AI Assisted Coding (III Year) Assignment

NAME : B. Rithwik

HT NO :2303A52330

BATCH :35

Lab 1: Exploring Additional AI Coding Tools beyond Copilot – Gemini (Colab)

and Cursor AI

Task1 -

Prompt: Write a Python function that takes a list of numbers and returns the mean, minimum, and maximum values. Also display the output.

Code :

```
def statistical_summary(data):
    if not data:
        return None

    mean_value = sum(data) / len(data)
    min_value = min(data)
    max_value = max(data)

    return mean_value, min_value, max_value


# Sample input
numbers = [10, 20, 30, 40, 50]

# Function call
mean_val, min_val, max_val = statistical_summary(numbers)

print("Mean:", mean_val)
print("Minimum:", min_val)
print("Maximum:", max_val)
```

```
Mean: 30.0
Minimum: 10
Maximum: 50
```

Explanation:

□ Gemini generated correct and readable code

□ Used built-in Python functions efficiently

□ Suitable for beginners with minimal modification

Task 2: Armstrong Number – AI Comparison

Prompt: write a python program to reverse a string without using functions and optimize the code

Code 1:

```
[8]
✓ 0s
def is_armstrong(num):
    digits = str(num)
    power = len(digits)
    total = sum(int(d)**power for d in digits)
    return total == num
```

Code 2:

```
def is_armstrong(number):
    temp = number
    n = len(str(number))
    result = 0

    while temp > 0:
        digit = temp % 10
        result += digit ** n
        temp //= 10

    return result == number
```

Explanation:

Gemini favors concise, modern Python. Copilot mirrors traditional logic taught in classrooms.

TASK-3

Prompt: Write a Python program to check if a year is a leap year

Code:

```
def is_leap_year(year):
    if year % 4 == 0 and year % 100 != 0:
        return True
    if year % 400 == 0:
        return True
    return False

year = int(input("Enter year: "))

if (year % 400 == 0) or (year % 4 == 0 and year % 100 != 0):
    print("Leap Year")
else:
    print("Not a Leap Year")


Enter year: 2020
Leap Year
```

Explanation:

□ Prompt 2 produced cleaner logic and user interaction

□ Cursor adapts structure significantly based on prompt wording

□ Demonstrates strong refactoring and formatting ability

TASK -4

Student Logic + AI Refactoring (Odd/Even Sum)

Code:

```
numbers = (1, 2, 3, 4, 5, 6)
even_sum = 0
odd_sum = 0

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

print("Even sum:", even_sum)
print("Odd sum:", odd_sum)
```

```
Even sum: 12
Odd sum: 9
```

Explanation:

□ Reduced code length

□ Eliminated manual counters

□ Used generator expressions

□ Improved readability and maintainability