

Assignment-9.2

Name: Adepu Tejaswini

Ht.no: 2303A51423

Batch. No: 21

Task Description -1 (Documentation – Function Summary Generation)

Prompt:

add a short functional summary as a docstring for every function in this file. The summary must explain the function purpose only, without describing logic or implementation steps.

Code:

```
def is_prime(num):  
    """Check if a number is prime."""  
    if num <= 1:  
        return False  
  
    for i in range(2, int(num**0.5) + 1):  
        if num % i == 0:  
            return False  
    return True  
number = int(input("Enter a number: "))  
if is_prime(number):  
    print(number, "is a Prime Number")  
else:  
    print(number, "is not a Prime Number")
```

Output:

```

PS C:\Users\AdepuTejaswini\AI assist> python -u "c:\Users\AdepuTejaswini\AI assist\code-9.py"
● Enter a number: 1
1 is not a Prime Number
PS C:\Users\AdepuTejaswini\AI assist> python -u "c:\Users\AdepuTejaswini\AI assist\code-9.py"
● Enter a number: 0
0 is not a Prime Number
PS C:\Users\AdepuTejaswini\AI assist> python -u "c:\Users\AdepuTejaswini\AI assist\code-9.py"
● Enter a number: 2
2 is a Prime Number
❖ PS C:\Users\AdepuTejaswini\AI assist>

```

Observation:

AI-generated function summaries help improve code readability by clearly describing the purpose of each function. The summaries make it easier for developers to understand the program structure without analyzing the implementation details. This approach also supports faster maintenance and documentation consistency.

Task Description -2 (Documentation – Logical Explanation for

Conditions and Loops)

Prompt:

add a brief explanation of the logic behind the each condition and loop statements. Explain only decision - making and loop statements

Code:

```

def is_prime(num):
    """Check if a number is prime."""
    # Check if the number is less than or equal to 1, as prime numbers are greater than 1
    if num <= 1:
        return False

    # Loop from 2 to the square root of the number to check for factors
    for i in range(2, int(num**0.5) + 1):
        # If the number is divisible by any of these factors, it is not prime
        if num % i == 0:
            return False
    # If no factors are found, the number is prime
    return True

number = int(input("Enter a number: "))
if is_prime(number):
    print(number, "is a Prime Number")
else:
    print(number, "is not a Prime Number")

```

Output:

```

PS C:\Users\AdepuTejaswini\AI assist> python -u "c:\Users\AdepuTejaswini\AI assist\code-9.py"
• Enter a number: 100
100 is not a Prime Number
PS C:\Users\AdepuTejaswini\AI assist> python -u "c:\Users\AdepuTejaswini\AI assist\code-9.py"
• Enter a number: -10
-10 is not a Prime Number
• PS C:\Users\AdepuTejaswini\AI assist> python -u "c:\Users\AdepuTejaswini\AI assist\code-9.py"
Enter a number: -2
-2 is not a Prime Number
PS C:\Users\AdepuTejaswini\AI assist> python -u "c:\Users\AdepuTejaswini\AI assist\code-9.py"
• Enter a number: 11
11 is a Prime Number
❖ PS C:\Users\AdepuTejaswini\AI assist>

```

Observation:

AI-generated explanations for conditions and loops clarify the decision-making flow and iteration behavior within the program. These logical comments help users quickly understand how the program controls execution paths. As a result, debugging and code comprehension become more efficient.

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

Prompt:

Generate a file summarizing the overall functionality of the script, including the purpose of the main function and how it interacts with the user.

Code:

```

"""This script defines a function called is_prime that checks whether a given number is a prime number or not.
The main function prompts the user to input a number, then calls the is_prime function with the user's input.
Based on the return value of the is_prime function, it prints out whether the number is a prime number or not.
The script uses basic control flow statements to determine the primality of the number and provides feedback to the user accordingly.
"""
def is_prime(num):
    # Check if a number is prime.
    if num <= 1:
        return False

    for i in range(2, int(num**0.5) + 1):
        if num % i == 0:
            return False
    return True

number = int(input("Enter a number: "))
if is_prime(number):
    print(number, "is a Prime Number")
else:
    print(number, "is not a Prime Number")

```

Output:

```

PS C:\Users\AdepuTejaswini> python -u "c:\Users\AdepuTejaswini\AI assist\code-9.py"
● Enter a number: 0
0 is not a Prime Number
PS C:\Users\AdepuTejaswini> python -u "c:\Users\AdepuTejaswini\AI assist\code-9.py"
● Enter a number: -2
-2 is not a Prime Number
PS C:\Users\AdepuTejaswini> python -u "c:\Users\AdepuTejaswini\AI assist\code-9.py"
● Enter a number: 11
11 is a Prime Number
❖ PS C:\Users\AdepuTejaswini>

```

Script Summary:

This script defines a function called `is_prime` that checks whether a given number is a prime number or not.

The main function prompts the user to input a number, then calls the `is_prime` function with the user's input.

Based on the return value of the `is_prime` function, it prints out whether the number is a prime number or not.

The script uses basic control flow statements to determine the primality of the number and provides feedback to the user accordingly.

Observation:

The AI successfully generated a concise file-level overview summarizing the overall purpose and functionality of the Python program. The overview was placed at the beginning of the file and described the program at a conceptual level without including implementation details. This improved the readability of the code and helped users quickly understand the program's objective.

Task Description -4 (Documentation – Refine Existing Documentation)

Prompt:

Rewrite the existing comments to improve clarity and consistency while keeping the original meaning unchanged. Do not modify the program logic.

Code:

```
def is_prime(num):
    """Check if a number is prime."""
    # A prime number is defined as a number greater than 1 that has no positive divisors other than 1 and itself.
    if num <= 1:
        return False

    # Check for factors from 2 up to the square root of the number.
    for i in range(2, int(num**0.5) + 1):
        # If the number is divisible by any of these factors, it is not prime.
        if num % i == 0:
            return False
    # If no factors are found, the number is prime.
    return True
number = int(input("Enter a number: "))
if is_prime(number):
    print(number, "is a Prime Number")
else:
    print(number, "is not a Prime Number")
```

Output:

```
PS C:\Users\AdepuTejaswini> python -u "c:\Users\AdepuTejaswini\AI assist\code-9.py"
Enter a number: -111
-111 is not a Prime Number
PS C:\Users\AdepuTejaswini> python -u "c:\Users\AdepuTejaswini\AI assist\code-9.py"
Enter a number: 65
65 is not a Prime Number
PS C:\Users\AdepuTejaswini> python -u "c:\Users\AdepuTejaswini\AI assist\code-9.py"
Enter a number: 7
7 is a Prime Number
PS C:\Users\AdepuTejaswini> █
```

Observation:

The AI improved the clarity and consistency of the existing comments without altering the program's functionality. The refined documentation provided clearer explanations while preserving the original technical meaning. This enhancement made the code easier to read, understand, and maintain.

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

Brief Prompt:

Write a short one-line documentation comment describing the purpose of the following function.

Code:

```
def Factorial(num):
    """Calculate the factorial of a given number."""
    if num == 0:
        return 1
    else:
        return num * Factorial(num-1)
number = int(input("Enter a number: "))
print("Factorial of", number, "is", Factorial(number))
```

Output:

```
PS C:\Users\AdepuTejaswini> python -u "c:\Users\AdepuTejaswini\AI assist\code-9.py"
Enter a number: 0
Factorial of 0 is 1
PS C:\Users\AdepuTejaswini> python -u "c:\Users\AdepuTejaswini\AI assist\code-9.py"
Enter a number: 5
Factorial of 5 is 120
PS C:\Users\AdepuTejaswini> python -u "c:\Users\AdepuTejaswini\AI assist\code-9.py"
Enter a number: 1
Factorial of 1 is 1
PS C:\Users\AdepuTejaswini>
```

Detailed Prompt:

Write a clear 3–5 line documentation comment for the following function describing its purpose, inputs, outputs, and overall behavior.

Code:

```
def Factorial(num):
    """
    Calculate the factorial of a given number.

    This function takes a non-negative integer as input and returns its factorial.
    The factorial of a number n is the product of all positive integers less than or equal to n.
    If the input number is 0, the function returns 1, as 0! is defined to be 1.

    Parameters:
    num (int): A non-negative integer for which the factorial is to be calculated.

    Returns:
    int: The factorial of the input number.
    """
    if num == 0:
        return 1
    else:
        return num * Factorial(num-1)
number = int(input("Enter a number: "))
print("Factorial of", number, "is", Factorial(number))
```

Output:

[illegible]

Observation:

Criteria	Brief Prompt Output (One-line Comment)	Detailed Prompt Output (3–5 line Docstring)
Completeness	Provides only the basic purpose of the function.	Includes purpose, input description, output, and behavior details.
Clarity	Clear but very minimal explanation.	More descriptive, structured, and easier to understand.
Accuracy	Correct but limited technical information.	Highly accurate with full technical context and parameter details.