SCHOOL OF COMPUTER SCIENCE AND ARTIFICIAL INTELLIGENCE		**DEPARTMENT OF COMPUTER SCIENCE ENGINEERING**	
Program Name: M. Tech/MCA	**Assignment Type: Lab**	**AcademicYear:**2025-2026	
Course Coordinator Name		Venkataramana Veeramsetty	
Course Code		**Course Title**	AI Assisted Problem Solving Using Python
Year/Sem	I/I	**Regulation**	R24
Date and Day of Assignment	Week1 - TUESDAY	**Time(s)**	
Duration	2 Hours	**Applicable to Batches**	M. Tech/MCA
AssignmentNumber:2.3(Present assignment number)/**24**(Total number of assignments)			

Q.No.	**Question**	***Expected Time to complete***
1	Lab 2: Exploring Additional AI Coding Tools – Gemini (Colab) and Cursor AI **Lab Objectives:** ● To explore and evaluate the functionality of Google Gemini for AI-assisted coding within Google Colab. ● To understand and use Cursor AI for code generation, explanation, and refactoring. ● To compare outputs and usability between Gemini, GitHub Copilot, and Cursor AI. ● To perform code optimization and documentation using AI tools. **Lab Outcomes (LOs):** After completing this lab, students will be able to: ● Generate Python code using Google Gemini in Google Colab. ● Analyze the effectiveness of code explanations and suggestions by Gemini. ● Set up and use Cursor AI for AI-powered coding assistance. ● Evaluate and refactor code using Cursor AI features. ● Compare AI tool behavior and code quality across different platforms. **Task Description#1** ● Use Google Gemini in Colab to write a function that reads a CSV file and calculates mean, min, max. **Expected Output#1** ● Functional code with output and screenshot **Task Description#2**	Week1 - TuesDay

- Compare Gemini and Copilot outputs for a palindrome check function.

Expected Output#2

- Side-by-side comparison and observations

Task Description#3

- Ask Gemini to explain a Python function (to calculate area of various shapes) line by line..

Expected Output#3

- Detailed explanation with code snippet

Task Description#4

- Install and configure Cursor AI. Use it to generate a Python function (e.g., sum of squares).

Expected Output#4

- Screenshots of working environments with few prompts to generate python code

Task Description#5

- Student need to write code to calculate sum of add number and even numbers in the list

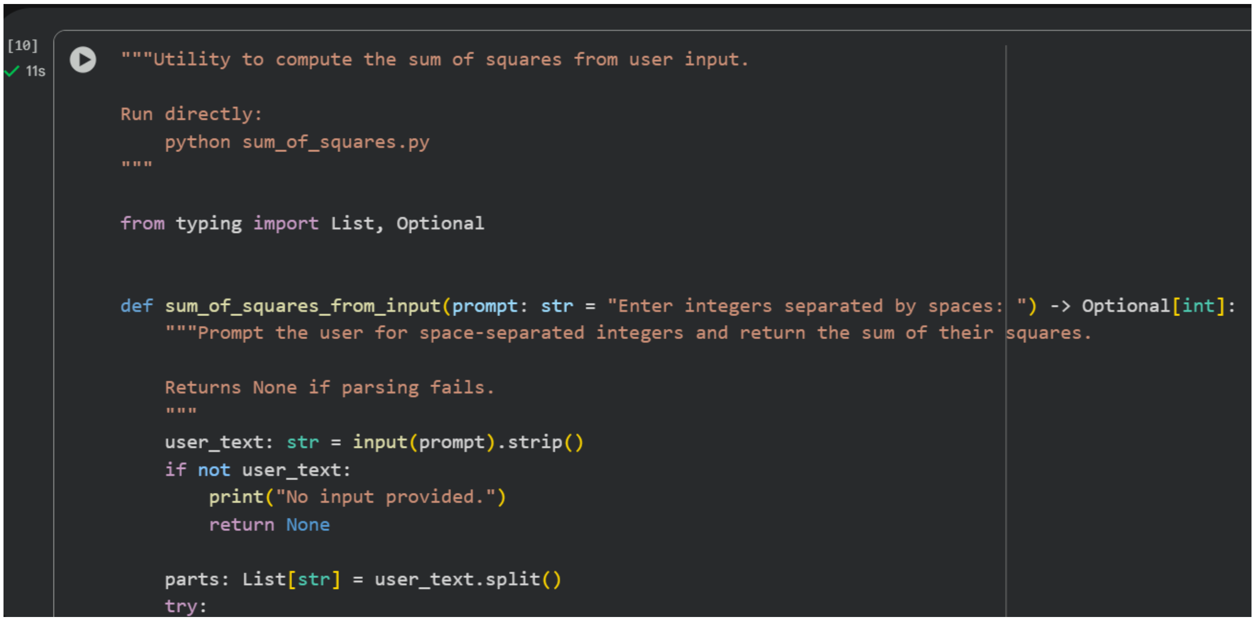
Expected Output#5

- Refactored code written by student with improved logic

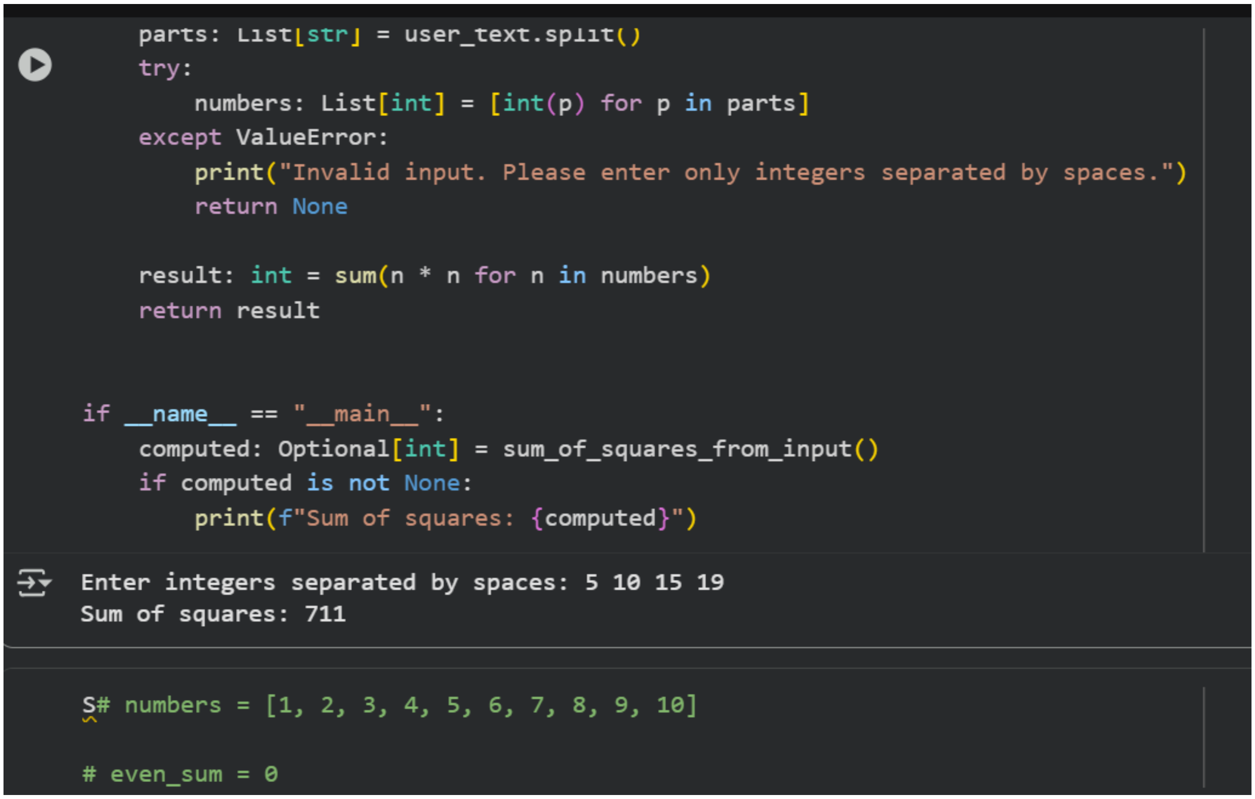
Note: Report should be submitted a word document for all tasks in a single document with prompts, comments & code explanation, and output and if required, screenshots

Evaluation Criteria:

Criteria	**Max Marks**
Successful Use of Gemini in Colab (Task#1 & #2)	2.5
Code Explanation Accuracy (Gemini) (Task#3)	2.5
Cursor AI Setup and Usage (Task#4)	2.5
Refactoring and Improvement Analysis (Task#5)	2.5
Total	**10 Marks**



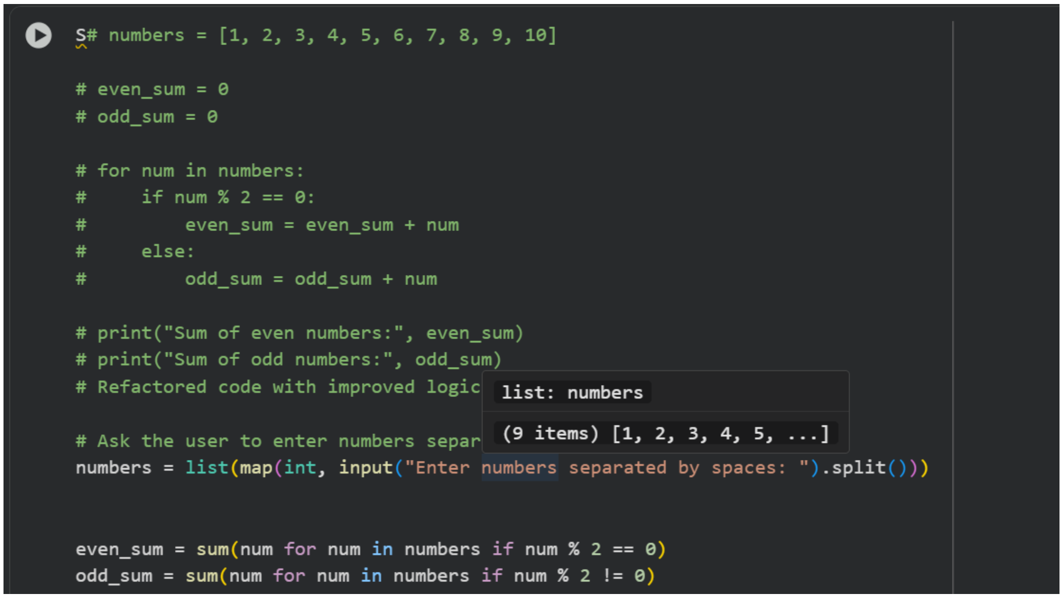
```
"""Utility to compute the sum of squares from user input.

Run directly:
    python sum_of_squares.py
"""

from typing import List, Optional


def sum_of_squares_from_input(prompt: str = "Enter integers separated by spaces: ") -> Optional[int]:
    """Prompt the user for space-separated integers and return the sum of their squares.

    Returns None if parsing fails.
    """
    user_text: str = input(prompt).strip()
    if not user_text:
        print("No input provided.")
        return None

    parts: List[str] = user_text.split()
    try:
```



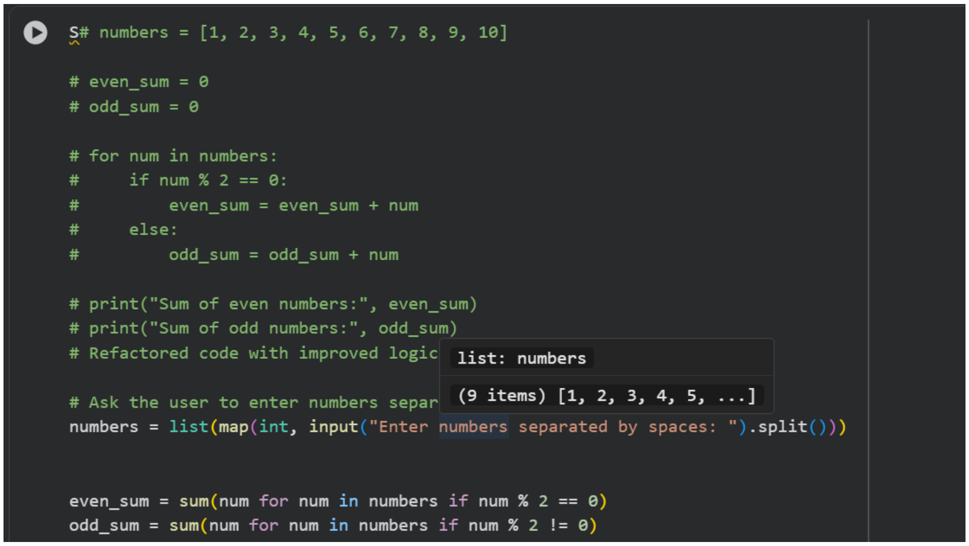
```
    parts: List[str] = user_text.split()
    try:
        numbers: List[int] = [int(p) for p in parts]
    except ValueError:
        print("Invalid input. Please enter only integers separated by spaces.")
        return None

    result: int = sum(n * n for n in numbers)
    return result


if __name__ == "__main__":
    computed: Optional[int] = sum_of_squares_from_input()
    if computed is not None:
        print(f"Sum of squares: {computed}")
```

```
Enter integers separated by spaces: 5 10 15 19
Sum of squares: 711
```

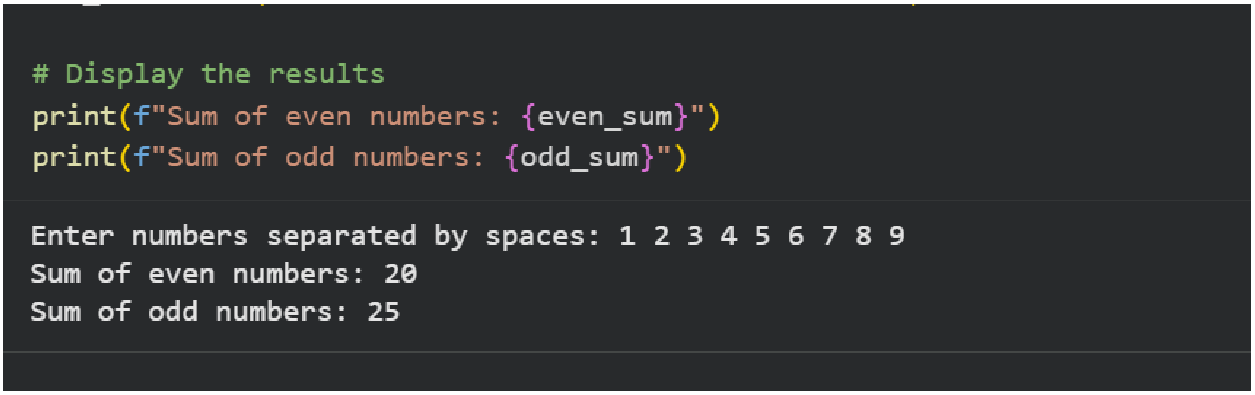
```
S# numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

# even_sum = 0
```



```
S# numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

# even_sum = 0
# odd_sum = 0

# for num in numbers:
#     if num % 2 == 0:
#         even_sum = even_sum + num
#     else:
#         odd_sum = odd_sum + num

# print("Sum of even numbers:", even_sum)
# print("Sum of odd numbers:", odd_sum)
# Refactored code with improved logic

# Ask the user to enter numbers separ
numbers = list(map(int, input("Enter numbers separated by spaces: ").split()))


even_sum = sum(num for num in numbers if num % 2 == 0)
odd_sum = sum(num for num in numbers if num % 2 != 0)
```



```
S# numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

# even_sum = 0
# odd_sum = 0

# for num in numbers:
#     if num % 2 == 0:
#         even_sum = even_sum + num
#     else:
#         odd_sum = odd_sum + num

# print("Sum of even numbers:", even_sum)
# print("Sum of odd numbers:", odd_sum)
# Refactored code with improved logic

# Ask the user to enter numbers separ
numbers = list(map(int, input("Enter numbers separated by spaces: ").split()))


even_sum = sum(num for num in numbers if num % 2 == 0)
odd_sum = sum(num for num in numbers if num % 2 != 0)
```



```
# Display the results
print(f"Sum of even numbers: {even_sum}")
print(f"Sum of odd numbers: {odd_sum}")
```

```
Enter numbers separated by spaces: 1 2 3 4 5 6 7 8 9
Sum of even numbers: 20
Sum of odd numbers: 25
```