

ASSIGNMENT-1

NAME:-P LAHARI

BATCH-16

HTNO:-2303A51071

Task 1: AI-Generated Logic Without Modularisation (*Factorial without Functions*)

The screenshot shows the Thonny Python IDE interface. The code editor window displays a script named 'untitled' with the following content:

```
1 num = int(input("Enter a number to calculate its factorial: "))
2 if num < 0:
3     print("Factorial is not defined for negative numbers")
4 else:
5     factorial = 1
6     for i in range(1, num + 1):
7         factorial *= i
8     print(f"The factorial of {num} is {factorial}")
9
```

The shell window below shows the execution of the code. The user runs the script with the command `>>> %Run -c $EDITOR_CONTENT`. The program prompts for a number, receives '5', calculates the factorial (120), and prints the result.

```
Shell <untitled> x
>>> %Run -c $EDITOR_CONTENT
Enter a number to calculate its factorial: 5
The factorial of 5 is 120
>>> |
```

Task 2: AI Code Optimization & Cleanup (Improving Efficiency and Readability)

The screenshot shows the Thonny Python IDE interface. The top window is titled "Thonny - <untitled> @ 7:24" and contains a code editor with the following Python script:

```
1 num = int(input("Enter a number to calculate its factorial: "))
2 if num < 0:
3     print("Factorial is not defined for negative numbers")
4 else:
5     result = 1
6     for value in range(1, num + 1):
7         result *= value
8     print(f"The factorial of {num} is {result}")
9
```

The bottom window is titled "Shell" and shows the output of running the script:

```
>>> %Run -c $EDITOR_CONTENT
Enter a number to calculate its factorial: 6
The factorial of 6 is 720
>>> |
```

Task 3: Modular Design Using AI Assistance (Factorial with Functions)

The screenshot shows the Thonny Python IDE interface. The top window is titled "Thonny - <untitled> @ 9:34" and contains a code editor with the following Python script:

```
1 def calculate_factorial(n):
2     if n < 0:
3         return "Factorial is not defined for negative numbers"
4     result = 1
5     for i in range(1, n + 1):
6         result *= i
7     return result
8 num = int(input("Enter a number to calculate its factorial: "))
9 output = calculate_factorial(num)
10 if isinstance(output, str):
11     print(output)
12 else:
13     print(f"The factorial of {num} is {output}")
14
```

The bottom window is titled "Shell" and shows the execution of the code:

```
>>> %Run -c $EDITOR_CONTENT
Enter a number to calculate its factorial: 5
The factorial of 5 is 120
>>>
```

Task 4: Comparative Analysis – Procedural vs Modular AI Code

Procedural (Without Function):-

The screenshot shows the Thonny Python IDE interface. The top window is the code editor with the title "Thonny - <untitled> @ 11:1". It contains the following Python code:

```
1 num = int(input("Enter a number to calculate its factorial: "))
2
3 if num < 0:
4     print("Factorial is not defined for negative numbers")
5 else:
6     result = 1
7     for i in range(1, num + 1):
8         result *= i
9
10    print(f"The factorial of {num} is {result}")
11
```

The bottom window is the "Shell" tab, which displays the output of running the script. The command `>>> %Run -c \$EDITOR_CONTENT` is run, followed by the user input "Enter a number to calculate its factorial: 4" and the program's response "The factorial of 4 is 24".

Modular (With Function)

The screenshot shows the Thonny Python IDE interface. The top part is the script editor with the following code:

```
1 def factorial(n):
2     if n < 0:
3         return "Factorial is not defined for negative numbers"
4     result = 1
5     for i in range(1, n + 1):
6         result *= i
7     return result
8 num = int(input("Enter a number to calculate its factorial: "))
9 res = factorial(num)
10 if isinstance(res, str):
11     print(res)
12 else:
13     print(f"The factorial of {num} is {res}")
14
```

The bottom part is the shell terminal with the following interaction:

```
Shell >>> %Run -c $EDITOR_CONTENT
Enter a number to calculate its factorial: 5
The factorial of 5 is 120
>>>
```

Task 5: AI-Generated Iterative vs Recursive Thinking

Iterative Approach

The screenshot shows the Thonny Python IDE interface. The top window is the code editor with the title 'Thonny - <untitled> @ 8:23'. It contains the following Python code:

```
1 # Iterative factorial program
2 num = int(input("Enter a number to calculate its factorial: "))
3 if num < 0:
4     print("Factorial is not defined for negative numbers")
5 else:
6     factorial = 1
7     for i in range(1, num + 1):
8         factorial *= i
9     print(f"The factorial of {num} is {factorial}")
10
```

The bottom window is the shell, titled 'Shell'. It shows the command '%Run -c \$EDITOR_CONTENT' followed by the program's output:

```
>>> %Run -c $EDITOR_CONTENT
Enter a number to calculate its factorial: 5
The factorial of 5 is 120
>>> |
```

Recursive Approach

The screenshot shows the Thonny Python IDE interface. The top menu bar includes File, Edit, View, Run, Tools, and Help. Below the menu is a toolbar with icons for file operations like Open, Save, and Run. The main window has two tabs: <untitled> and Shell.

The code editor tab (<untitled>) contains the following Python script:

```
1 # Recursive factorial program
2 def factorial(n):
3     if n <= 1:
4         return 1
5     return n * factorial(n - 1)
6 num = int(input("Enter a number to calculate its factorial: "))
7 if num < 0:
8     print("Factorial is not defined for negative numbers")
9 else:
10    result = factorial(num)
11    print(f"The factorial of {num} is {result}")
12
```

The shell tab shows the output of running the script:

```
>>> %Run -c $EDITOR_CONTENT
Enter a number to calculate its factorial: 5
The factorial of 5 is 120
>>> |
```