

# AI ASSISTANT CODING:

## LAB ASSIGNMENT -1.1

NAME:N.JYOSHNA SRI

HALL NO:2403A52106

### TASK-1

The screenshot displays the Visual Studio Code (VS Code) interface. The main editor window shows a Python file named `# Program to check if a number is an Arm.py`. The code is as follows:

```
1 #Program to calculate the factorial of a number using a loop (no function)
2
3 num = int(input("Enter a number: "))
4 factorial = 1
5
6 for i in range(1, num + 1):
7     factorial *= i
8
9 print(f"Factorial of {num} is {factorial}")
```

The bottom panel shows the TERMINAL output:

```
PS C:\2403a52106> & c:\Users\HP\AppData\Local\Programs\Python\Python313\python.exe "c:/2403a52106/# Program to check if a number is an Arm.py"
Enter a number: 23
Factorial of 23 is 25852016738884976640000
PS C:\2403a52106>
```

On the right side, the CHAT panel is open, displaying an explanation of the code:

**EXPLAIN THE CODE**

Used 1 reference

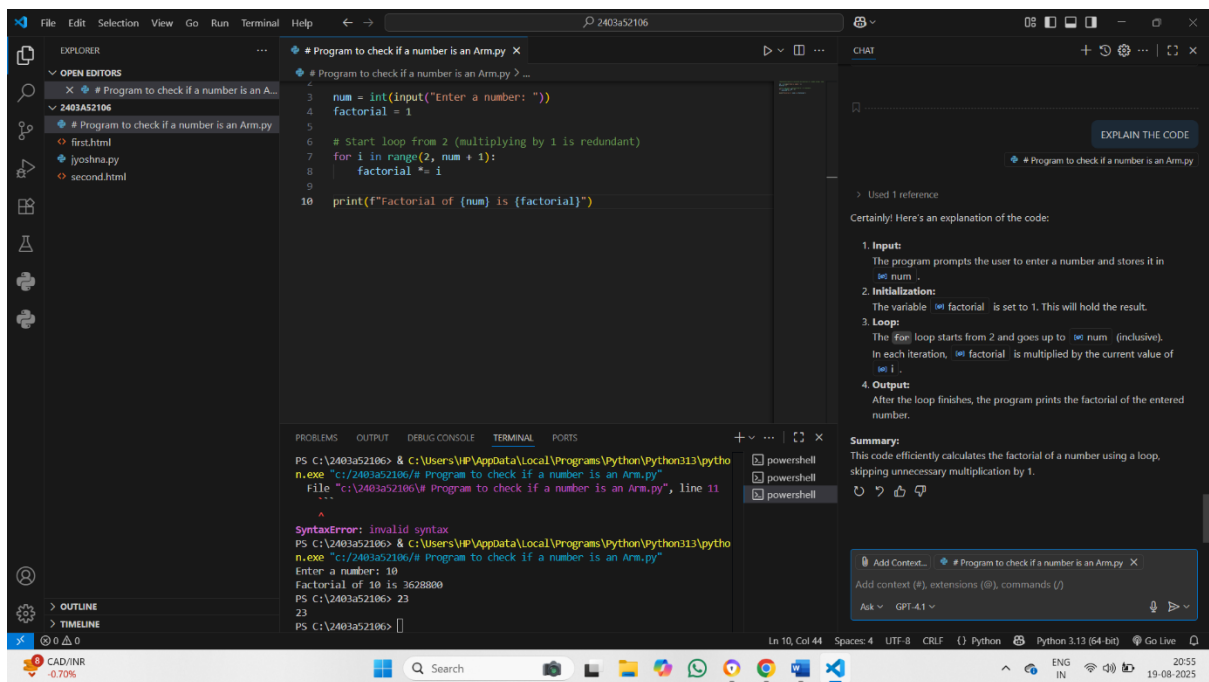
Certainly! Here's an explanation of the code:

- Input:** The program asks the user to enter a number and stores it in `num`.
- Initialization:** The variable `factorial` is set to 1. This will hold the result.
- Loop:** A `for` loop runs from 1 to `num` (inclusive). In each iteration, `factorial` is multiplied by the current value of `i`.
- Output:** After the loop, the program prints the factorial of the entered number.

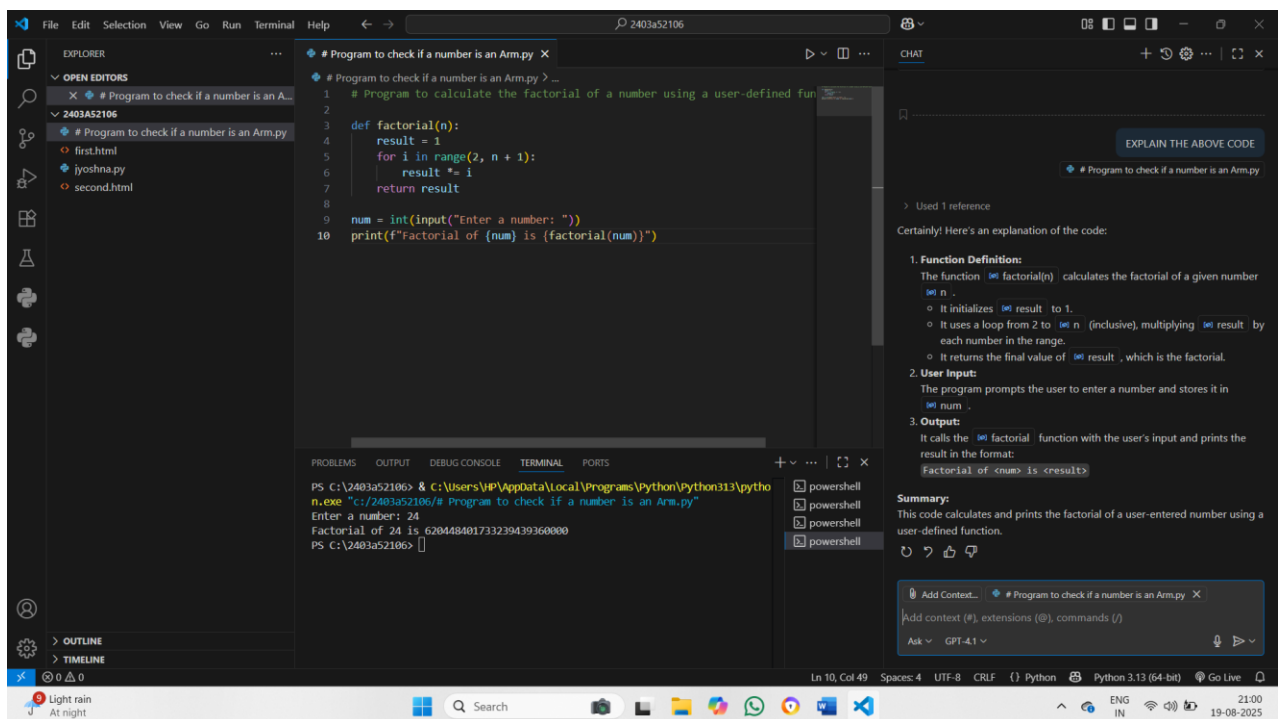
**Summary:** This code calculates the factorial of a number using a loop, without defining any functions.

At the bottom of the chat panel, there is an input field with the text "Add context (#), extensions (@), commands (/)" and a dropdown menu set to "GPT-4.1".

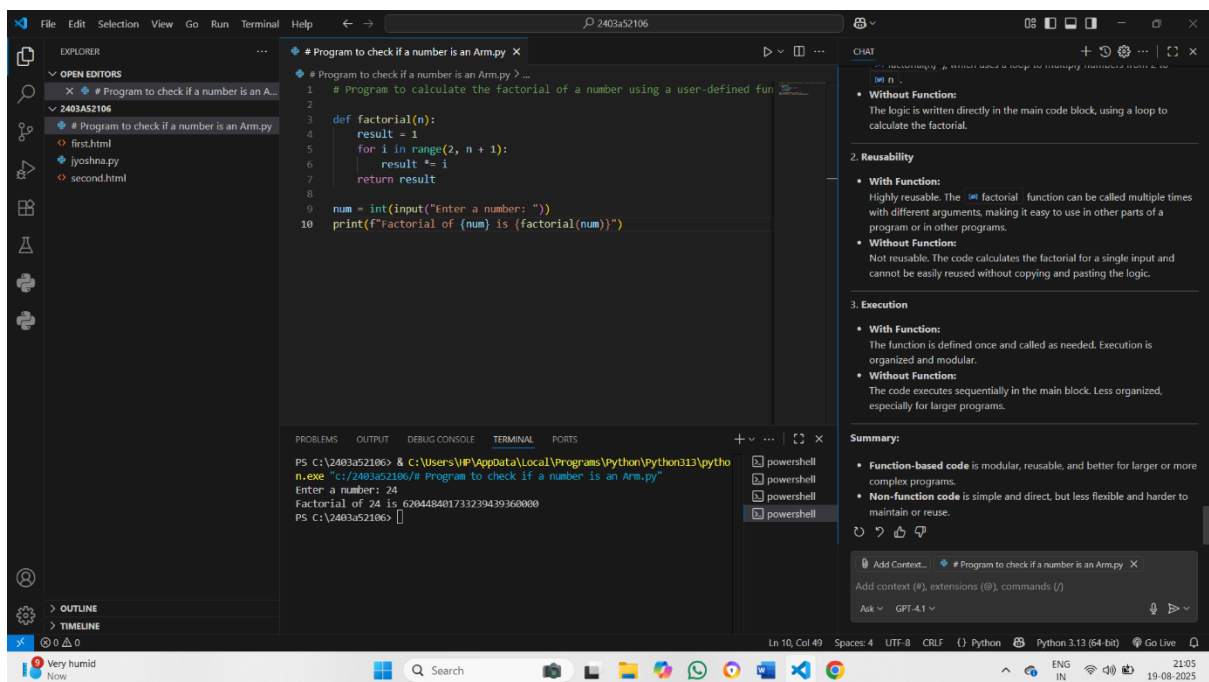
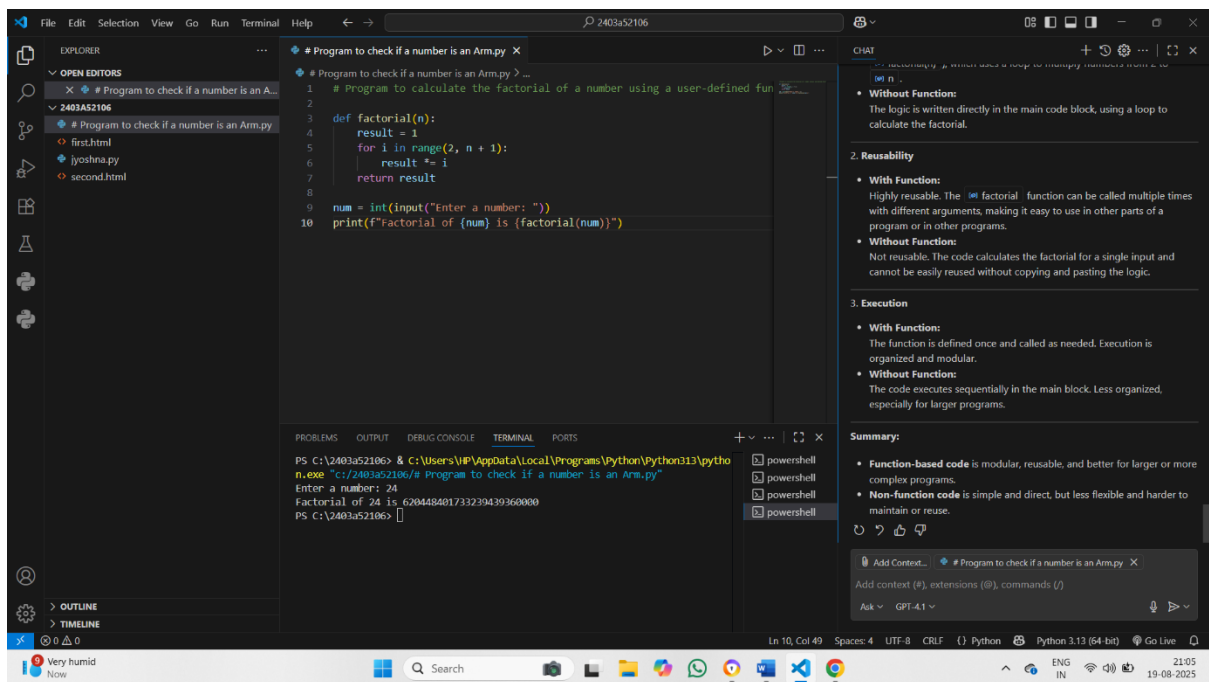
### TASK-2:



## TASK-3:



## TASK-4:



Task-5:

File Edit Selection View Go Run Terminal Help 2403a52106

EXPLORER

OPEN EDITORS

Program to check if a number is an Arm.py

first.html  
jyosha.py  
second.html

Program to check if a number is an Arm.py

```
11 # Program to check if a number is an Arm.py ...
12 # Iterative version of factorial function
13 def factorial_iterative(n):
14     result = 1
15     for i in range(2, n + 1):
16         result *= i
17     return result
18
19 # Recursive version of factorial function
20 def factorial_recursive(n):
21     if n == 0 or n == 1:
22         return 1
23     else:
24         return n * factorial_recursive(n - 1)
25
26 # Example usage
27 num = int(input("Enter a number: "))
28 print(f"Iterative: Factorial of {num} is {factorial_iterative(num)}")
29 print(f"Recursive: Factorial of {num} is {factorial_recursive(num)}")
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\2403a52106> & C:\Users\VP\AppData\Local\Programs\Python\Python313\python.exe "C:\2403a52106/# Program to check if a number is an Arm.py"  
Enter a number: 24  
Factorial of 24 is 620448401733239439360000  
PS C:\2403a52106> & C:\Users\VP\AppData\Local\Programs\Python\Python313\python.exe "C:\2403a52106/# Program to check if a number is an Arm.py"  
Enter a number: 23  
Factorial of 23 is 25852016738884976640000  
Enter a number: 3  
Iterative: Factorial of 3 is 6  
Recursive: Factorial of 3 is 6  
PS C:\2403a52106>

CHAT

1. User-Defined Factorial Function (Iterative)

```
def factorial(n):
    result = 1
    for i in range(2, n + 1):
        result *= i
    return result
```

- Calculates the factorial of `n` using a loop.
- Multiplies `result` by each number from 2 to `n`.
- Returns the final result.

2. User Input and Output

```
num = int(input("Enter a number: "))
print(f"Factorial of {num} is {factorial(num)}")
```

- Prompts the user to enter a number.
- Prints the factorial using the `factorial` function.

3. Iterative Version

```
def factorial_iterative(n):
    result = 1
    for i in range(2, n + 1):
        result *= i
    return result
```

- Another function that calculates factorial iteratively (same logic as

Ln 29, Col 70 Spaces: 4 UTF-8 CRLF Python Python 3.13 (64-bit) Go Live

File Edit Selection View Go Run Terminal Help 2403a52106

EXPLORER

OPEN EDITORS

Program to check if a number is an Arm.py

first.html  
jyosha.py  
second.html

Program to check if a number is an Arm.py

```
11 # Program to check if a number is an Arm.py ...
12 # Iterative version of factorial function
13 def factorial_iterative(n):
14     result = 1
15     for i in range(2, n + 1):
16         result *= i
17     return result
18
19 # Recursive version of factorial function
20 def factorial_recursive(n):
21     if n == 0 or n == 1:
22         return 1
23     else:
24         return n * factorial_recursive(n - 1)
25
26 # Example usage
27 num = int(input("Enter a number: "))
28 print(f"Iterative: Factorial of {num} is {factorial_iterative(num)}")
29 print(f"Recursive: Factorial of {num} is {factorial_recursive(num)}")
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\2403a52106> & C:\Users\VP\AppData\Local\Programs\Python\Python313\python.exe "C:\2403a52106/# Program to check if a number is an Arm.py"  
Enter a number: 24  
Factorial of 24 is 620448401733239439360000  
PS C:\2403a52106> & C:\Users\VP\AppData\Local\Programs\Python\Python313\python.exe "C:\2403a52106/# Program to check if a number is an Arm.py"  
Enter a number: 23  
Factorial of 23 is 25852016738884976640000  
Enter a number: 3  
Iterative: Factorial of 3 is 6  
Recursive: Factorial of 3 is 6  
PS C:\2403a52106>

CHAT

4. Recursive Version

```
def factorial_recursive(n):
    if n == 0 or n == 1:
        return 1
    else:
        return n * factorial_recursive(n - 1)
```

- Calculates factorial using recursion.
- If `n` is 0 or 1, returns 1.
- Otherwise, returns `n` multiplied by the factorial of `n-1`.

5. Example Usage

```
num = int(input("Enter a number: "))
print(f"Iterative: Factorial of {num} is {factorial_iterative(num)}")
print(f"Recursive: Factorial of {num} is {factorial_recursive(num)}")
```

- Prompts the user for another number.
- Prints the factorial using both iterative and recursive functions.

Summary:

The code demonstrates three ways to calculate the factorial of a number:

- Using a user-defined iterative function
- Using another iterative function
- Using a recursive function

It also shows how to get user input and display results for each method.

Ln 29, Col 70 Spaces: 4 UTF-8 CRLF Python Python 3.13 (64-bit) Go Live