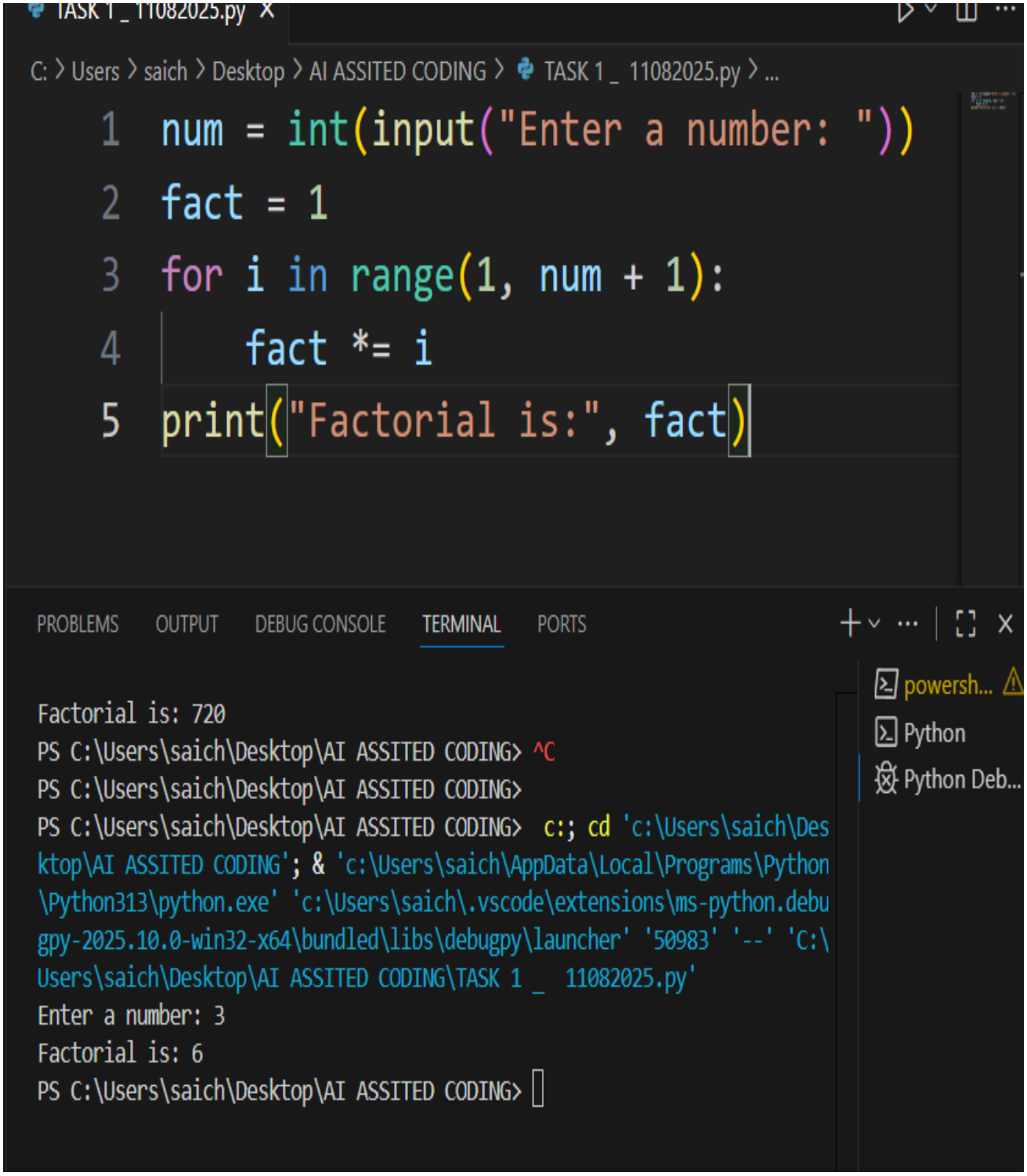
ASSIGNMENT 1

TASK 1 :



EXPALNATION :

1.

```
num = int(input("Enter a number: "))
```

Prompts the user to enter a number, converts the input to an integer, and stores it in `num`.

2.

```
fact = 1
```

Initializes the variable `fact` to 1. This will hold the factorial result.

3.

```
for i in range(1, num + 1):
```

Starts a loop from 1 to `num` (inclusive).

4.

```
    fact *= i
```

Multiplies `fact` by the current value of `i` in each iteration.

5.

```
print("Factorial is:", fact)
```

Prints the final value of `fact`, which is the factorial of the entered number.

TASK 2 :

```python
import math
num = int(input("Enter a number: "))
print("Factorial is:", math.factorial(num))
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

Enter a number: 3
Factorial is: 6
PS C:\Users\saich\Desktop\AI ASSITED CODING> ^C
PS C:\Users\saich\Desktop\AI ASSITED CODING>
PS C:\Users\saich\Desktop\AI ASSITED CODING>  c:; cd 'c:\Users\saich\Desktop\AI ASSITED CODING'; & 'c:\Users
\saich\AppData\Local\Programs\Python\Python313\python.exe' 'c:\Users\saich\.vscode\extensions\ms-python.debu
gpy-2025.10.0-win32-x64\bundled\libs\debugpy\launcher' '51251' '--' 'C:\Users\saich\Desktop\AI ASSITED CODIN
G\TASK 2 _  11082025.py'
Enter a number: 6
Factorial is: 720
PS C:\Users\saich\Desktop\AI ASSITED CODING> ^C
PS C:\Users\saich\Desktop\AI ASSITED CODING>
PS C:\Users\saich\Desktop\AI ASSITED CODING>  c:; cd 'c:\Users\saich\Desktop\AI ASSITED CODING'; & 'c:\Users
\saich\AppData\Local\Programs\Python\Python313\python.exe' 'c:\Users\saich\.vscode\extensions\ms-python.debu
gpy-2025.10.0-win32-x64\bundled\libs\debugpy\launcher' '53686' '--' 'C:\Users\saich\Desktop\AI ASSITED CODIN
G\TASK 2 _  11082025.py'
Enter a number: 5
Factorial is: 120
PS C:\Users\saich\Desktop\AI ASSITED CODING>
```

EXPALNATION :

```python
import math
```

Imports Python's built-in `math` module, which provides mathematical functions.

```python
num = int(input("Enter a number: "))
```

Prompts the user to enter a number, converts the input to an integer, and stores it in `num`.

```python
print("Factorial is:", math.factorial(num))6
```

Calculates the factorial of `num` using `math.factorial(num)` and prints the result.

TASK 3 :

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

num = int(input("Enter a number: "))
print("Factorial is:", factorial(num))
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\Users\saich> & C:/Users/saich/AppData/Local/Programs/Python/Python
313/python.exe "c:/Users/saich/Desktop/AI ASSITED CODING/TASK 2 _  11082
025.py"
Enter a number: 6
Factorial is: 720
PS C:\Users\saich>
```

powersh...
Python
Python Deb...
powershell
Python

EXPALNATION :

```python
def factorial(n):
```

Defines a function named `factorial` that takes one argument `n`.

```python
    result = 1
```

Initializes a variable `result` to 1. This will store the factorial value.

```python
    for i in range(2, n + 1):
```

Starts a loop from 2 up to and including `n`.

```python
        result *= i
```

Multiplies `result` by the current value of `i` in each iteration.

```python
    return result
```

Returns the final value of `result` (the factorial of `n`).

```python
num = int(input("Enter a number: "))
```

Prompts the user to enter a number, converts it to an integer, and stores it in `num`.

```python
print("Factorial is:", factorial(num))
```

TASK 4 :

```python
num = int(input("Enter a number: "))
fact = 1
for i in range(1, num + 1):
    fact *= i
print("Factorial is:", fact)

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

num = int(input("Enter a number: "))
print("Factorial is:", factorial(num))3
```

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\saich\Desktop\AI ASSITED CODING> ^C
PS C:\Users\saich\Desktop\AI ASSITED CODING>
PS C:\Users\saich\Desktop\AI ASSITED CODING>  c:; cd 'c:\Users\saich\Desktop\AI ASSITED CODING'; & 'c:\Users
\saich\AppData\Local\Programs\Python\Python313\python.exe' 'c:\Users\saich\.vscode\extensions\ms-python.debu
gpy-2025.10.0-win32-x64\bundled\libs\debugpy\launcher' '53686' '--' 'C:\Users\saich\Desktop\AI ASSITED CODIN
G\TASK 2 _  11082025.py'
Enter a number: 5
Factorial is: 120
PS C:\Users\saich\Desktop\AI ASSITED CODING> X^C
PS C:\Users\saich\Desktop\AI ASSITED CODING>
PS C:\Users\saich\Desktop\AI ASSITED CODING>  c:; cd 'c:\Users\saich\Desktop\AI ASSITED CODING'; & 'c:\Users
\saich\AppData\Local\Programs\Python\Python313\python.exe' 'c:\Users\saich\.vscode\extensions\ms-python.debu
gpy-2025.10.0-win32-x64\bundled\libs\debugpy\launcher' '53851' '--' 'C:\Users\saich\Desktop\AI ASSITED CODIN
G\TASK 4 _  11082025.py'
Enter a number: 3
Factorial is: 6
Enter a number: 3
Factorial is: 6
```

EXPALNATION :

```
num = int(input("Enter a number: "))
```

Prompts the user to enter a number, converts it to an integer, and stores it in `num`.

```
fact = 1
```

Initializes the variable `fact` to 1. This will hold the factorial result.

```
for i in range(1, num + 1):
```

Starts a loop from 1 up to and including `num`.

```
    fact *= i
```

Multiplies `fact` by the current value of `i` in each iteration.

```
print("Factorial is:", fact)
```

Prints the final value of `fact`, which is the factorial of the entered number.

Factorial With Functions

```
def factorial(n):
```

Defines a function named `factorial` that takes one argument `n`.

```
    result = 1
```

Initializes the variable `result` to 1 inside the function.

```
    for i in range(1, n + 1):
```

Starts a loop from 1 up to and including `n`.

```
        result *= i
```

Multiplies `result` by the current value of `i` in each iteration.

```
    return result
```

Returns the final value of `result` (the factorial of `n`).

```
num = int(input("Enter a number: "))
```

Prompts the user to enter a number, converts it to an integer, and stores it in `num`.

```
print("Factorial is:", factorial(num))
```

TASK 5 :

```python
def factorial_iterative(n):
    """Calculate factorial using an iterative approach."""
    result = 1
    for i in range(2, n + 1):
        result *= i
    return result

# Recursive factorial implementation
def factorial_recursive(n):
    """Calculate factorial using a recursive approach."""
    if n == 0 or n == 1:
        return 1
    else:
        return n * factorial_recursive(n - 1)
```

```
PROBLEMS 1  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

ta\Local\Programs\Python\Python313\python.exe' 'c:\Users\saich\.vscode\extensions\ms-python.debugpy-2025.10.0-
bundled\libs\debugpy\launcher' '53851' '--' 'C:\Users\saich\Desktop\AI ASSITED CODING\TASK 4 _  11082025.py'
Enter a number: 3
Factorial is: 6
Enter a number: 3
Factorial is: 6
PS C:\Users\saich\Desktop\AI ASSITED CODING> ^C
PS C:\Users\saich\Desktop\AI ASSITED CODING>
PS C:\Users\saich\Desktop\AI ASSITED CODING>  c:; cd 'c:\Users\saich\Desktop\AI ASSITED CODING'; & 'c:\Users\s
ta\Local\Programs\Python\Python313\python.exe' 'c:\Users\saich\.vscode\extensions\ms-python.debugpy-2025.10.0-
bundled\libs\debugpy\launcher' '53967' '--' 'C:\Users\saich\Desktop\AI ASSITED CODING\TASK 5 _  11082025.py'
PS C:\Users\saich\Desktop\AI ASSITED CODING> ^C
PS C:\Users\saich\Desktop\AI ASSITED CODING>
PS C:\Users\saich\Desktop\AI ASSITED CODING>  c:; cd 'c:\Users\saich\Desktop\AI ASSITED CODING'; & 'c:\Users\s
ta\Local\Programs\Python\Python313\python.exe' 'c:\Users\saich\.vscode\extensions\ms-python.debugpy-2025.10.0-
bundled\libs\debugpy\launcher' '53998' '--' 'C:\Users\saich\Desktop\AI ASSITED CODING\TASK 5 _  11082025.py'
PS C:\Users\saich\Desktop\AI ASSITED CODING> 5
5
PS C:\Users\saich\Desktop\AI ASSITED CODING>
```

EXPALNATION :

- `def factorial_iterative(n):`
  Defines a function named `factorial_iterative` that takes an integer `n`.
- `result = 1`
  Initializes a variable `result` to 1. This will hold the running product.
- `for i in range(2, n + 1):`
  Starts a loop with `i` going from 2 up to and including `n`.
- `result *= i`
  Multiplies `result` by `i` in each iteration, accumulating the product.
- `return result`
  Returns the final value of `result`, which is the factorial of `n`.

- `def factorial_recursive(n):`
  Defines a function named `factorial_recursive` that takes an integer `n`.
- `if n == 0 or n == 1:`
  Checks if `n` is 0 or 1, which are the base cases for factorial.
- `return 1`
  Returns 1 if the base case is met.
- `else:`
  If `n` is greater than 1, proceeds to the recursive case.
- `return n * factorial_recursive(n - 1)`
  Returns `n` multiplied by the factorial of `n-1`, calling the same function recursively until the base case is reached.