

Assignment 3.1

2303A510f2 – Anitha

Batch - 14

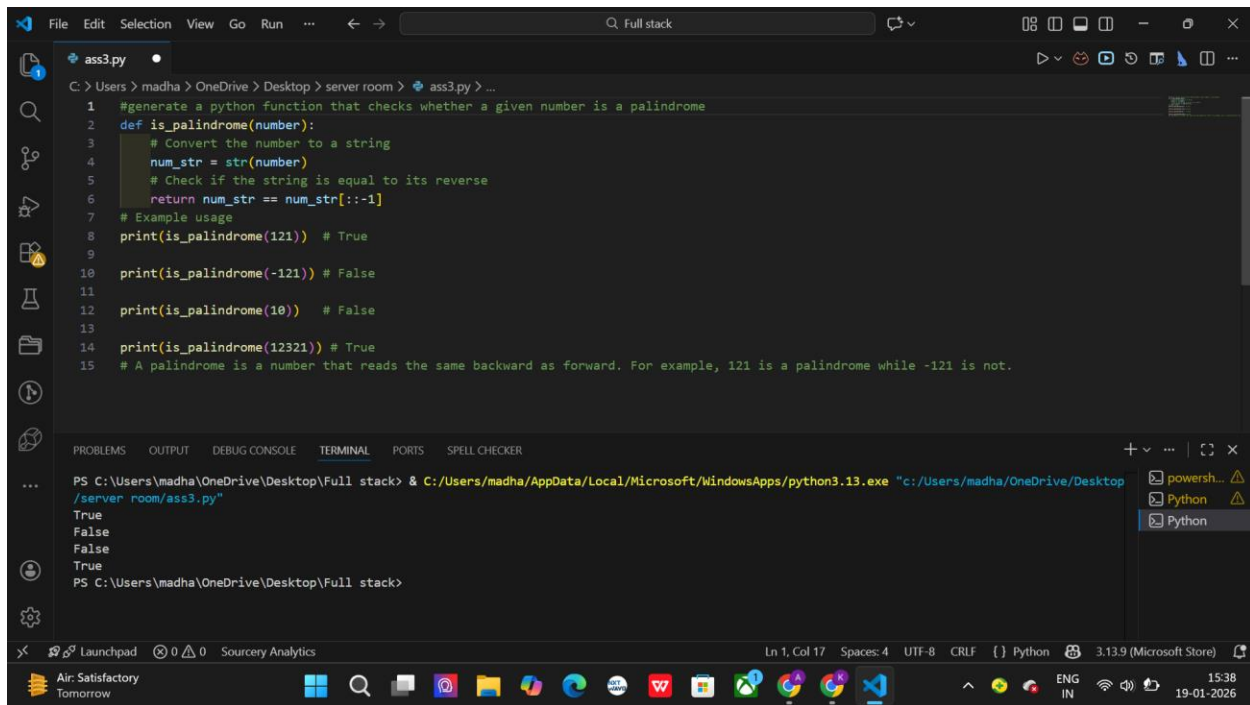
Lab Experiment: Prompt Engineering – Improving Prompts and Context Management

Question 1: Zero-Shot Prompting (Palindrome Number Program)

Prompt:

Write a Python function that checks whether a given number is a palindrome and returns True or False.

Code & Output:



The screenshot shows a Visual Studio Code editor window with a file named `ass3.py` open. The file contains a Python function `is_palindrome` that checks if a number is a palindrome. The function converts the number to a string, reverses it, and compares it to the original. The terminal output shows the function being called with various numbers: 121 (True), -121 (False), 10 (False), and 12321 (True). A comment at the end explains that a palindrome reads the same backward as forward.

```
1 #generate a python function that checks whether a given number is a palindrome
2 def is_palindrome(number):
3     # Convert the number to a string
4     num_str = str(number)
5     # Check if the string is equal to its reverse
6     return num_str == num_str[::-1]
7 # Example usage
8 print(is_palindrome(121)) # True
9
10 print(is_palindrome(-121)) # False
11
12 print(is_palindrome(10)) # False
13
14 print(is_palindrome(12321)) # True
15 # A palindrome is a number that reads the same backward as forward. For example, 121 is a palindrome while -121 is not.
```

Terminal Output:

```
PS C:\Users\madha\OneDrive\Desktop\Full stack> C:/Users/madha/AppData/Local/Microsoft/WindowsApps/python3.13.exe "c:/Users/madha/OneDrive/Desktop/server room/ass3.py"
True
False
False
True
PS C:\Users\madha\OneDrive\Desktop\Full stack>
```

Explanation:

- Define a function called `is_palindrome_number` that takes a single argument `num`.
- Convert the number to a string using `str(num)`
- Reverse the string using `[::-1]`
- Compare the original string with the reversed string using `==`
- Return the result of the comparison
- Print the result of the function call

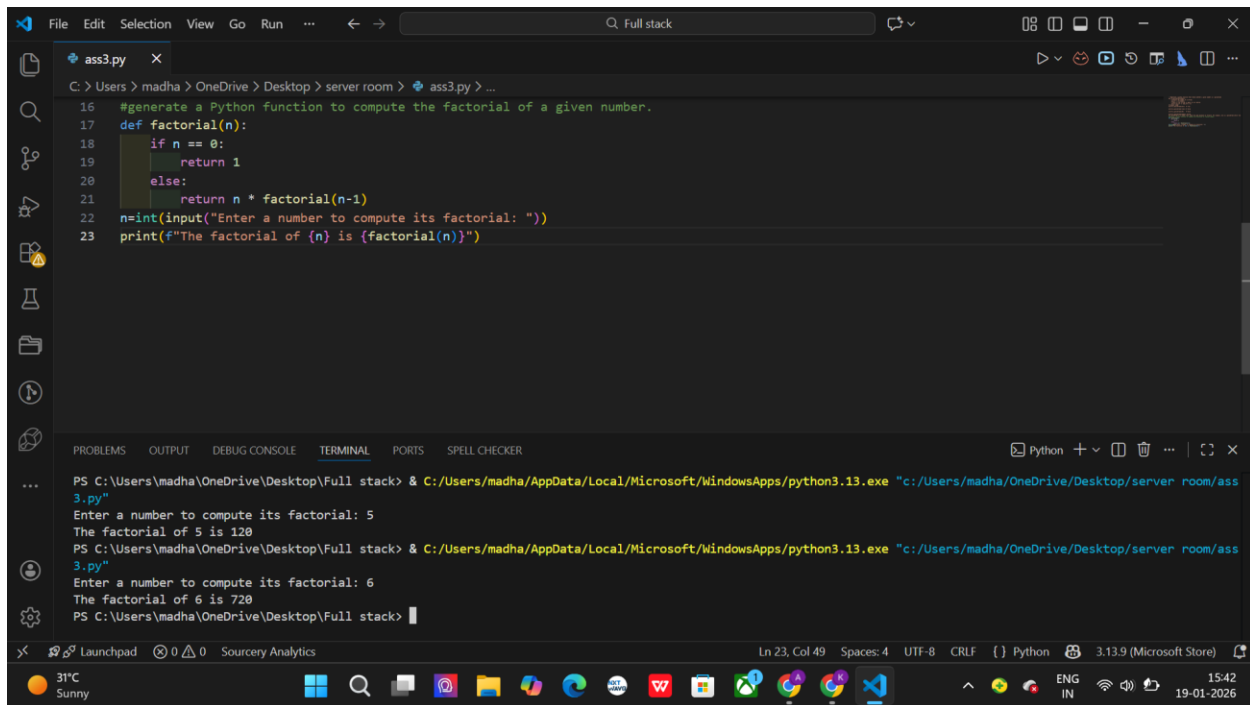
Question 2: One-Shot Prompting (Factorial Calculation)

Prompt:

Example: Input: 5 → Output: 120

Using this example, write a Python function to compute the factorial of a given number.

Code & Output :



The screenshot shows a Visual Studio Code editor window with a file named `ass3.py` open. The file contains a Python function `factorial(n)` that calculates the factorial of a given number `n`. The function uses a recursive approach: if `n` is 0, it returns 1; otherwise, it returns `n * factorial(n-1)`. Below the function definition, there is a prompt for user input and a print statement to display the result.

```
16 #generate a Python function to compute the factorial of a given number.
17 def factorial(n):
18     if n == 0:
19         return 1
20     else:
21         return n * factorial(n-1)
22 n=int(input("Enter a number to compute its factorial: "))
23 print(f"The factorial of {n} is {factorial(n)}")
```

The terminal window at the bottom shows the execution of the script. It prompts the user to enter a number, and the output shows the factorial of 5 is 120 and the factorial of 6 is 720.

```
PS C:\Users\madha\OneDrive\Desktop\Full stack & C:/Users/madha/AppData/Local/Microsoft/WindowsApps/python3.13.exe "c:/Users/madha/OneDrive/Desktop/server room/ass3.py"
Enter a number to compute its factorial: 5
The factorial of 5 is 120
PS C:\Users\madha\OneDrive\Desktop\Full stack & C:/Users/madha/AppData/Local/Microsoft/WindowsApps/python3.13.exe "c:/Users/madha/OneDrive/Desktop/server room/ass3.py"
Enter a number to compute its factorial: 6
The factorial of 6 is 720
PS C:\Users\madha\OneDrive\Desktop\Full stack > |
```

Explanation:

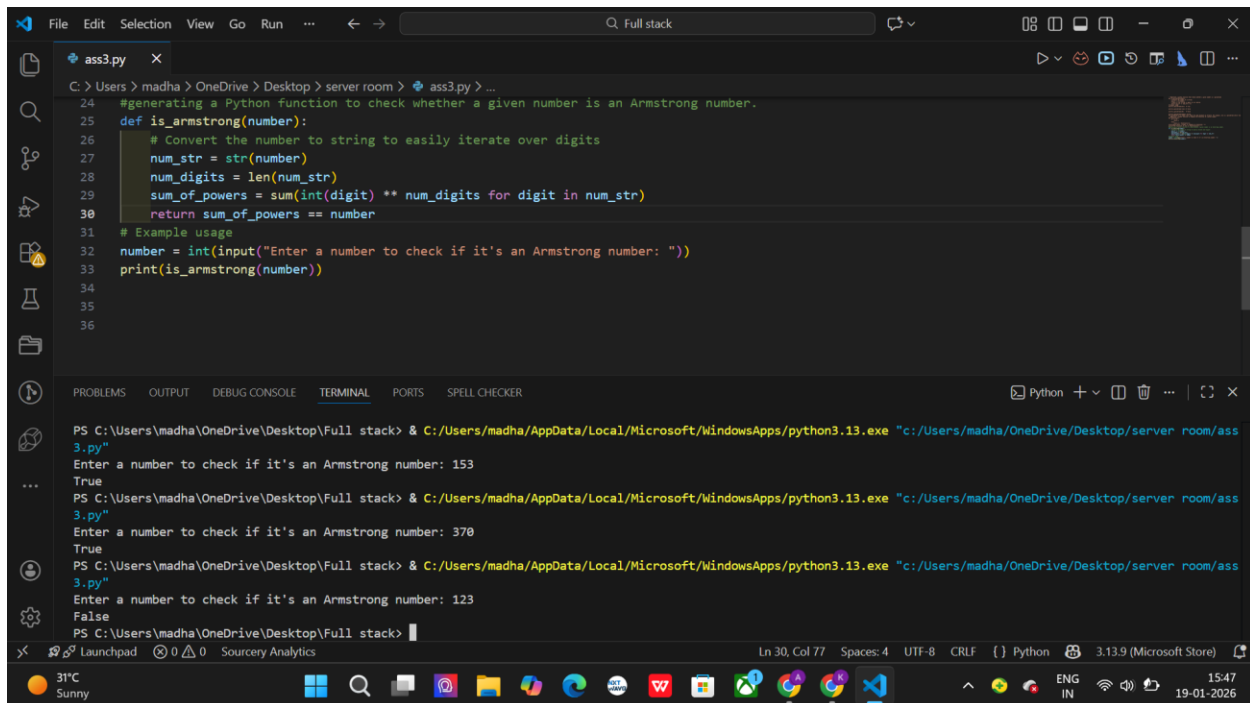
- Define a function called `factorial` that takes a single argument `n`.
- If `n` is 0, return 1
- Else, return `n * factorial(n-1)`
- Print the result of the function call

Question 3: Few-Shot Prompting (Armstrong Number Check)

Prompt:

Write a Python function to check whether a number is an Armstrong number.

Code & Output:



The screenshot shows a Visual Studio Code editor window with a file named `ass3.py` open. The code defines a function `is_armstrong(number)` that checks if a number is an Armstrong number. The function converts the number to a string, iterates over its digits, calculates the sum of their cubes, and compares it to the original number. Below the code, the terminal shows the execution of the script, which prompts the user to enter a number and prints the result (True or False) for three different inputs: 153, 370, and 123.

```
24 #generating a Python function to check whether a given number is an Armstrong number.
25 def is_armstrong(number):
26     # Convert the number to string to easily iterate over digits
27     num_str = str(number)
28     num_digits = len(num_str)
29     sum_of_powers = sum(int(digit) ** num_digits for digit in num_str)
30     return sum_of_powers == number
31 # Example usage
32 number = int(input("Enter a number to check if it's an Armstrong number: "))
33 print(is_armstrong(number))
34
35
36
```

Terminal Output:

```
PS C:\Users\madha\OneDrive\Desktop\Full stack> & C:/Users/madha/AppData/Local/Microsoft/WindowsApps/python3.13.exe "c:/Users/madha/OneDrive/Desktop/server room/ass3.py"
Enter a number to check if it's an Armstrong number: 153
True
PS C:\Users\madha\OneDrive\Desktop\Full stack> & C:/Users/madha/AppData/Local/Microsoft/WindowsApps/python3.13.exe "c:/Users/madha/OneDrive/Desktop/server room/ass3.py"
Enter a number to check if it's an Armstrong number: 370
True
PS C:\Users\madha\OneDrive\Desktop\Full stack> & C:/Users/madha/AppData/Local/Microsoft/WindowsApps/python3.13.exe "c:/Users/madha/OneDrive/Desktop/server room/ass3.py"
Enter a number to check if it's an Armstrong number: 123
False
PS C:\Users\madha\OneDrive\Desktop\Full stack>
```

Explanation:

- Define a function called `is_armstrong_number` that takes a single argument `num`.
- Initialize a variable called `sum` to 0.
- Initialize a variable called `temp` to `num`.
- Initialize a variable called `count` to the length of the number.
- While `temp` is greater than 0, do the following:
- Get the last digit of `temp` using `temp % 10`.
- Add the cube of the last digit to `sum`.
- Divide `temp` by 10 to remove the last digit.

Question 4: Context-Managed Prompting (Optimized Number Classification)

Prompt:

Write an optimized Python program that classifies a number as prime, composite, or neither. Ensure proper input validation and efficient logic.

Code & Output:

```
File Edit Selection View Go Run ... Full stack
ass3.py
C:\Users\madha> OneDrive\ Desktop\ server room> ass3.py
26 #generate an optimized Python program that classifies a number as prime, composite, or neither
27 def classify_number(n):
28     if n < 2:
29         return "neither prime nor composite"
30     is_prime = True
31     for i in range(2, int(n**0.5) + 1):
32         if n % i == 0:
33             is_prime = False
34             break
35     if is_prime:
36         return "prime"
37     else:
38         return "composite"
39 number = int(input("Enter a number to classify: "))
40 print(f"{number} is {classify_number(number)}")

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS SPELL CHECKER
Python

PS C:\Users\madha\OneDrive\Desktop\Full stack> C:/Users/madha/AppData/Local/Microsoft/WindowsApps/python3.13.exe "c:/Users/madha/OneDrive/Desktop/server room/ass3.py"
Enter a number to classify: 2
2 is prime
PS C:\Users\madha\OneDrive\Desktop\Full stack> C:/Users/madha/AppData/Local/Microsoft/WindowsApps/python3.13.exe "c:/Users/madha/OneDrive/Desktop/server room/ass3.py"
Enter a number to classify: 1
1 is neither prime nor composite
PS C:\Users\madha\OneDrive\Desktop\Full stack> C:/Users/madha/AppData/Local/Microsoft/WindowsApps/python3.13.exe "c:/Users/madha/OneDrive/Desktop/server room/ass3.py"
Enter a number to classify: 8
8 is composite
PS C:\Users\madha\OneDrive\Desktop\Full stack>
```

Explanation:

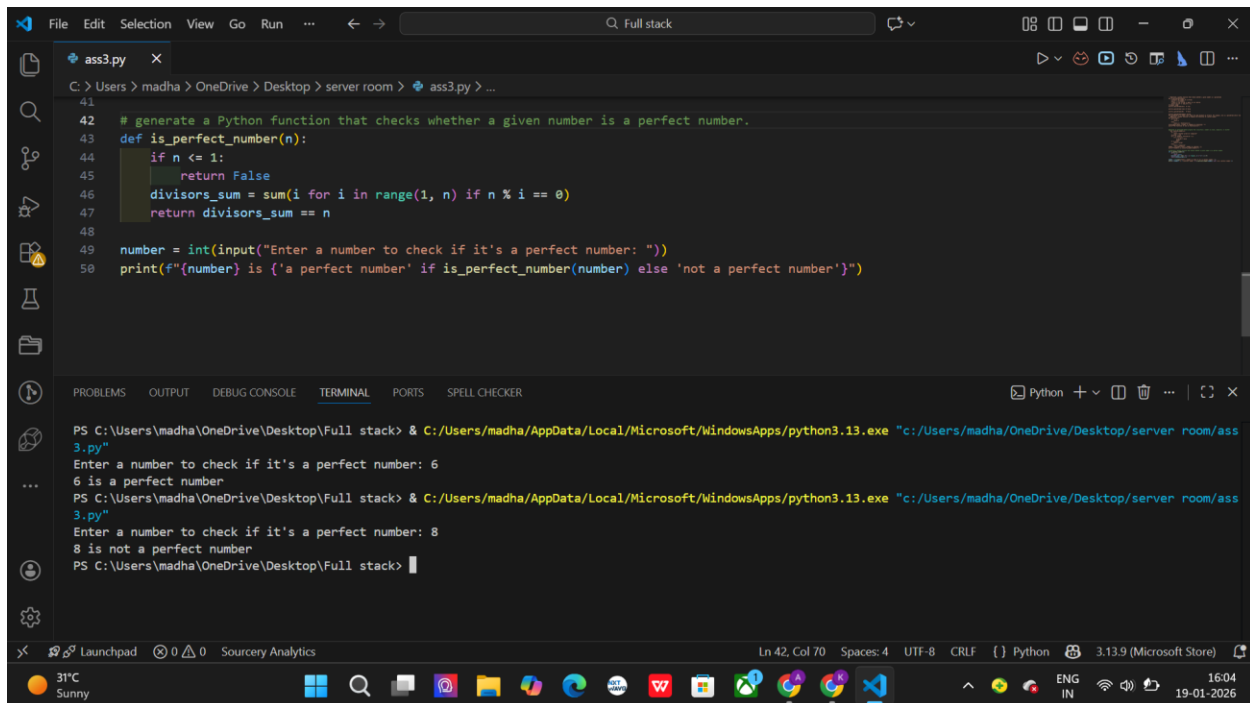
- Define a function called `classify_number` that takes a single argument `num`.
- If `num` is less than or equal to 1, return "Neither".
- For `i` in range (2, `int(num**0.5) + 1`), do the following:
- If `num` is divisible by `i`, return "Composite".
- Return "Prime".
- Print the result of the function call.

Question 5: Zero-Shot Prompting (Perfect Number Check)

Prompt:

Write a Python function to check whether a given number is a perfect number.

Code & Output:



The screenshot shows a Visual Studio Code editor window with a file named `ass3.py` open. The file contains a Python function `is_perfect_number(n)` that checks if a number is perfect. The function returns `False` for `n <= 1`, calculates the sum of divisors for `n > 1`, and returns `True` if the sum equals `n`. Below the editor, the terminal shows the execution of the script, where the user enters `6` and `8` to test the function. The output shows that `6` is a perfect number and `8` is not.

```
41
42 # generate a Python function that checks whether a given number is a perfect number.
43 def is_perfect_number(n):
44     if n <= 1:
45         return False
46     divisors_sum = sum(i for i in range(1, n) if n % i == 0)
47     return divisors_sum == n
48
49 number = int(input("Enter a number to check if it's a perfect number: "))
50 print(f"{number} is {'a perfect number' if is_perfect_number(number) else 'not a perfect number'}")
```

```
PS C:\Users\madha\OneDrive\Desktop\Full stack> C:/Users/madha/AppData/Local/Microsoft/WindowsApps/python3.13.exe "c:/Users/madha/OneDrive/Desktop/server room/ass3.py"
Enter a number to check if it's a perfect number: 6
6 is a perfect number
PS C:\Users\madha\OneDrive\Desktop\Full stack> C:/Users/madha/AppData/Local/Microsoft/WindowsApps/python3.13.exe "c:/Users/madha/OneDrive/Desktop/server room/ass3.py"
Enter a number to check if it's a perfect number: 8
8 is not a perfect number
PS C:\Users\madha\OneDrive\Desktop\Full stack>
```

Explanation:

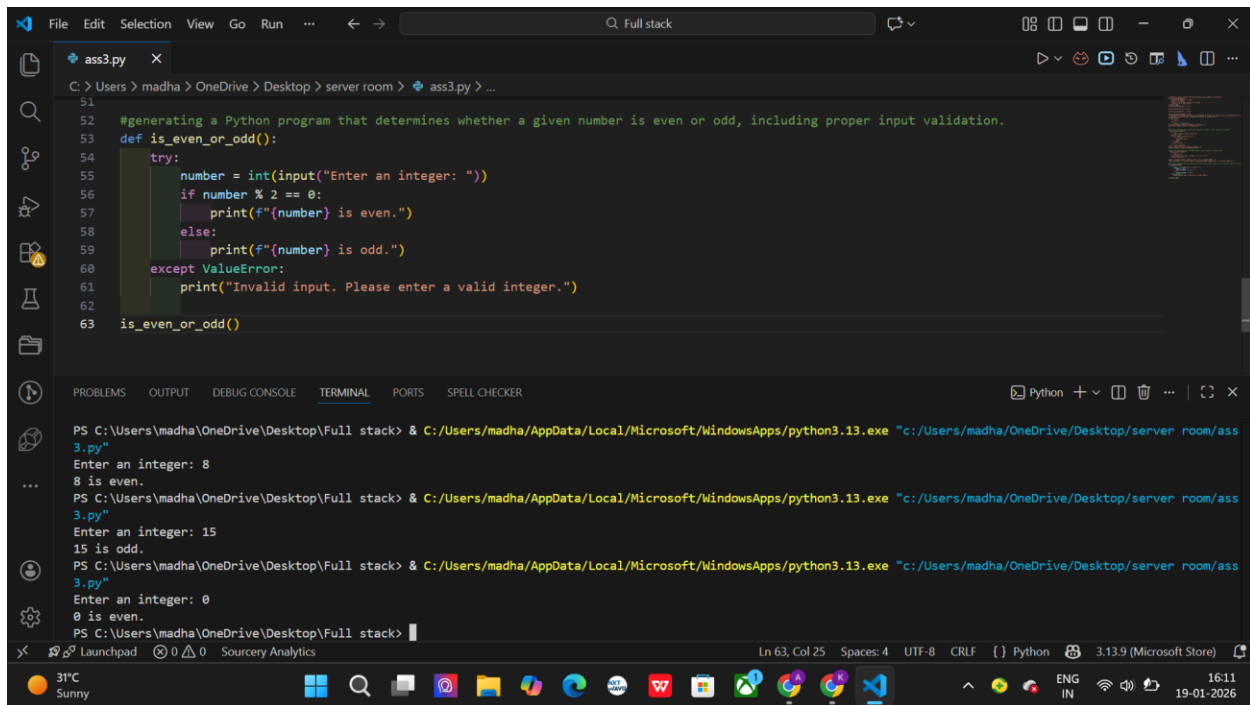
- Define a function called `is_perfect_number` that takes a single argument `num`.
- Initialize a variable called `sum` to 0.
- For `i` in range (1, `num`), do the following:
- If `num` is divisible by `i`, add `i` to `sum`.
- Return `True` if `sum` is equal to `num`, otherwise return `False`.
- Print the result of the function call.

Question 6: Few-Shot Prompting (Even or Odd Classification with Validation)

Prompt:

Write a Python program to classify a number as even or odd with proper validation.

Code & Output:



The screenshot shows a Visual Studio Code editor window with a Python file named `ass3.py` open. The file contains a function `is_even_or_odd()` that takes an integer input and prints whether it is even or odd, with proper input validation using a try-except block. The terminal at the bottom shows the execution of the script, with three test cases: 8 (even), 15 (odd), and 0 (even). The status bar at the bottom indicates the file is at line 63, column 25, and the Python version is 3.13.9.

```
51
52 #generating a Python program that determines whether a given number is even or odd, including proper input validation.
53 def is_even_or_odd():
54     try:
55         number = int(input("Enter an integer: "))
56         if number % 2 == 0:
57             print(f"{number} is even.")
58         else:
59             print(f"{number} is odd.")
60     except ValueError:
61         print("Invalid input. Please enter a valid integer.")
62
63 is_even_or_odd()
```

PS C:\Users\madha\OneDrive\Desktop\Full stack> C:/Users/madha/AppData/Local/Microsoft/WindowsApps/python3.13.exe "c:/Users/madha/OneDrive/Desktop/server room/ass3.py"

Enter an integer: 8
8 is even.

PS C:\Users\madha\OneDrive\Desktop\Full stack> C:/Users/madha/AppData/Local/Microsoft/WindowsApps/python3.13.exe "c:/Users/madha/OneDrive/Desktop/server room/ass3.py"

Enter an integer: 15
15 is odd.

PS C:\Users\madha\OneDrive\Desktop\Full stack> C:/Users/madha/AppData/Local/Microsoft/WindowsApps/python3.13.exe "c:/Users/madha/OneDrive/Desktop/server room/ass3.py"

Enter an integer: 0
0 is even.

PS C:\Users\madha\OneDrive\Desktop\Full stack>

Explanation:

- Define a function called `classify_number` that takes a single argument `num`.
- If `num` is divisible by 2, return "Even".
- Else, return "Odd".
- Print the result of the function call.
- Print the number and the result of the function call.