

Assignment-9.2

Name: J.akshaya

Ht.no: 2303A51629

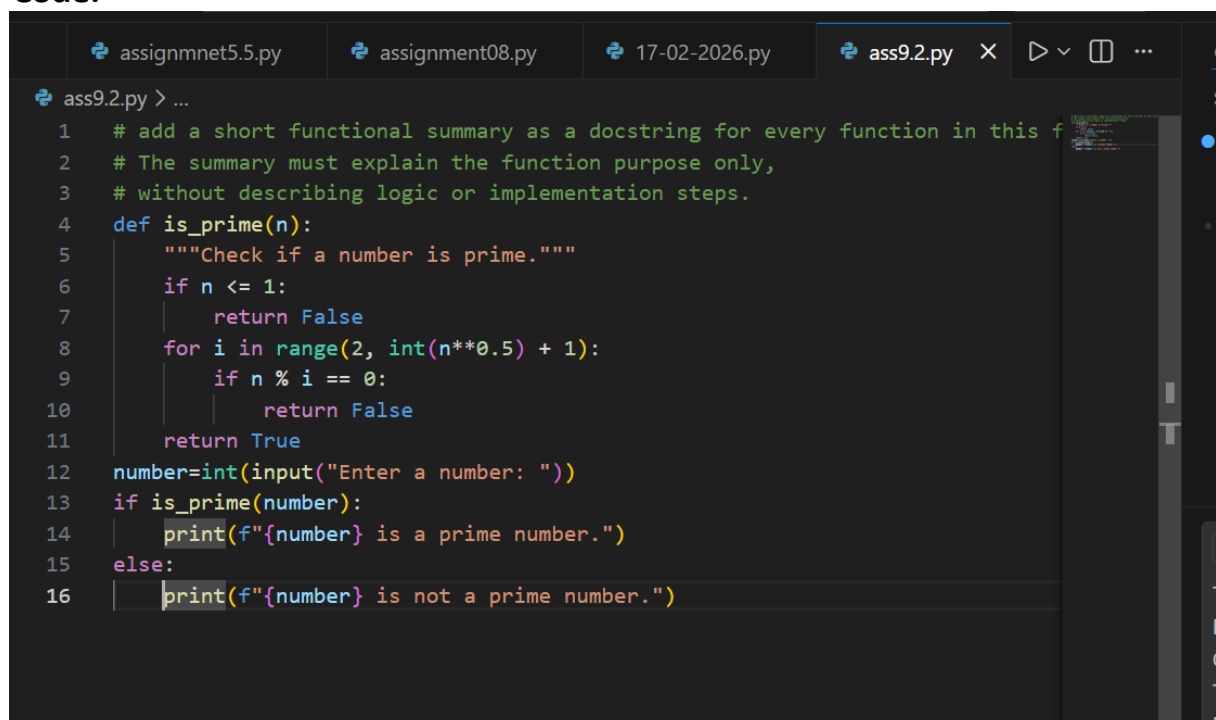
Batch. No: 22

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:

A screenshot of a code editor window with a dark theme. The editor has several tabs at the top: 'ass9.2.py', 'assignment08.py', '17-02-2026.py', and 'ass9.2.py' (selected). The selected tab shows a Python script. The script starts with a comment block: '# add a short functional summary as a docstring for every function in this f', '# The summary must explain the function purpose only,', and '# without describing logic or implementation steps.'. Below this is a function definition 'def is_prime(n):' with a docstring '"""Check if a number is prime."""'. The function body contains logic to check if a number is prime: 'if n <= 1: return False', 'for i in range(2, int(n**0.5) + 1):', 'if n % i == 0: return False', and 'return True'. After the function, there is a line 'number=int(input("Enter a number: "))', followed by an 'if' statement 'if is_prime(number):' which prints 'number is a prime number.', and an 'else' block which prints 'number is not a prime number.'. The line numbers 1 through 16 are visible on the left side of the code editor.

```
1 # add a short functional summary as a docstring for every function in this f
2 # The summary must explain the function purpose only,
3 # without describing logic or implementation steps.
4 def is_prime(n):
5     """Check if a number is prime."""
6     if n <= 1:
7         return False
8     for i in range(2, int(n**0.5) + 1):
9         if n % i == 0:
10            return False
11    return True
12 number=int(input("Enter a number: "))
13 if is_prime(number):
14     print(f"{number} is a prime number.")
15 else:
16     print(f"{number} is not a prime number.")
```

Output:

```

/Local/WindowsApps/python3.13.exe -c:/Users/aksha/OneDrive/Desktop/
assisantcoding - Copy/ass9.2.py"
PS C:\Users\aksha\OneDrive\Desktop\assisantcoding - Copy> & C:/Users/aksha/AppData/Local/Microsoft/WindowsApps/python3.13.exe "c:/Users/aksha/OneDrive/Desktop/assisantcoding - Copy/ass9.2.py"
Enter a number: 10
10 is not a prime number.
PS C:\Users\aksha\OneDrive\Desktop\assisantcoding - Copy> & C:/Users/aksha/AppData/Local/Microsoft/WindowsApps/python3.13.exe "c:/Users/aksha/OneDrive/Desktop/assisantcoding - Copy/ass9.2.py"
Enter a number: 15
15 is not a prime number.
PS C:\Users\aksha\OneDrive\Desktop\assisantcoding - Copy>

```

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:

```

16 # print(f'{number} is not a prime number.')
17
18 # add a brief explanation of the logic behind the each condition
19 # and loop statements. Explain only decision - making and
20 # loop statements
21 # Code:
22 def is_prime(n):
23     """Check if a number is prime."""
24     if n <= 1:
25         return False # A prime number must be greater than 1
26     for i in range(2, int(n**0.5) + 1):
27         if n % i == 0:
28             return False # If n is divisible by any number other than 1 and
29             return True # If n is not divisible by any number, it is prime
30 number = int(input("Enter a number: "))
31 if is_prime(number):
32     print(f"{number} is a prime number.") # If the function returns True, t
33
34

```

Output:

```
pData/Local/Microsoft/WindowsApps/python3.13.exe "c:/Users/aksha/OneDrive/Des
ktop/assisantcoding - Copy/ass9.2.py"
Enter a number: 10
10 is not a prime number.
PS C:\Users\aksha\OneDrive\Desktop\assisantcoding - Copy> & C:/Users/aksha/Ap
pData/Local/Microsoft/WindowsApps/python3.13.exe "c:/Users/aksha/OneDrive/Des
ktop/assisantcoding - Copy/ass9.2.py"
Enter a number: 15
15 is not a prime number.
PS C:\Users\aksha\OneDrive\Desktop\assisantcoding - Copy> & C:/Users/aksha/Ap
pData/Local/Microsoft/WindowsApps/python3.13.exe "c:/Users/aksha/OneDrive/Des
ktop/assisantcoding - Copy/ass9.2.py"
Enter a number: 17
17 is a prime number.
PS C:\Users\aksha\OneDrive\Desktop\assisantcoding - Copy> & C:/Users/aksha/Ap
pData/Local/Microsoft/WindowsApps/python3.13.exe "c:/Users/aksha/OneDrive/Des
ktop/assisantcoding - Copy/ass9.2.py"
Enter a number: 25
PS C:\Users\aksha\OneDrive\Desktop\assisantcoding - Copy>
PS C:\Users\aksha\OneDrive\Desktop\assisantcoding - Copy> █
```

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:

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

def is_prime(n):
    """Check if a number is prime."""
    if n <= 1:
        return False # A prime number must be greater than 1
    for i in range(2, int(n**0.5) + 1):
        if n % i == 0:
            return False # If n is divisible by any number other than 1 and
    return True # If n is not divisible by any number, it is prime
number = int(input("Enter a number: "))
if is_prime(number):
    print(f"{number} is a prime number.") # If the function returns True, t
else:
    print(f"{number} is not a prime number.") # If the function returns Fal
```

Output:

```

Enter a number: 17
17 is a prime number.
PS C:\Users\aksha\OneDrive\Desktop\assisantcoding - Copy> & C:/Users/aksha/AppData/Local/Microsoft/WindowsApps/python3.13.exe "c:/Users/aksha/OneDrive/Desktop/assisantcoding - Copy/ass9.2.py"
Enter a number: 25
PS C:\Users\aksha\OneDrive\Desktop\assisantcoding - Copy>
PS C:\Users\aksha\OneDrive\Desktop\assisantcoding - Copy> & C:/Users/aksha/AppData/Local/Microsoft/WindowsApps/python3.13.exe "c:/Users/aksha/OneDrive/Desktop/assisantcoding - Copy/ass9.2.py"
Enter a number: 12
12 is not a prime number.
PS C:\Users\aksha\OneDrive\Desktop\assisantcoding - Copy> & C:/Users/aksha/AppData/Local/Microsoft/WindowsApps/python3.13.exe "c:/Users/aksha/OneDrive/Desktop/assisantcoding - Copy/ass9.2.py"
Enter a number: -2
-2 is not a prime number.
PS C:\Users\aksha\OneDrive\Desktop\assisantcoding - Copy> & C:/Users/aksha/AppData/Local/Microsoft/WindowsApps/python3.13.exe "c:/Users/aksha/OneDrive/Desktop/assisantcoding - Copy/ass9.2.py"
Enter a number: 4
4 is not a prime number.
PS C:\Users\aksha\OneDrive\Desktop\assisantcoding - Copy>

```

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:

```
# 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(n):
    """Check if a number is prime."""
    if n <= 1:
        return False # A prime number must be greater than 1
    for i in range(2, int(n**0.5) + 1):
        if n % i == 0:
            return False # If n is divisible by any number other than 1 and
    return True # If n is not divisible by any number, it is prime
number = int(input("Enter a number: "))
if is_prime(number):
    print(f"{number} is a prime number.") # If the function returns True, t
else:
    print(f"{number} is not a prime number.") # If the function returns Fal
```

Output:

```
Enter a number: -2
-2 is not a prime number.
S C:\Users\aksha\OneDrive\Desktop\assisantcoding - Copy> &
Data/Local/Microsoft/WindowsApps/python3.13.exe "c:/Users,
top/assisantcoding - Copy/ass9.2.py"
Enter a number: 4
4 is not a prime number.
S C:\Users\aksha\OneDrive\Desktop\assisantcoding - Copy> &
Data/Local/Microsoft/WindowsApps/python3.13.exe "c:/Users,
top/assisantcoding - Copy/ass9.2.py"
Enter a number: 20
20 is not a prime number.
S C:\Users\aksha\OneDrive\Desktop\assisantcoding - Copy>
```

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(n):
    """Calculate the factorial of a number."""
    if n < 0:
        return None # Factorial is not defined for negative numbers
    elif n == 0 or n == 1:
        return 1 # The factorial of 0 and 1 is 1
    else:
        result = 1
        for i in range(2, n + 1):
            result *= i # Multiply result by each number from 2 to n
        return result
number = int(input("Enter a number: "))
print(f"The factorial of {number} is {factorial(number)}.") # Print the
```

Output:

```
pData/Local/Microsoft/WindowsApps/python3.13.exe - C:/Users/aksha/OneDrive/Des
ktop/assisantcoding - Copy/ass9.2.py"
Enter a number: 5
The factorial of 5 is 120.
PS C:\Users\aksha\OneDrive\Desktop\assisantcoding - Copy> & C:/Users/aksha/App
pData/Local/Microsoft/WindowsApps/python3.13.exe "c:/Users/aksha/OneDrive/Des
ktop/assisantcoding - Copy/ass9.2.py"
Enter a number: 10
The factorial of 10 is 3628800.
PS C:\Users\aksha\OneDrive\Desktop\assisantcoding - Copy> & C:/Users/aksha/App
pData/Local/Microsoft/WindowsApps/python3.13.exe "c:/Users/aksha/OneDrive/Des
ktop/assisantcoding - Copy/ass9.2.py"
Enter a number: 6
The factorial of 6 is 720.
PS C:\Users\aksha\OneDrive\Desktop\assisantcoding - Copy> & C:/Users/aksha/App
pData/Local/Microsoft/WindowsApps/python3.13.exe "c:/Users/aksha/OneDrive/Des
ktop/assisantcoding - Copy/ass9.2.py"
Enter a number: 12
The factorial of 12 is 479001600.
PS C:\Users\aksha\OneDrive\Desktop\assisantcoding - Copy> |
```

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:

```
PS C:\Users\aksha> & C:/Users/aksha/AppData/Local/Microsoft/Win  
dowsApps/python3.13.exe "c:/Users/aksha/OneDrive/Desktop/assisantc  
oding - Copy/ass9.2.py"  
Enter a number: 0  
The factorial of 0 is 1.  
PS C:\Users\aksha\OneDrive\Desktop\assisantcoding - Copy> & C:/Use  
rs/aksha/AppData/Local/Microsoft/WindowsApps/python3.13.exe "c:/Us  
ers/aksha/OneDrive/Desktop/assisantcoding - Copy/ass9.2.py"  
Enter a number: 1  
The factorial of 1 is 1.  
PS C:\Users\aksha\OneDrive\Desktop\assisantcoding - Copy> & C:/Use  
rs/aksha/AppData/Local/Microsoft/WindowsApps/python3.13.exe "c:/Us  
ers/aksha/OneDrive/Desktop/assisantcoding - Copy/ass9.2.py"  
Enter a number: 160  
The factorial of 160 is 471472363599206132240694321176194377951192  
623045460204976904578317542573467421580346978030238114995699562728  
104819596262106947389303901748942909887857509625114880781313585012  
959529941660203611234871833992565791817698209861793313332044734813  
70009600000000000000000000000000000000000000000000000000000000000000.  
PS C:\Users\aksha\OneDrive\Desktop\assisantcoding - Copy>
```

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.