School of Computer Science and Artificial Intelligence

Lab Assignment #3.2

Program : B. Tech (CSE)

Specialization :AIML

Course Title :AI Assisted Coding

Course Code :23CS002PC304

Semester : VI

Academic Session : 2025-2026

Name of Student :P.Sushanthi

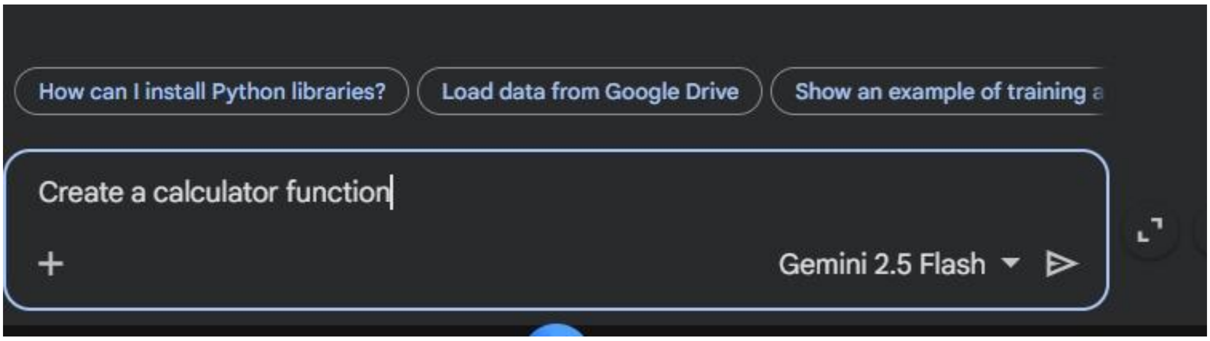
Enrollment No. : 2303A52102

Batch No. : 33

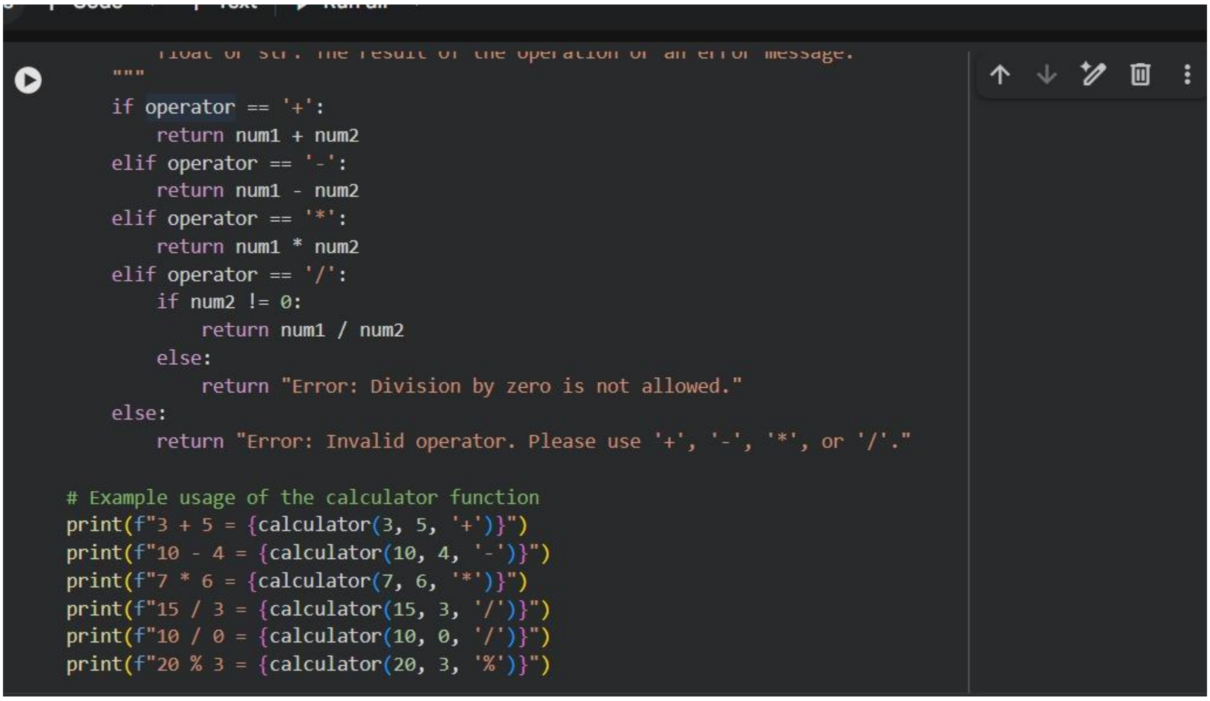
Date :13/01/26

Task 1 – Progressive Prompting (Calculator)

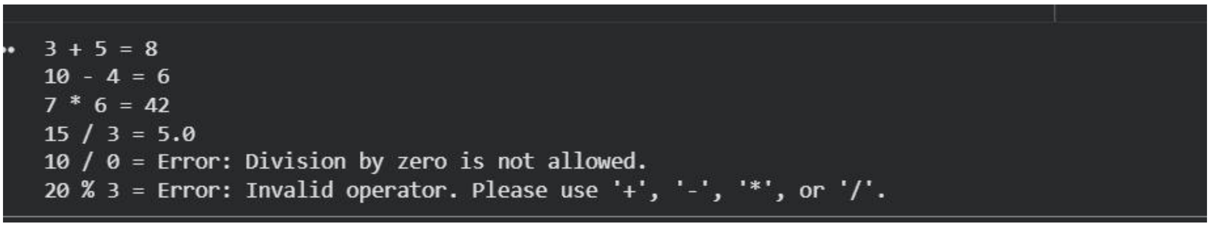
Prompt1:


How can I install Python libraries?
Load data from Google Drive
Show an example of training a
Gemini 2.5 Flash

Create a calculator function

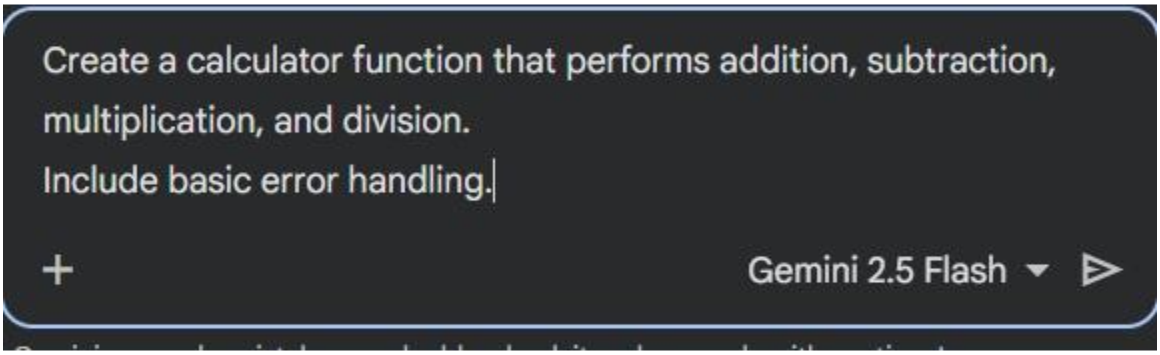


```
        float or str: The result of the operation or an error message.
    """
    if operator == '+':
        return num1 + num2
    elif operator == '-':
        return num1 - num2
    elif operator == '*':
        return num1 * num2
    elif operator == '/':
        if num2 != 0:
            return num1 / num2
        else:
            return "Error: Division by zero is not allowed."
    else:
        return "Error: Invalid operator. Please use '+', '-', '*', or '/'."

# Example usage of the calculator function
print(f"3 + 5 = {calculator(3, 5, '+')}")
print(f"10 - 4 = {calculator(10, 4, '-')}")
print(f"7 * 6 = {calculator(7, 6, '*')}")
print(f"15 / 3 = {calculator(15, 3, '/')}")
print(f"10 / 0 = {calculator(10, 0, '/')}")
print(f"20 % 3 = {calculator(20, 3, '%')}")
```

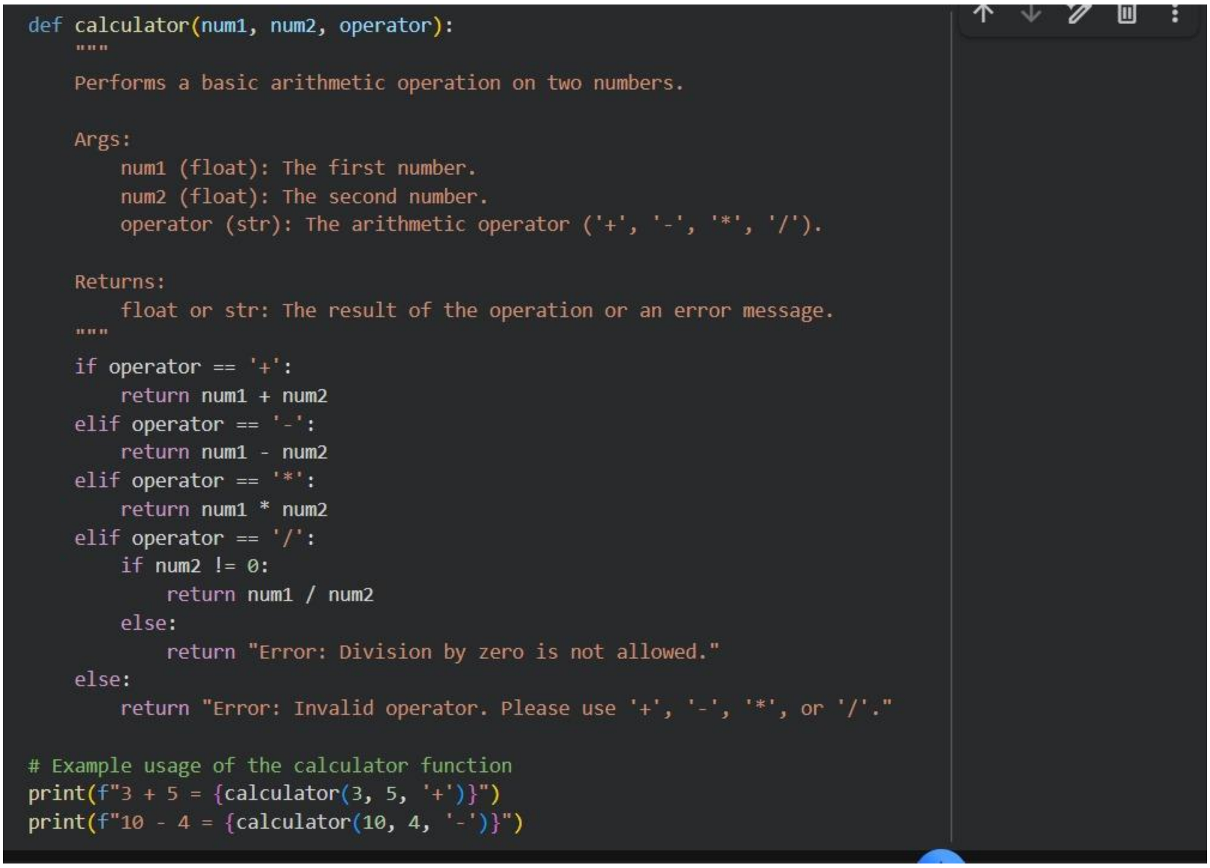


```
3 + 5 = 8
10 - 4 = 6
7 * 6 = 42
15 / 3 = 5.0
10 / 0 = Error: Division by zero is not allowed.
20 % 3 = Error: Invalid operator. Please use '+', '-', '*', or '/'.
```

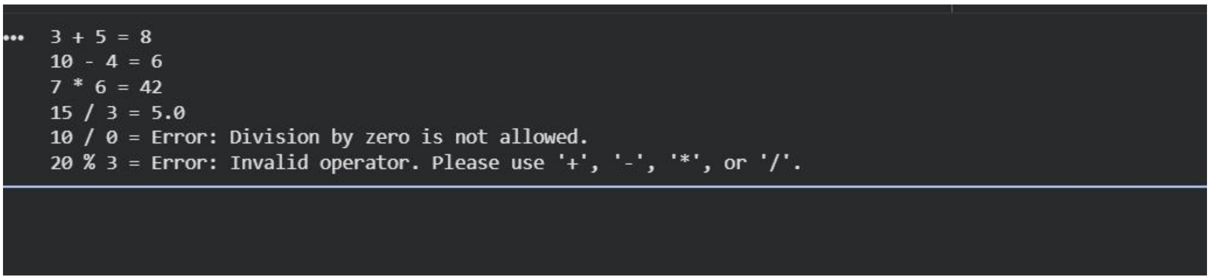
Prompt2:



Create a calculator function that performs addition, subtraction, multiplication, and division.
Include basic error handling.

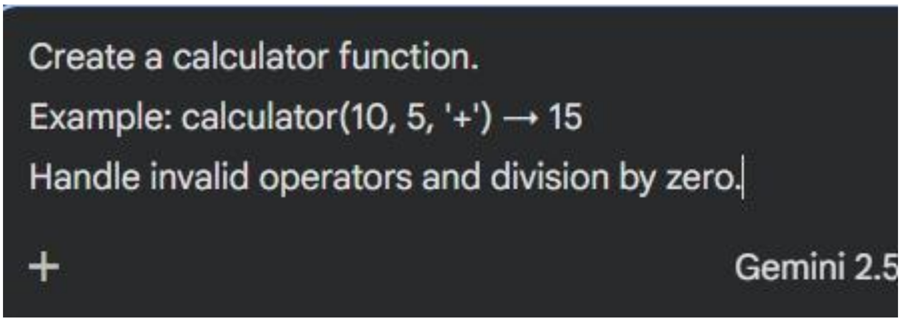


```python
def calculator(num1, num2, operator):
    """
    Performs a basic arithmetic operation on two numbers.

    Args:
        num1 (float): The first number.
        num2 (float): The second number.
        operator (str): The arithmetic operator ('+', '-', '*', '/').

    Returns:
        float or str: The result of the operation or an error message.
    """
    if operator == '+':
        return num1 + num2
    elif operator == '-':
        return num1 - num2
    elif operator == '*':
        return num1 * num2
    elif operator == '/':
        if num2 != 0:
            return num1 / num2
        else:
            return "Error: Division by zero is not allowed."
    else:
        return "Error: Invalid operator. Please use '+', '-', '*', or '/'."

# Example usage of the calculator function
print(f"3 + 5 = {calculator(3, 5, '+')}")
print(f"10 - 4 = {calculator(10, 4, '-')}")
```



```
3 + 5 = 8
10 - 4 = 6
7 * 6 = 42
15 / 3 = 5.0
10 / 0 = Error: Division by zero is not allowed.
20 % 3 = Error: Invalid operator. Please use '+', '-', '*', or '/'.
```

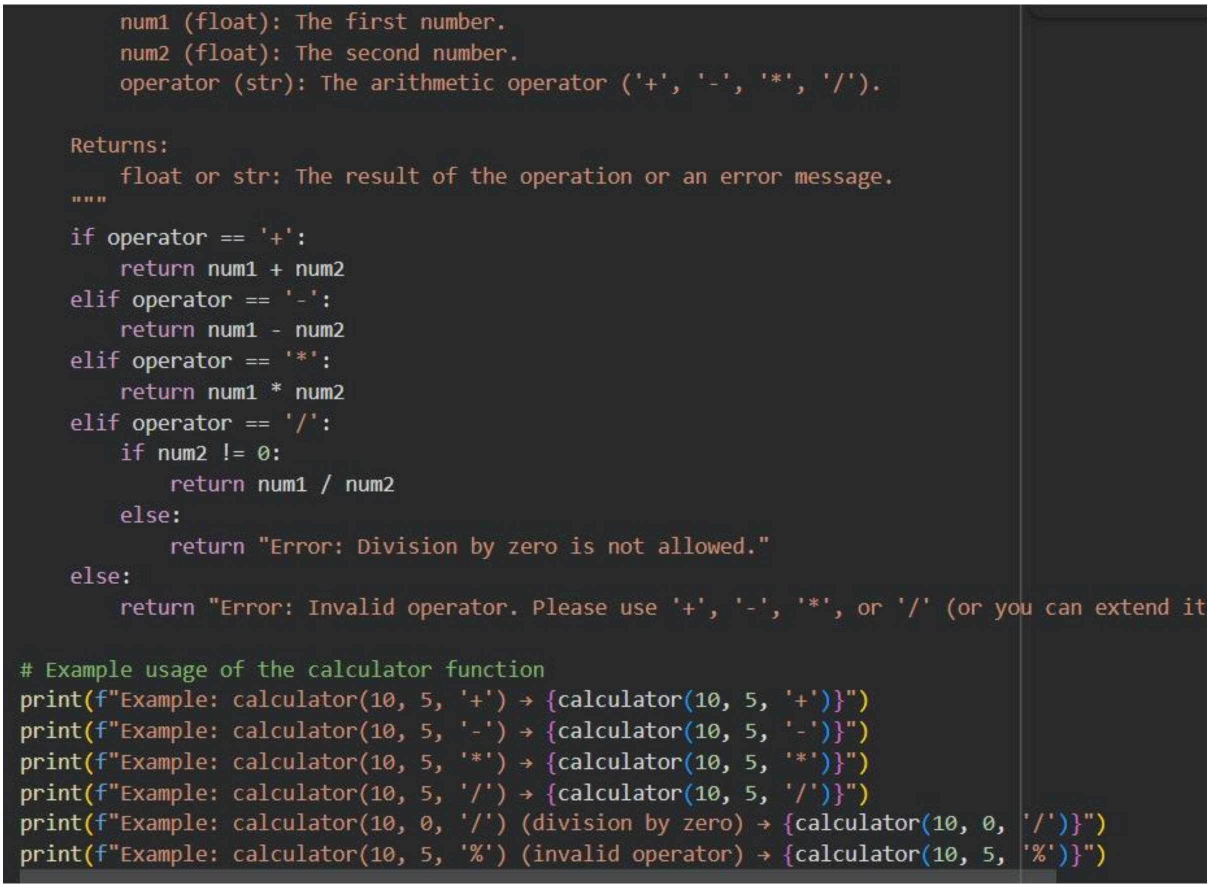
Prompt 3:



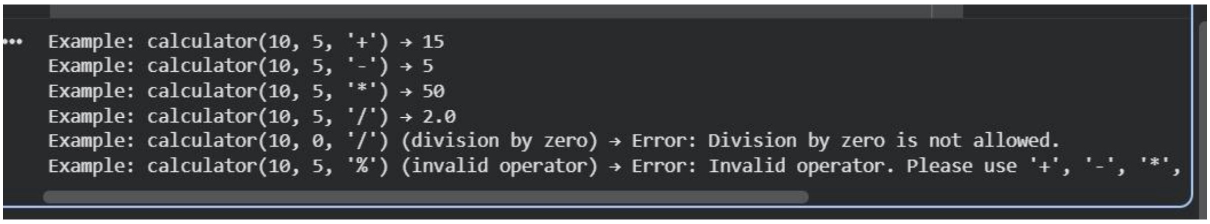
Create a calculator function.

Example: calculator(10, 5, '+') → 15

Handle invalid operators and division by zero.



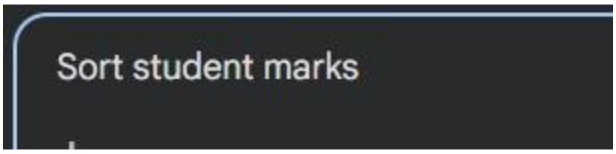
```
        num1 (float): The first number.
        num2 (float): The second number.
        operator (str): The arithmetic operator ('+', '-', '*', '/').

    Returns:
        float or str: The result of the operation or an error message.
    """
    if operator == '+':
        return num1 + num2
    elif operator == '-':
        return num1 - num2
    elif operator == '*':
        return num1 * num2
    elif operator == '/':
        if num2 != 0:
            return num1 / num2
        else:
            return "Error: Division by zero is not allowed."
    else:
        return "Error: Invalid operator. Please use '+', '-', '*', or '/' (or you can extend it

# Example usage of the calculator function
print(f"Example: calculator(10, 5, '+') → {calculator(10, 5, '+')}")
print(f"Example: calculator(10, 5, '-') → {calculator(10, 5, '-')}")
print(f"Example: calculator(10, 5, '*') → {calculator(10, 5, '*')}")
print(f"Example: calculator(10, 5, '/') → {calculator(10, 5, '/')}")
print(f"Example: calculator(10, 0, '/') (division by zero) → {calculator(10, 0, '/')}")
print(f"Example: calculator(10, 5, '%') (invalid operator) → {calculator(10, 5, '%')}")
```



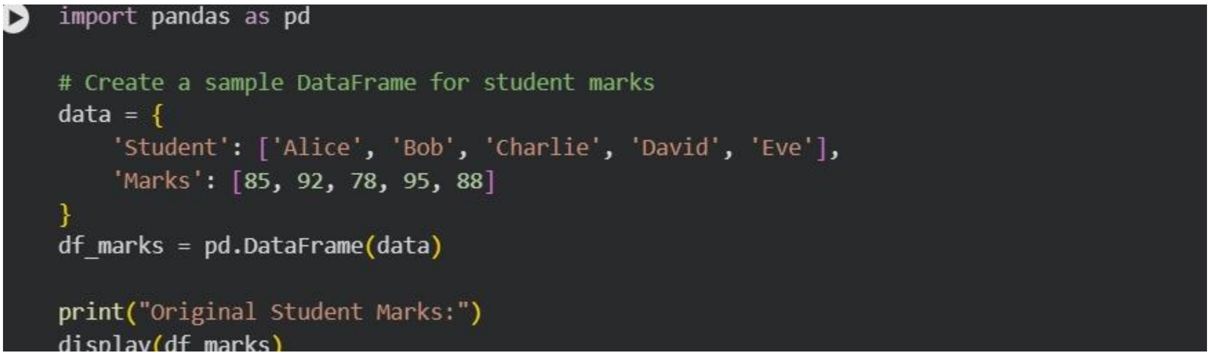
```
Example: calculator(10, 5, '+') → 15
Example: calculator(10, 5, '-') → 5
Example: calculator(10, 5, '*') → 50
Example: calculator(10, 5, '/') → 2.0
Example: calculator(10, 0, '/') (division by zero) → Error: Division by zero is not allowed.
Example: calculator(10, 5, '%') (invalid operator) → Error: Invalid operator. Please use '+', '-', '*',
```

Task 2 – Sorting Student Marks

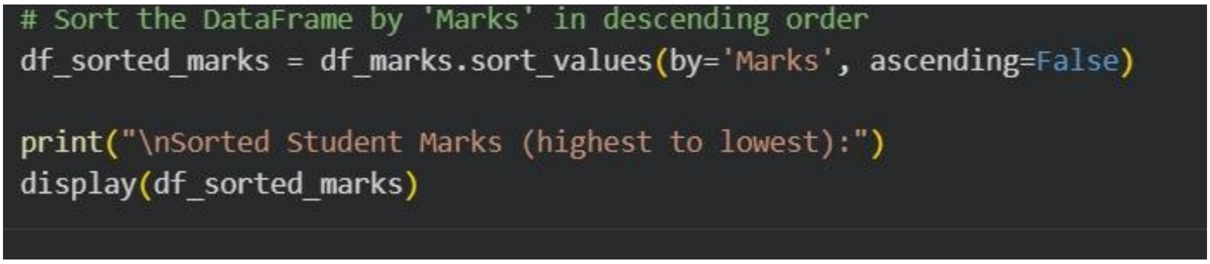
Initial Prompt:



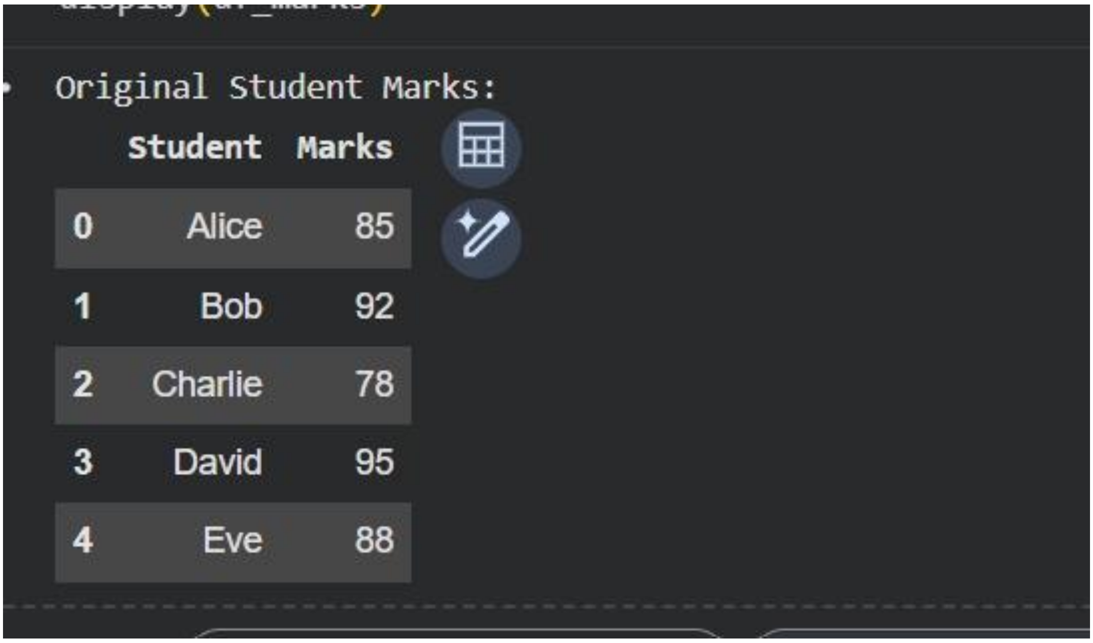
Sort student marks



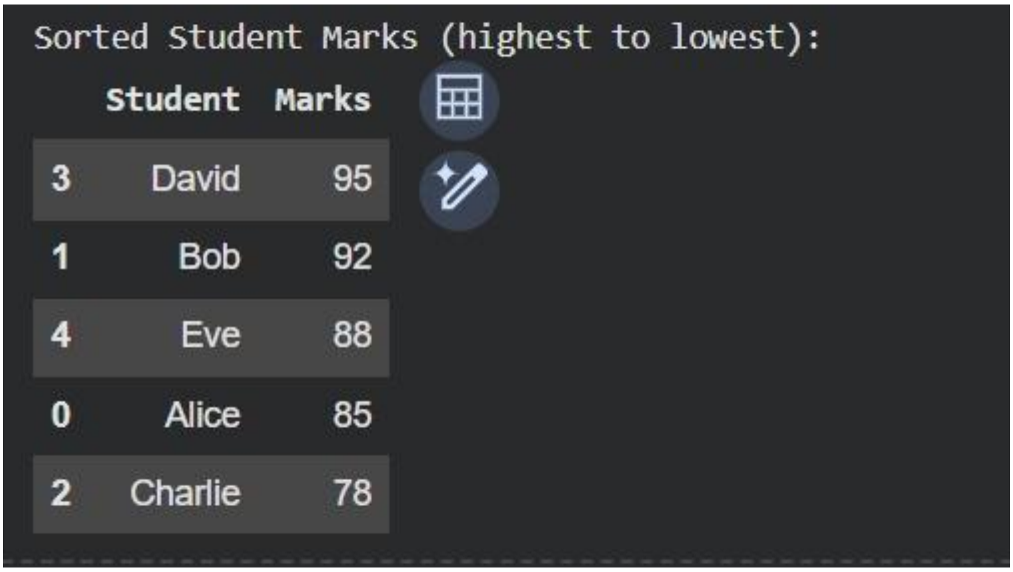
```python
import pandas as pd

# Create a sample DataFrame for student marks
data = {
    'Student': ['Alice', 'Bob', 'Charlie', 'David', 'Eve'],
    'Marks': [85, 92, 78, 95, 88]
}
df_marks = pd.DataFrame(data)

print("Original Student Marks:")
display(df_marks)
```



```python
# Sort the DataFrame by 'Marks' in descending order
df_sorted_marks = df_marks.sort_values(by='Marks', ascending=False)

print("\nSorted Student Marks (highest to lowest):")
display(df_sorted_marks)
```

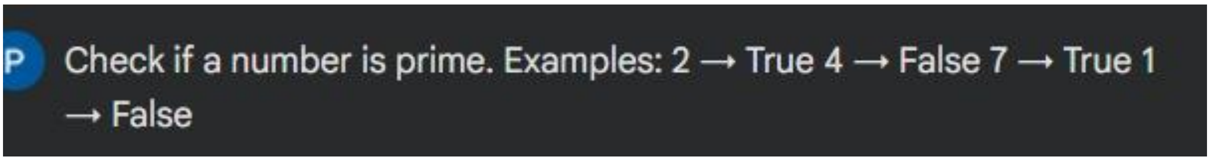


```
Original Student Marks:
   Student  Marks
0    Alice     85
1      Bob     92
2  Charlie     78
3    David     95
4      Eve     88
```

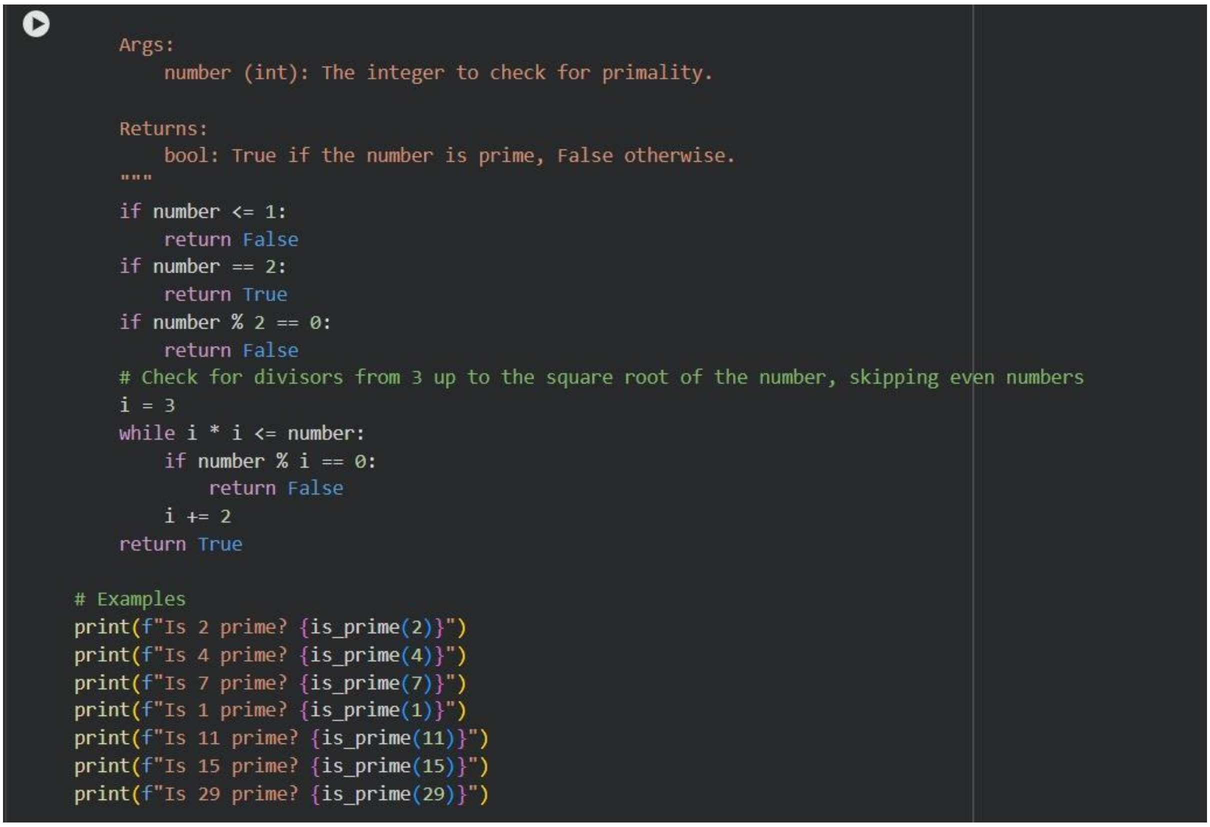


```
Sorted Student Marks (highest to lowest):
   Student  Marks
3    David     95
1      Bob     92
4      Eve     88
0    Alice     85
2  Charlie     78
```

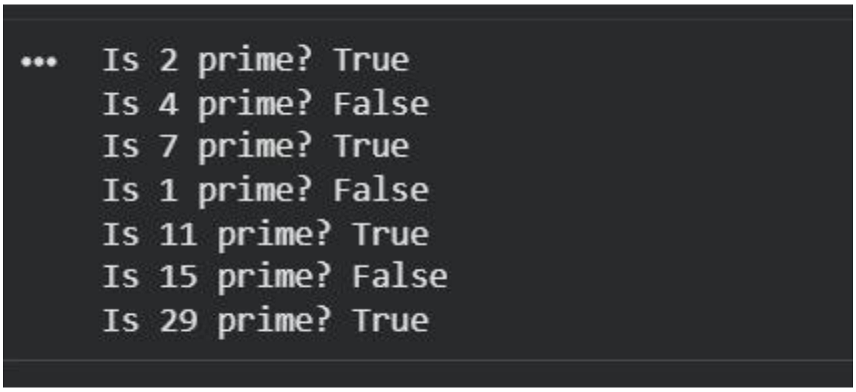
Task 3 – Few-Shot Prompting (Prime Number)



Check if a number is prime. Examples: 2 → True 4 → False 7 → True 1 → False

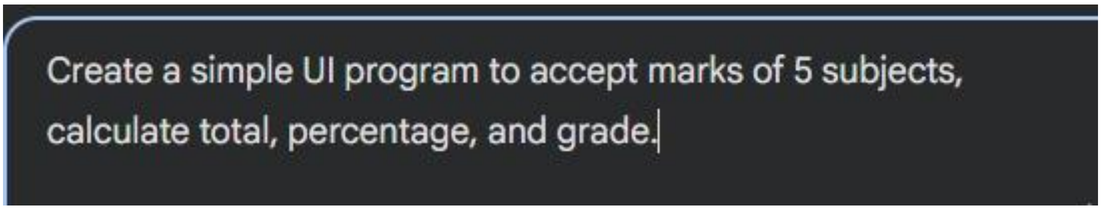


```
    Args:
        number (int): The integer to check for primality.

    Returns:
        bool: True if the number is prime, False otherwise.
    """
    if number <= 1:
        return False
    if number == 2:
        return True
    if number % 2 == 0:
        return False
    # Check for divisors from 3 up to the square root of the number, skipping even numbers
    i = 3
    while i * i <= number:
        if number % i == 0:
            return False
        i += 2
    return True

# Examples
print(f"Is 2 prime? {is_prime(2)}")
print(f"Is 4 prime? {is_prime(4)}")
print(f"Is 7 prime? {is_prime(7)}")
print(f"Is 1 prime? {is_prime(1)}")
print(f"Is 11 prime? {is_prime(11)}")
print(f"Is 15 prime? {is_prime(15)}")
print(f"Is 29 prime? {is_prime(29)}")
```

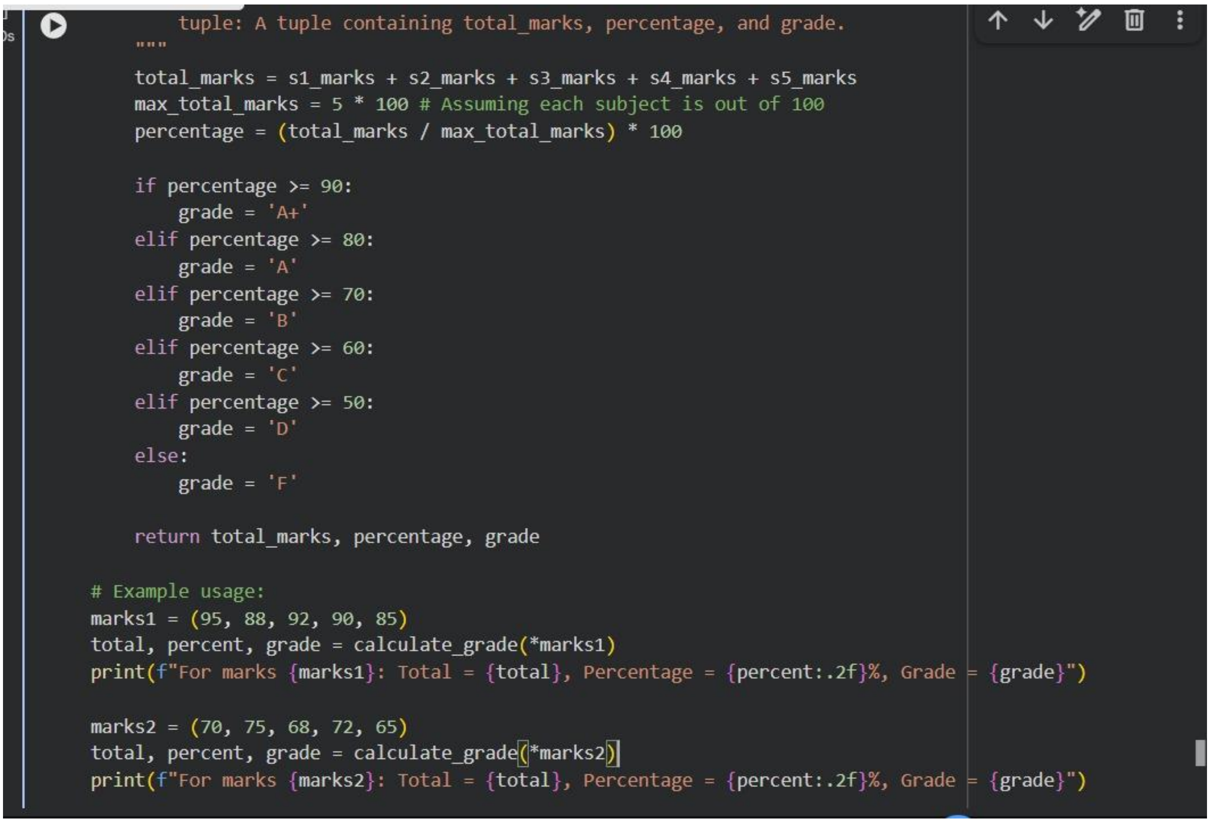


```
Is 2 prime? True
Is 4 prime? False
Is 7 prime? True
Is 1 prime? False
Is 11 prime? True
Is 15 prime? False
Is 29 prime? True
```

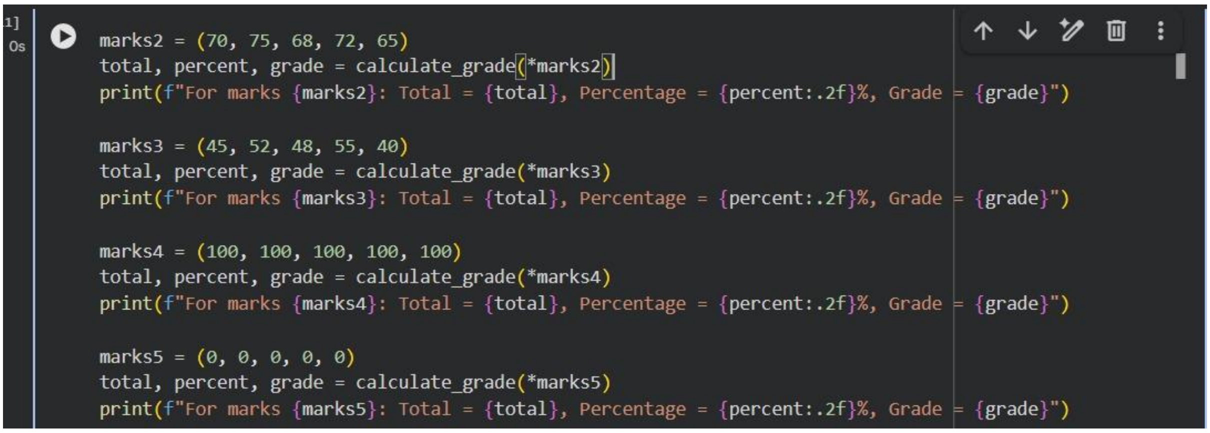
Task 4 – Prompt-Guided UI Design



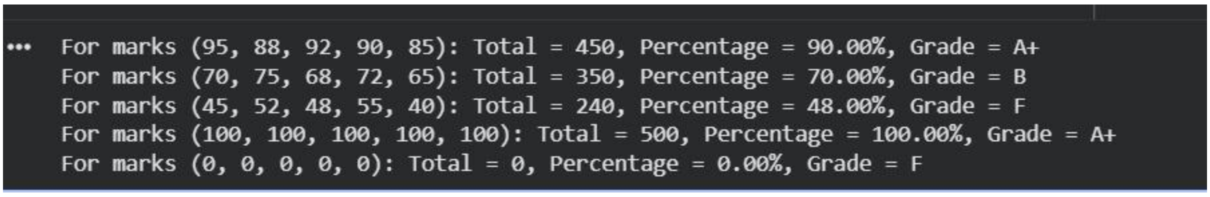
Create a simple UI program to accept marks of 5 subjects, calculate total, percentage, and grade.



```
        tuple: A tuple containing total_marks, percentage, and grade.
    """
    total_marks = s1_marks + s2_marks + s3_marks + s4_marks + s5_marks
    max_total_marks = 5 * 100 # Assuming each subject is out of 100
    percentage = (total_marks / max_total_marks) * 100

    if percentage >= 90:
        grade = 'A+'
    elif percentage >= 80:
        grade = 'A'
    elif percentage >= 70:
        grade = 'B'
    elif percentage >= 60:
        grade = 'C'
    elif percentage >= 50:
        grade = 'D'
    else:
        grade = 'F'

    return total_marks, percentage, grade

# Example usage:
marks1 = (95, 88, 92, 90, 85)
total, percent, grade = calculate_grade(*marks1)
print(f"For marks {marks1}: Total = {total}, Percentage = {percent:.2f}%, Grade = {grade}")

marks2 = (70, 75, 68, 72, 65)
total, percent, grade = calculate_grade(*marks2)
print(f"For marks {marks2}: Total = {total}, Percentage = {percent:.2f}%, Grade = {grade}")
```



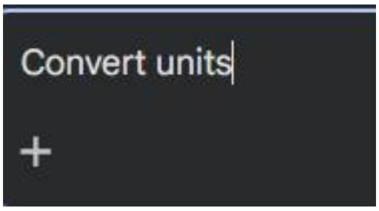
```
marks2 = (70, 75, 68, 72, 65)
total, percent, grade = calculate_grade(*marks2)
print(f"For marks {marks2}: Total = {total}, Percentage = {percent:.2f}%, Grade = {grade}")

marks3 = (45, 52, 48, 55, 40)
total, percent, grade = calculate_grade(*marks3)
print(f"For marks {marks3}: Total = {total}, Percentage = {percent:.2f}%, Grade = {grade}")

marks4 = (100, 100, 100, 100, 100)
total, percent, grade = calculate_grade(*marks4)
print(f"For marks {marks4}: Total = {total}, Percentage = {percent:.2f}%, Grade = {grade}")

marks5 = (0, 0, 0, 0, 0)
total, percent, grade = calculate_grade(*marks5)
print(f"For marks {marks5}: Total = {total}, Percentage = {percent:.2f}%, Grade = {grade}")
```



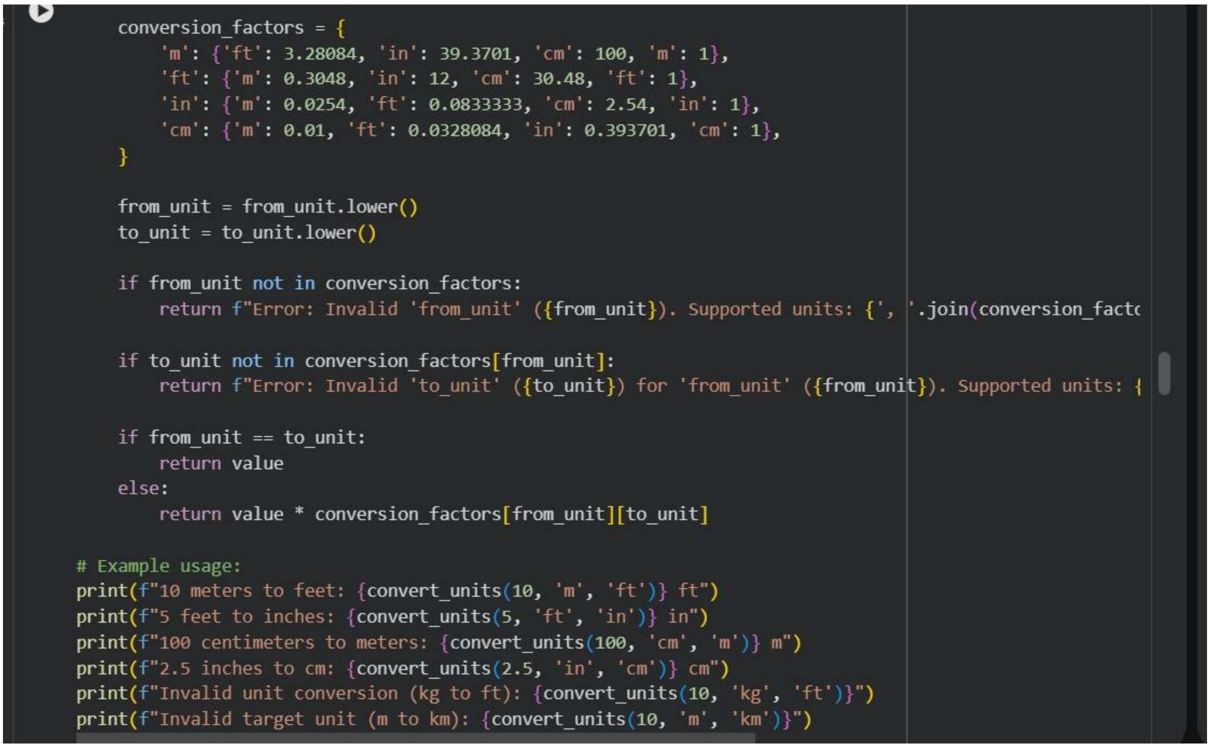
```
For marks (95, 88, 92, 90, 85): Total = 450, Percentage = 90.00%, Grade = A+
For marks (70, 75, 68, 72, 65): Total = 350, Percentage = 70.00%, Grade = B
For marks (45, 52, 48, 55, 40): Total = 240, Percentage = 48.00%, Grade = F
For marks (100, 100, 100, 100, 100): Total = 500, Percentage = 100.00%, Grade = A+
For marks (0, 0, 0, 0, 0): Total = 0, Percentage = 0.00%, Grade = F
```

Task 5 – Unit Conversion Analysis

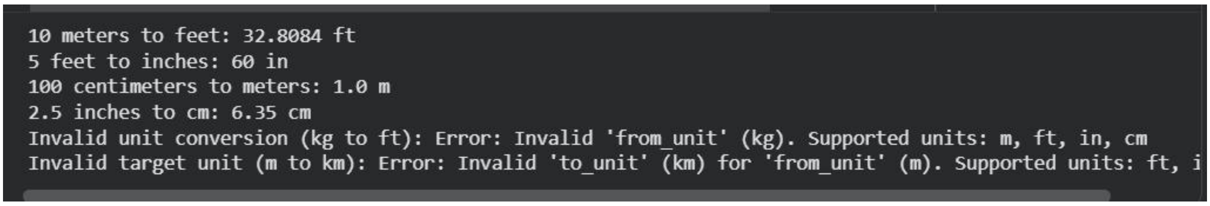
Basic Prompt:



Convert units

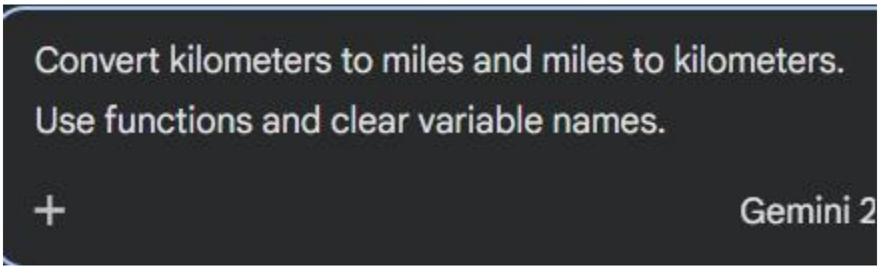


```
    conversion_factors = {
        'm': {'ft': 3.28084, 'in': 39.3701, 'cm': 100, 'm': 1},
        'ft': {'m': 0.3048, 'in': 12, 'cm': 30.48, 'ft': 1},
        'in': {'m': 0.0254, 'ft': 0.0833333, 'cm': 2.54, 'in': 1},
        'cm': {'m': 0.01, 'ft': 0.0328084, 'in': 0.393701, 'cm': 1},
    }

    from_unit = from_unit.lower()
    to_unit = to_unit.lower()

    if from_unit not in conversion_factors:
        return f"Error: Invalid 'from_unit' ({from_unit}). Supported units: {', '.join(conversion_fact

    if to_unit not in conversion_factors[from_unit]:
        return f"Error: Invalid 'to_unit' ({to_unit}) for 'from_unit' ({from_unit}). Supported units: {

    if from_unit == to_unit:
        return value
    else:
        return value * conversion_factors[from_unit][to_unit]

# Example usage:
print(f"10 meters to feet: {convert_units(10, 'm', 'ft')} ft")
print(f"5 feet to inches: {convert_units(5, 'ft', 'in')} in")
print(f"100 centimeters to meters: {convert_units(100, 'cm', 'm')} m")
print(f"2.5 inches to cm: {convert_units(2.5, 'in', 'cm')} cm")
print(f"Invalid unit conversion (kg to ft): {convert_units(10, 'kg', 'ft')}")
print(f"Invalid target unit (m to km): {convert_units(10, 'm', 'km')}")
```

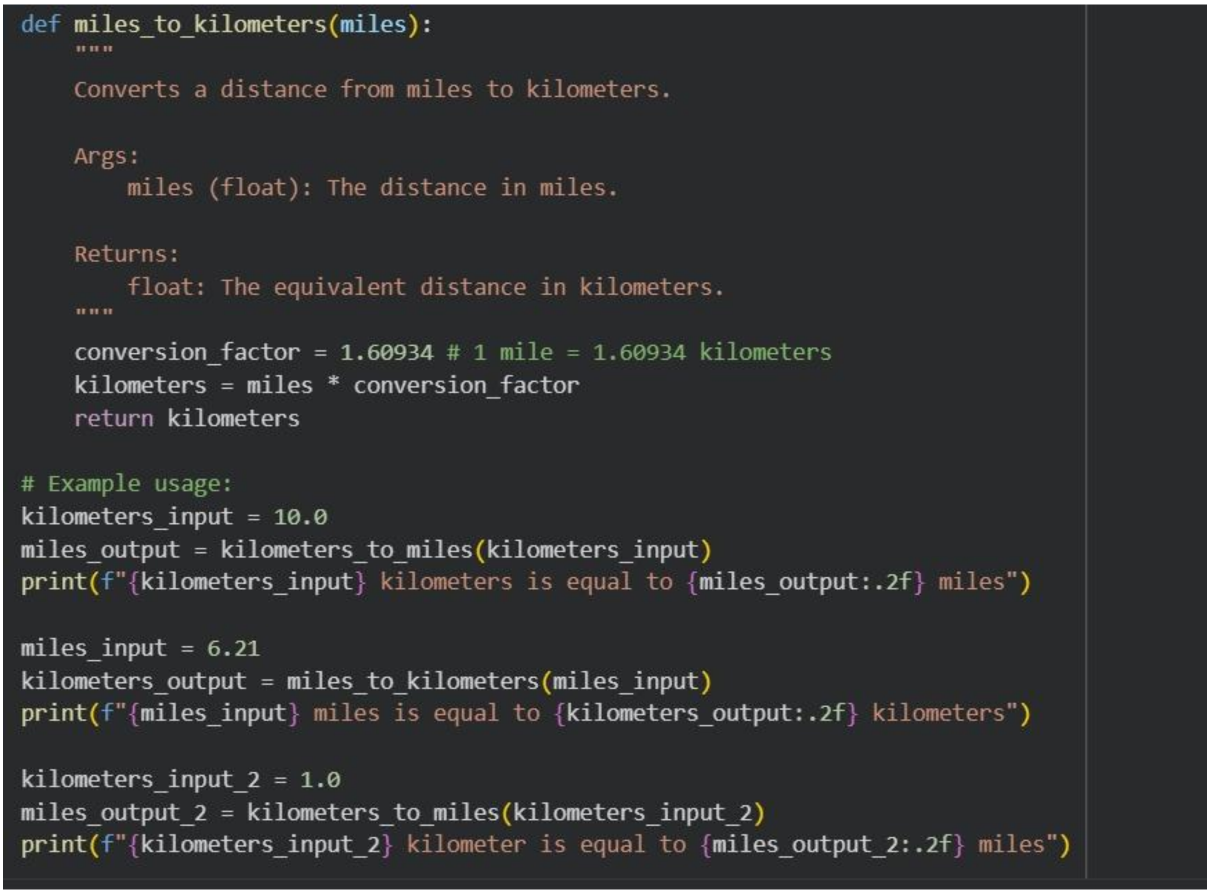


```
10 meters to feet: 32.8084 ft
5 feet to inches: 60 in
100 centimeters to meters: 1.0 m
2.5 inches to cm: 6.35 cm
Invalid unit conversion (kg to ft): Error: Invalid 'from_unit' (kg). Supported units: m, ft, in, cm
Invalid target unit (m to km): Error: Invalid 'to_unit' (km) for 'from_unit' (m). Supported units: ft, i
```

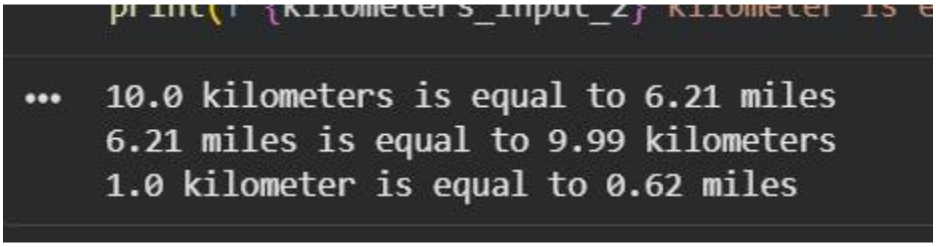
Improved Prompt:



Convert kilometers to miles and miles to kilometers.
Use functions and clear variable names.



```python
def miles_to_kilometers(miles):
    """
    Converts a distance from miles to kilometers.

    Args:
        miles (float): The distance in miles.

    Returns:
        float: The equivalent distance in kilometers.
    """
    conversion_factor = 1.60934 # 1 mile = 1.60934 kilometers
    kilometers = miles * conversion_factor
    return kilometers

# Example usage:
kilometers_input = 10.0
miles_output = kilometers_to_miles(kilometers_input)
print(f"{kilometers_input} kilometers is equal to {miles_output:.2f} miles")

miles_input = 6.21
kilometers_output = miles_to_kilometers(miles_input)
print(f"{miles_input} miles is equal to {kilometers_output:.2f} kilometers")

kilometers_input_2 = 1.0
miles_output_2 = kilometers_to_miles(kilometers_input_2)
print(f"{kilometers_input_2} kilometer is equal to {miles_output_2:.2f} miles")
```



```
    print(f"{kilometers_input_2} kilometer is e
```

```
•••  10.0 kilometers is equal to 6.21 miles
     6.21 miles is equal to 9.99 kilometers
     1.0 kilometer is equal to 0.62 miles
```