

NAME :- KAUSHAL KUMAR.

HALLTICKET NO :-2403A51317

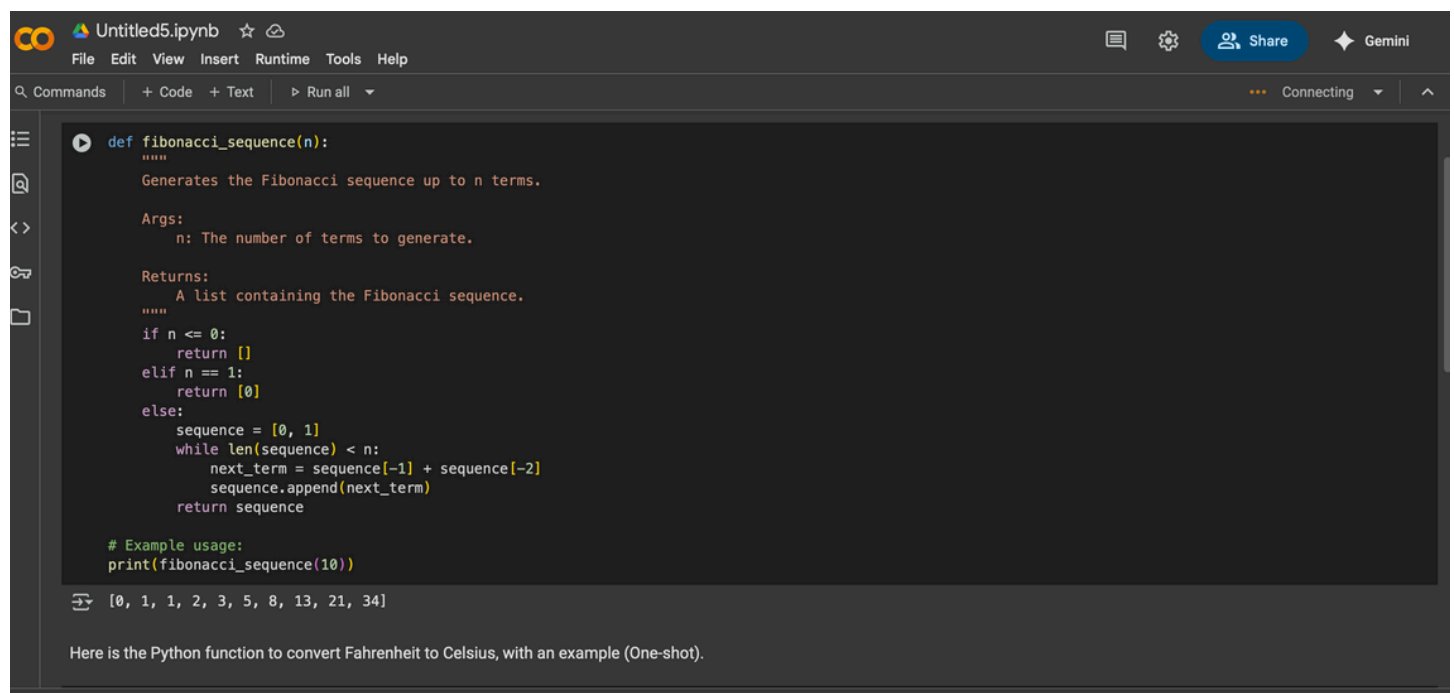
ASSIGNMENT :-4.2

Task1:

Zero-shot: Prompt AI with only the instruction — Write a Python function to generate the Fibonacci sequence up to n terms

Expected Output#1

- **A working function without using any sample inputs/outputs.**



The screenshot shows a Jupyter Notebook titled 'Untitled5.ipynb'. The code cell contains a Python function `def fibonacci_sequence(n):` with a docstring that describes its purpose and arguments. The function logic includes a base case for `n <= 0` returning an empty list, a base case for `n == 1` returning `[0]`, and a loop for `n > 1` that builds the sequence iteratively. Below the function, an example usage is shown: `print(fibonacci_sequence(10))`. The output cell displays the list `[0, 1, 1, 2, 3, 5, 8, 13, 21, 34]`. At the bottom of the notebook, there is a text prompt: 'Here is the Python function to convert Fahrenheit to Celsius, with an example (One-shot).'

```
def fibonacci_sequence(n):
    """
    Generates the Fibonacci sequence up to n terms.

    Args:
        n: The number of terms to generate.

    Returns:
        A list containing the Fibonacci sequence.
    """
    if n <= 0:
        return []
    elif n == 1:
        return [0]
    else:
        sequence = [0, 1]
        while len(sequence) < n:
            next_term = sequence[-1] + sequence[-2]
            sequence.append(next_term)
        return sequence

# Example usage:
print(fibonacci_sequence(10))
```

[0, 1, 1, 2, 3, 5, 8, 13, 21, 34]

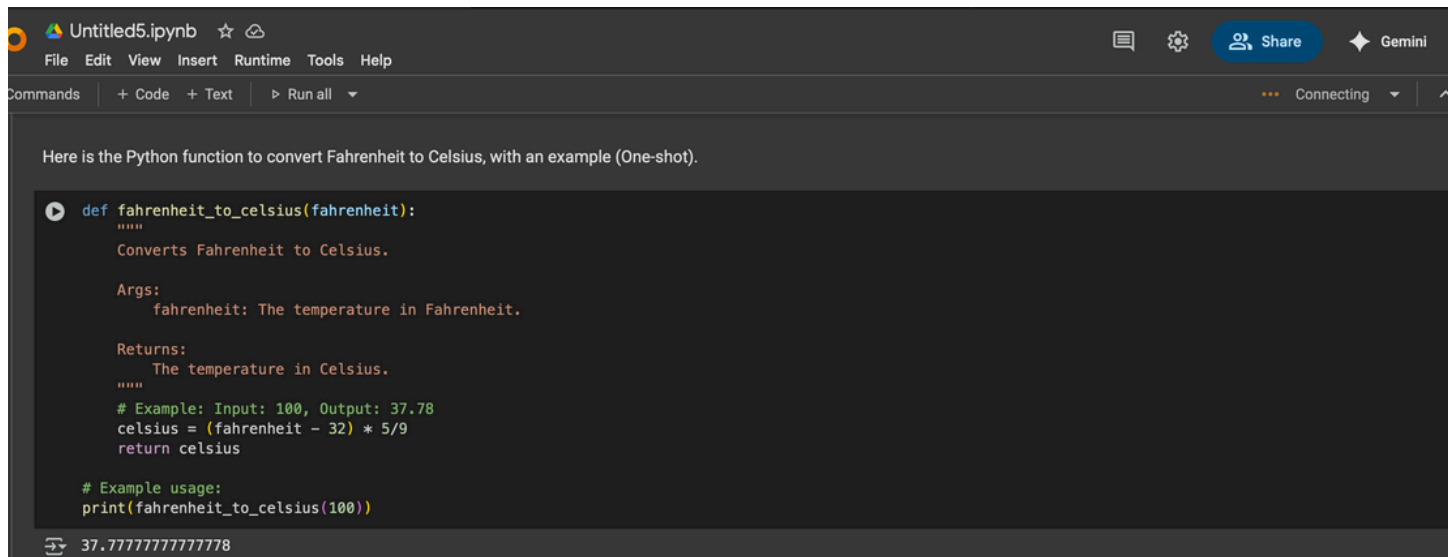
Here is the Python function to convert Fahrenheit to Celsius, with an example (One-shot).

TASK 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.**



Untitled5.ipynb ☆ ☁

File Edit View Insert Runtime Tools Help

Commands + Code + Text ▶ Run all

Connecting

Here is the Python function to convert Fahrenheit to Celsius, with an example (One-shot).

```
def fahrenheit_to_celsius(fahrenheit):  
    """  
    Converts Fahrenheit to Celsius.  
  
    Args:  
        fahrenheit: The temperature in Fahrenheit.  
  
    Returns:  
        The temperature in Celsius.  
    """  
    # Example: Input: 100, Output: 37.78  
    celsius = (fahrenheit - 32) * 5/9  
    return celsius  
  
# Example usage:  
print(fahrenheit_to_celsius(100))
```

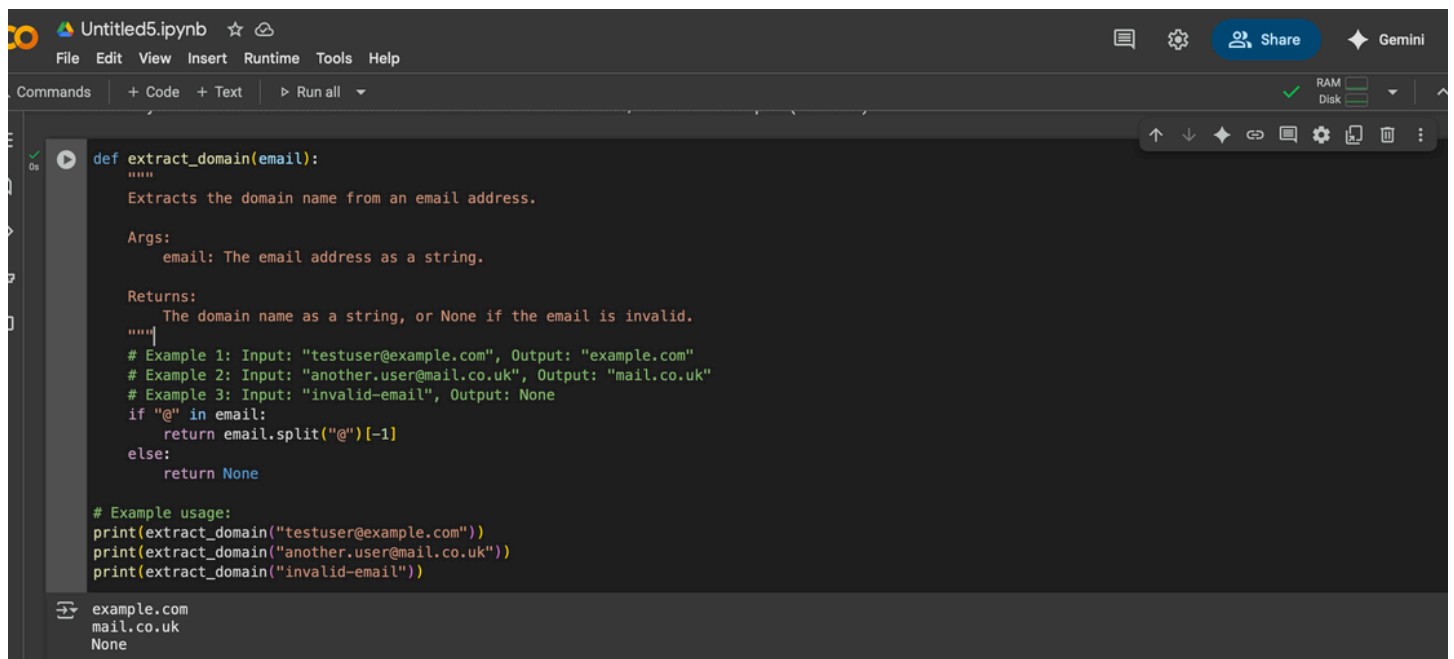
37.77777777777778

TASK 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).



Untitled5.ipynb ☆ ☁

File Edit View Insert Runtime Tools Help

Commands + Code + Text ▶ Run all

RAM Disk

```
def extract_domain(email):  
    """  
    Extracts the domain name from an email address.  
  
    Args:  
        email: The email address as a string.  
  
    Returns:  
        The domain name as a string, or None if the email is invalid.  
    """  
    # Example 1: Input: "testuser@example.com", Output: "example.com"  
    # Example 2: Input: "another.user@mail.co.uk", Output: "mail.co.uk"  
    # Example 3: Input: "invalid-email", Output: None  
    if "@" in email:  
        return email.split("@")[-1]  
    else:  
        return None  
  
# Example usage:  
print(extract_domain("testuser@example.com"))  
print(extract_domain("another.user@mail.co.uk"))  
print(extract_domain("invalid-email"))
```

example.com
mail.co.uk
None

TASK 4:-

Output comparison + student explanation on how examples helped the model.

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.

```
def find_maximum(num1, num2, num3):
    """
    Determines the maximum of three numbers without using the built-in max() function.

    Args:
        num1: The first number.
        num2: The second number.
        num3: The third number.

    Returns:
        The maximum of the three numbers.
    """
    if num1 >= num2 and num1 >= num3:
        return num1
    elif num2 >= num1 and num2 >= num3:
        return num2
    else:
        return num3

# Sample inputs and outputs
print(f"The maximum of 5, 10, 3 is: {find_maximum(5, 10, 3)}")
print(f"The maximum of -1, -5, 0 is: {find_maximum(-1, -5, 0)}")
print(f"The maximum of 7, 7, 7 is: {find_maximum(7, 7, 7)}")
```

Output:

```
The maximum of 5, 10, 3 is: 10
The maximum of -1, -5, 0 is: 0
The maximum of 7, 7, 7 is: 7
```

TASK 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.

```
def find_maximum(num1, num2, num3):
    """
    Determines the maximum of three numbers without using the built-in max() function.

    Args:
        num1: The first number.
        num2: The second number.
        num3: The third number.

    Returns:
        The maximum of the three numbers.
    """
    if num1 >= num2 and num1 >= num3:
        return num1
    elif num2 >= num1 and num2 >= num3:
        return num2
    else:
        return num3

# Sample inputs
print(f"The maximum of 5, 10, 3 is: {find_maximum(5, 10, 3)}") # Expected: 10
print(f"The maximum of -1, -5, 0 is: {find_maximum(-1, -5, 0)}") # Expected: 0
print(f"The maximum of 7, 7, 7 is: {find_maximum(7, 7, 7)}") # Expected: 7
```

Output:

```
The maximum of 5, 10, 3 is: 10
The maximum of -1, -5, 0 is: 0
The maximum of 7, 7, 7 is: 7
```