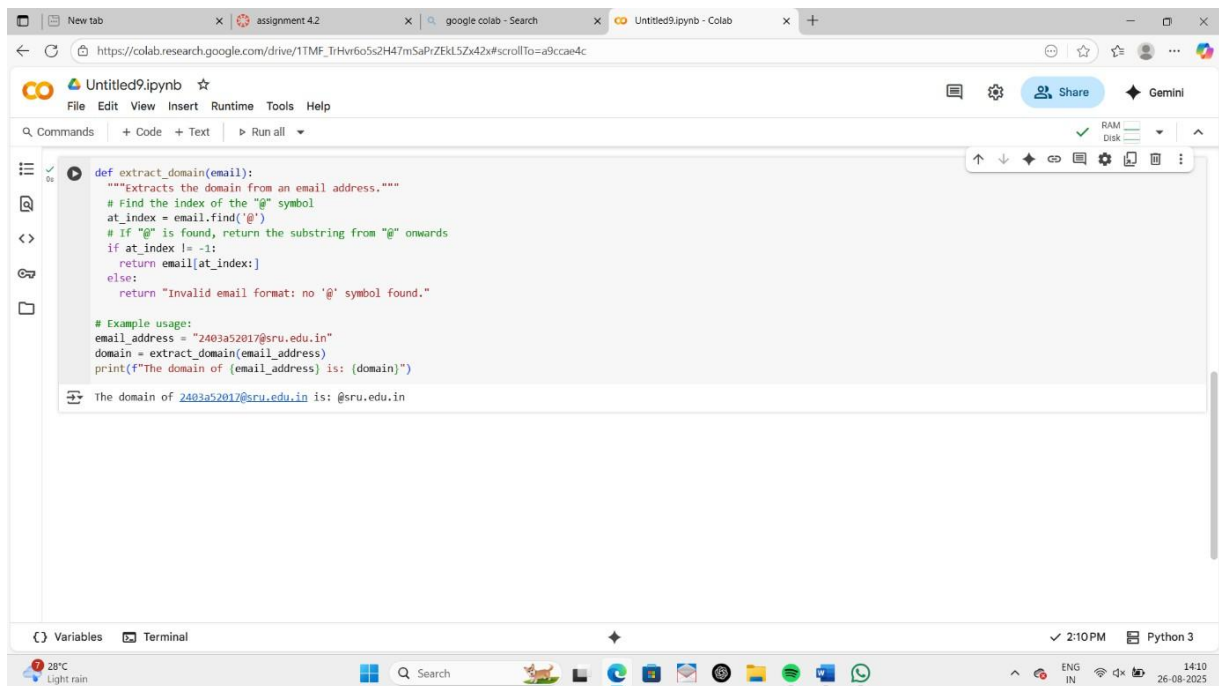• Few-shot: Give 2–3 examples to create a function that extracts the domain name from an email address.
Expected Output#3
• Accurate function that returns only the domain portion of an email (e.g.,@gmail.com)

Prompt:

Generate a code to extract the domain from the email address take input as 2403a52017@sru.edu.in and output=@sru.edu.in

Code:

OBSERVATION:

The code in cell defines function extract_domain that takes an email address as input. It attempts to split the email address at the "@" symbol and returns the part after the "@" with a "@" prefix to represent the domain. It also includes error handling using a try-except block to catch IndexError if the input is not a valid email format (i.e., it doesn't contain an "@" symbol), in which case it returns "Invalid email format".

Task Description#4

• Compare zero-shot vs few-shot prompting for generating a function that checks

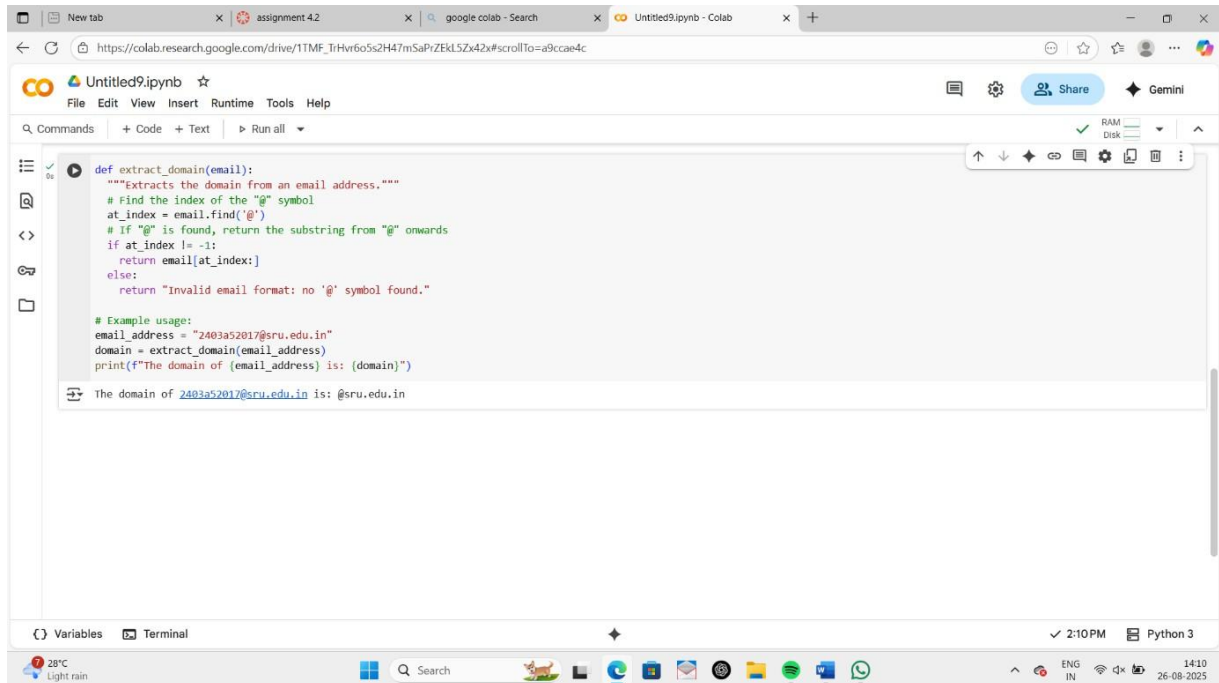whether a word is a palindrome, ignoring punctuation and case.

Expected Output#4

• Output comparison + student explanation on how examples helped the mode

PROMPT:

ZEROSHOT:

generate a function that checks whether a word is a palindrome, ignoring punctuation and case? with user input?

Code

Few shot:

Prompt:

Generate a code to check whether input is palindrome or not by user given input is mom and output as mom

Code:

Observation

"Looking at palindrome functions again, I see how the examples provided helped clarify the requirements. The examples like mom specifically showed that punctuation and case should be ignored. If only a simple example like 'level' was given, the code might not have included the steps to remove punctuation and convert to lowercase. This shows how giving a few varied examples (few-

shot) is really useful for making sure the code handles different situations correctly, compared to just one simple example (one-shot)."

Task Description#5

• Use few-shot prompting with 3 sample inputs to generate a function that determines the maximum of three numbers without using the built-in max() function.
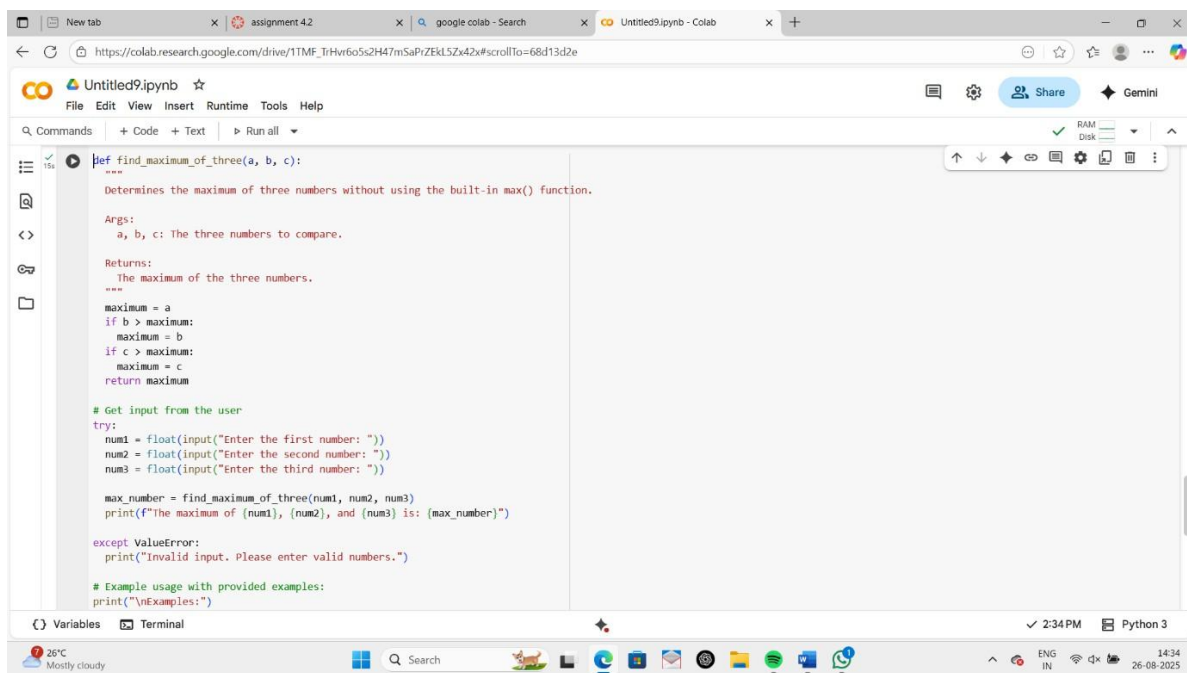
Expected Output#5

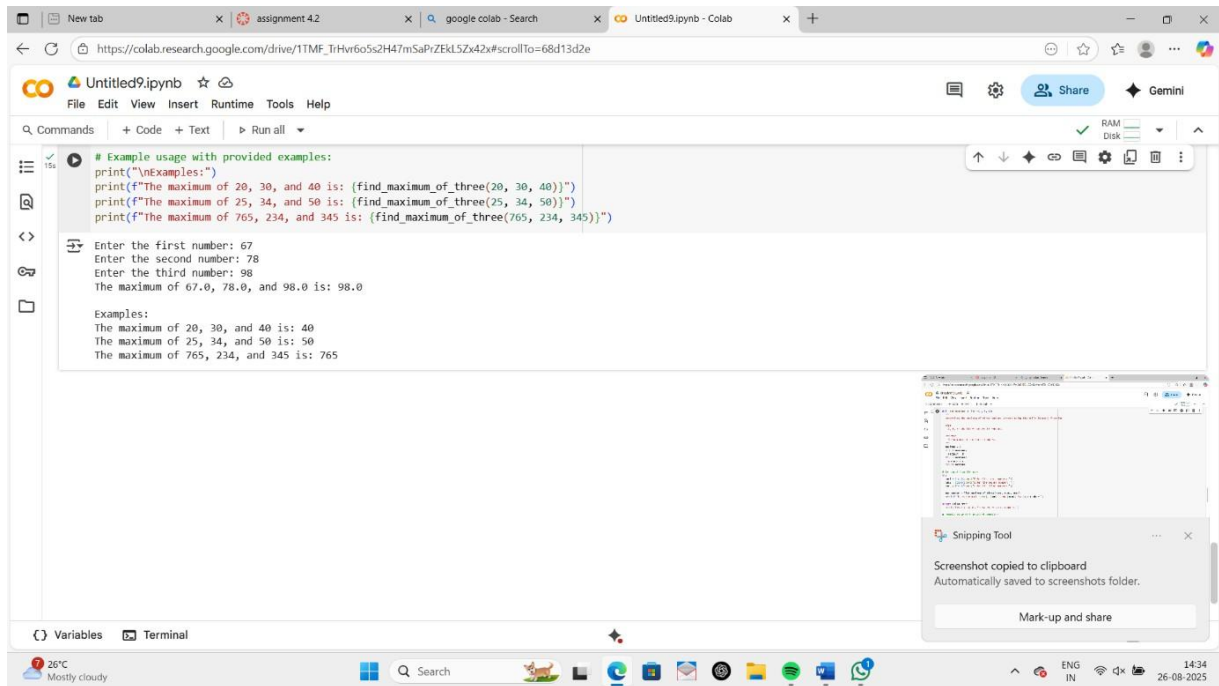• A function that handles all cases with correct logic based on example patterns.

Prompt

create a python function that determines the maximum of three numbers without using the built-in max() function with user given input? ex1: a=20,b=30,c=40;output=40 ex2:a=25,b=34,c=50;output=50 ex3:a=765,b=234,c=345;output=765?

Code

```
# Example usage with provided examples:
print("\nExamples:")
print(f"The maximum of 20, 30, and 40 is: {find_maximum_of_three(20, 30, 40)}")
print(f"The maximum of 25, 34, and 50 is: {find_maximum_of_three(25, 34, 50)}")
print(f"The maximum of 765, 234, and 345 is: {find_maximum_of_three(765, 234, 345)}")
```

```
Enter the first number: 67
Enter the second number: 78
Enter the third number: 98
The maximum of 67.0, 78.0, and 98.0 is: 98.0

Examples:
The maximum of 20, 30, and 40 is: 40
The maximum of 25, 34, and 50 is: 50
The maximum of 765, 234, and 345 is: 765
```

Observation

The code in cell defines a function find_maximum_of_three that takes three numbers as input. It initializes a variable maximum with the value of the first number. Then, it compares the second number with the current maximum and updates maximum if the second number is greater. Finally, it compares the third number with the current maximum and updates it again if the third number is greater. The function returns the final maximum value. The code also includes a part that prompts the user for three numbers and calls this function to find and print the maximum, with error handling for non-numeric input.