Assignment – 9.2
AIAC

Name:Mohammed Rahil Raza

Ht.no:2303A51433

bt.no:21

Task Description -1 (Documentation – Function Summary Generation)

Task:
Use AI to generate concise functional summaries for each Python function in a given script.

Instructions:

- Provide a Python script to the AI.
- Ask the AI to write a short summary describing the purpose of each function.
- Ensure summaries are brief and technically accurate.
- Do not include code implementation details.

```python
def add_numbers(a, b):
    return a + b


def check_even(number):
    return number % 2 == 0


def find_maximum(values):
    max_value = values[0]
    for val in values:
        if val > max_value:
            max_value = val
    return max_value


a = int(input("Enter first number: "))
b = int(input("Enter second number: "))
print("Sum:", add_numbers(a, b))

num = int(input("Enter number to check even: "))
print("Is Even:", check_even(num))

numbers = list(map(int, input("Enter numbers separated by space: ").split()))
print("Maximum:", find_maximum(numbers))
```

Task Description -2 (Documentation – Logical Explanation for Conditions and Loops)

Task:
Use AI to document the logic behind conditional statements and loops in a Python program.

Instructions:

- Provide a Python program without comments.
- Instruct AI to explain only decision-making logic and loop behavior.
- Skip basic syntax explanations.

```
def calculate_average(numbers):
    total = 0
    count = 0

    for num in numbers:
        if num >= 0:
            total += num
            count += 1

    if count == 0:
        return 0

    return total / count


nums = list(map(int, input("Enter numbers separated by space: ").split()))
print("Average:", calculate_average(nums))
```

Task Description -3 (Documentation – File-Level Overview)

Task:
Use AI to generate a high-level overview describing the functionality of an entire Python file.

Instructions:

- Provide the complete Python file to AI.
- Ask AI to write a brief overview summarizing the file's purpose and functionality.
- Place the overview at the top of the file.

Expected Output -3:
A Python file with a clear and concise file-level overview at the beginning.

```python
"""
This file contains basic utility functions for numerical operations,
including subtraction, even number checking, and maximum value detection.
"""

def subtract(a, b):
    return a - b


def check_even(n):
    return n % 2 == 0


def find_max(values):
    maximum = values[0]
    for val in values:
        if val > maximum:
            maximum = val
    return maximum


a = int(input("Enter first number: "))
b = int(input("Enter second number: "))
print("Subtraction:", subtract(a, b))
```

Task Description -4 (Documentation – Refine Existing Documentation)

Task:
Use AI to improve clarity and consistency of existing documentation in Python code.

Instructions:

- Provide Python code containing basic or unclear comments.
- Ask AI to rewrite the documentation to improve clarity and consistency.
- Ensure technical meaning remains unchanged.

Expected Output -4:
Python code with refined and improved documentation that is clear and consistent.

def calculate_simple_interest(principal, rate, time): """ Compute simple interest based on principal amount, annual interest rate, and time period.

```
Parameters:
    principal (float): Initial amount of money
    rate (float): Interest rate percentage per year
    time (float): Time duration in years

Returns:
    float: Calculated simple interest
"""
return (principal * rate * time) / 100
```

p = float(input("Enter principal: ")) r = float(input("Enter rate: ")) t = float(input("Enter time: ")) print("Simple Interest:", calculate_simple_interest(p, r, t))

```python
def calculate_simple_interest(principal, rate, time):
    """
    Compute simple interest based on principal amount,
    annual interest rate, and time period.

    Parameters:
        principal (float): Initial amount of money
        rate (float): Interest rate percentage per year
        time (float): Time duration in years

    Returns:
        float: Calculated simple interest
    """
    return (principal * rate * time) / 100


p = float(input("Enter principal: "))
r = float(input("Enter rate: "))
t = float(input("Enter time: "))
print("Simple Interest:", calculate_simple_interest(p, r, t))
```

Task Description -5 (Documentation – Prompt Detail Impact Study)

Task:
Study the impact of prompt detail on AI-generated documentation quality.

Instructions:

Create two prompts: one brief and one detailed.

- Use both prompts to document the same Python function.
- Compare the generated outputs.

Expected Output -5:
A comparison table highlighting differences in completeness, clarity, and accuracy of documentation.