ASSIGNMENT-1.5

HT.NO : 2303a52438

Name : T.Rishitha

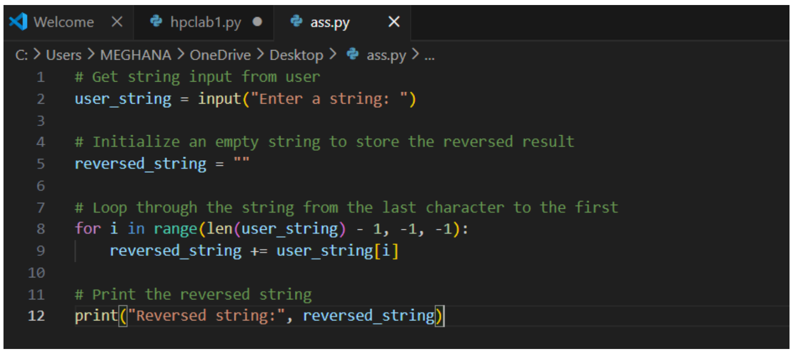
Batch-45

TASK-1

PROMPT:

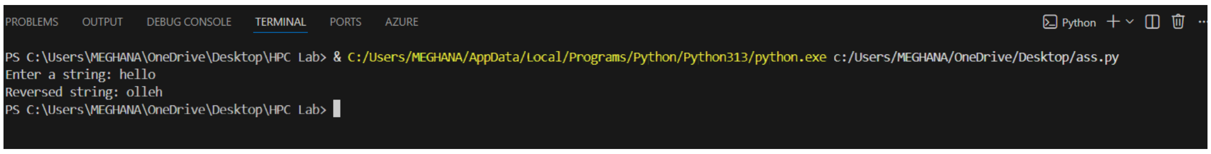
Write a simple and easy-to-understand Python program that takes a string as user input, reverses it step by step using basic logic, prints the reversed string, and does not use any user-defined functions.

CODE:



```python
# Get string input from user
user_string = input("Enter a string: ")

# Initialize an empty string to store the reversed result
reversed_string = ""

# Loop through the string from the last character to the first
for i in range(len(user_string) - 1, -1, -1):
    reversed_string += user_string[i]

# Print the reversed string
print("Reversed string:", reversed_string)
```

OUTPUT:



```
PS C:\Users\MEGHANA\OneDrive\Desktop\HPC Lab> & C:/Users/MEGHANA/AppData/Local/Programs/Python/Python313/python.exe c:/Users/MEGHANA/OneDrive/Desktop/ass.py
Enter a string: hello
Reversed string: olleh
PS C:\Users\MEGHANA\OneDrive\Desktop\HPC Lab>
```

JUSTIFICATION:

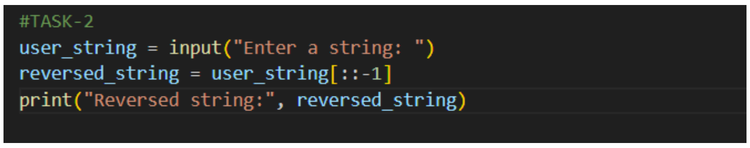
This prompt is good because it helps beginners practice the basics of Python. It focuses on simple input, loops, and string handling without shortcuts. The task is clear, easy to measure, and builds real understanding step by step.

TASK-2

PROMPT:

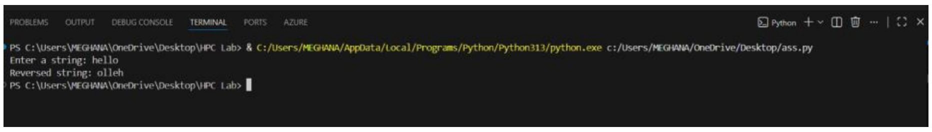
Give Python code that reverses a string simpler, easier to read, and faster, without using functions.

CODE:



```python
#TASK-2
user_string = input("Enter a string: ")
reversed_string = user_string[::-1]
print("Reversed string:", reversed_string)
```

OUTPUT:



```
PS C:\Users\MEGHANA\OneDrive\Desktop\HPC Lab> & C:/Users/MEGHANA/AppData/Local/Programs/Python/Python313/python.exe c:/Users/MEGHANA/OneDrive/Desktop/ass.py
Enter a string: hello
Reversed string: olleh
PS C:\Users\MEGHANA\OneDrive\Desktop\HPC Lab>
```

JUSTIFICATION:

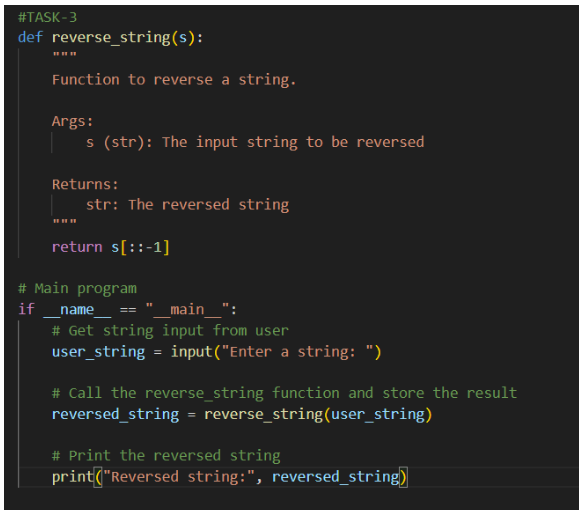
This prompt is justified because it helps learners practice a fundamental Python skill in a way that is **clear, efficient, and beginner�friendly**, while focusing on readability and performance.

TASK-3

PROMPT:

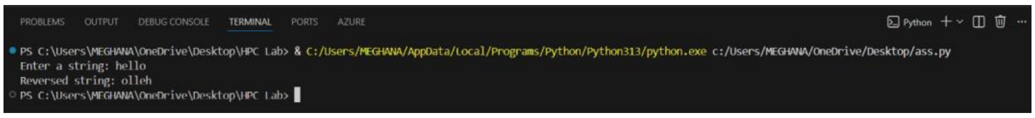
Write a Python program that defines a function to reverse a string, returns the reversed string, uses the function in the main program, and includes meaningful comments explaining each step.

CODE:



```python
#TASK-3
def reverse_string(s):
    """
    Function to reverse a string.

    Args:
        s (str): The input string to be reversed

    Returns:
        str: The reversed string
    """
    return s[::-1]

# Main program
if __name__ == "__main__":
    # Get string input from user
    user_string = input("Enter a string: ")

    # Call the reverse_string function and store the result
    reversed_string = reverse_string(user_string)

    # Print the reversed string
    print("Reversed string:", reversed_string)
```

OUTPUT:



```
PS C:\Users\MEGHANA\OneDrive\Desktop\HPC Lab> & C:/Users/MEGHANA/AppData/Local/Programs/Python/Python313/python.exe c:/Users/MEGHANA/OneDrive/Desktop/ass.py
Enter a string: hello
Reversed string: olleh
PS C:\Users\MEGHANA\OneDrive\Desktop\HPC Lab>
```

JUSTIFICATION:

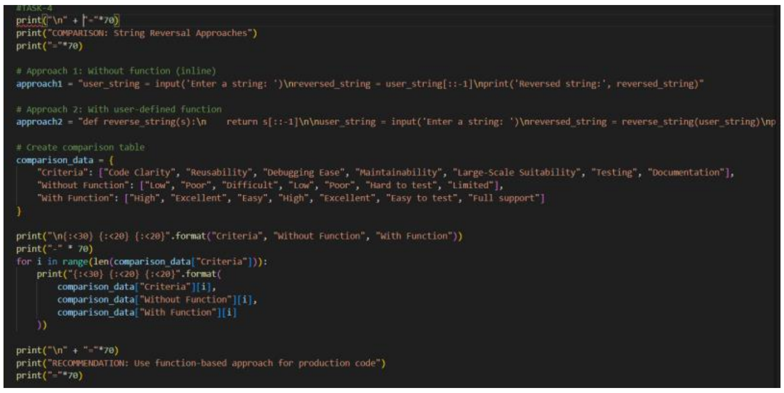
his prompt is well-designed for teaching clean, modular Python programming while reinforcing core concepts through a simple, focused task.

TASK-4

PROMPT:

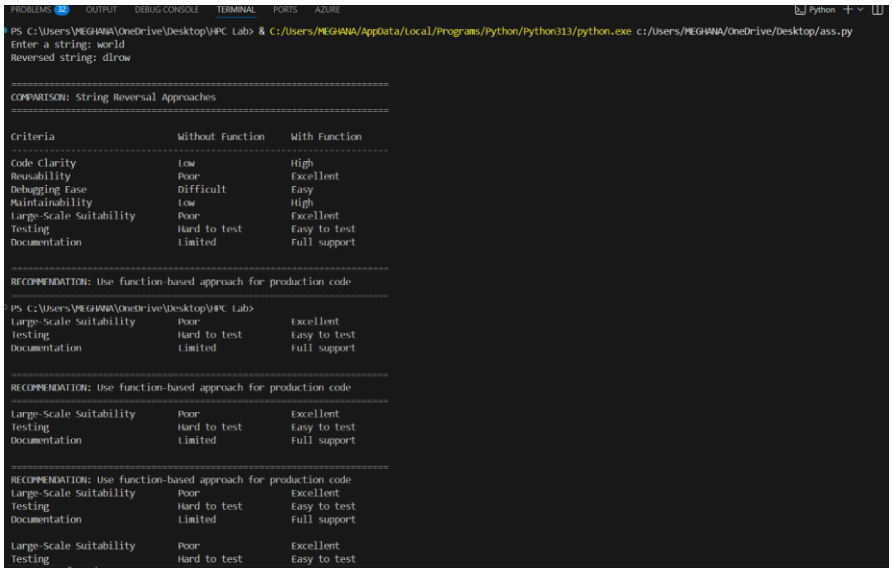
Write a Python program that compares two string reversal approaches: one without functions and one using a user-defined function. Analyze them based on code clarity, reusability, debugging ease, and suitability for large-scale applications, and display the comparison in a clear table or summary.

CODE:

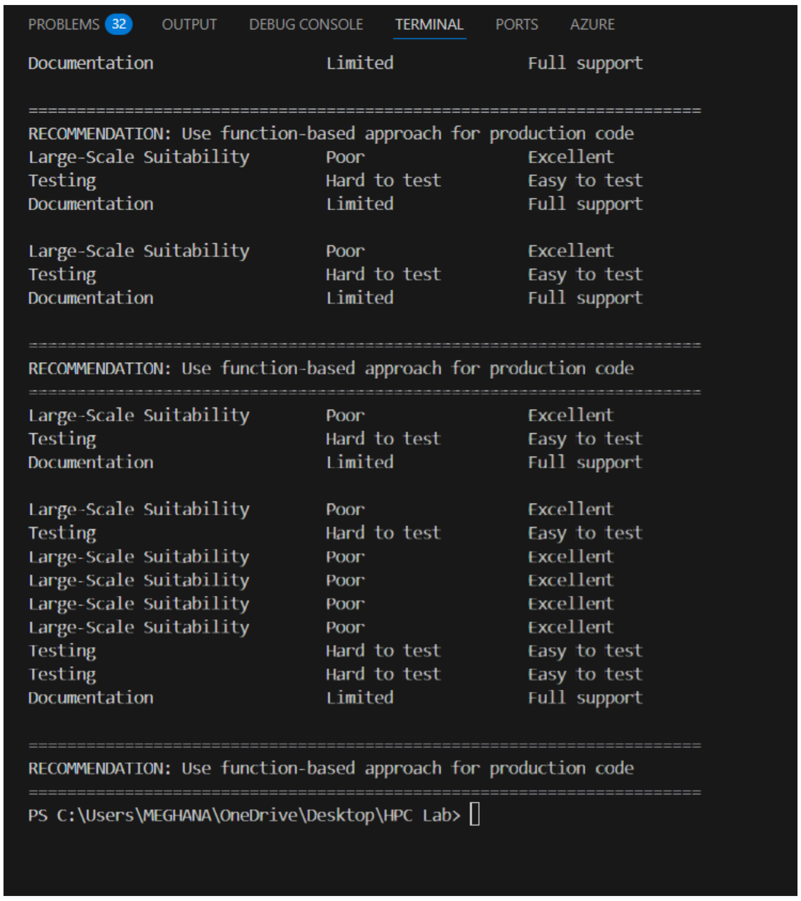


```python
#TASK-4
print("\n" + "="*70)
print("COMPARISON: String Reversal Approaches")
print("="*70)

# Approach 1: Without function (inline)
approach1 = "user_string = input('Enter a string: ')\nreversed_string = user_string[::-1]\nprint('Reversed string:', reversed_string)"

# Approach 2: With user-defined function
approach2 = "def reverse_string(s):\n    return s[::-1]\n\nuser_string = input('Enter a string: ')\nreversed_string = reverse_string(user_string)\np

# Create comparison table
comparison_data = {
    "Criteria": ["Code Clarity", "Reusability", "Debugging Ease", "Maintainability", "Large-Scale Suitability", "Testing", "Documentation"],
    "Without Function": ["Low", "Poor", "Difficult", "Low", "Poor", "Hard to test", "Limited"],
    "With Function": ["High", "Excellent", "Easy", "High", "Excellent", "Easy to test", "Full support"]
}

print("\n{:<30} {:<20} {:<20}".format("Criteria", "Without Function", "With Function"))
print("-" * 70)
for i in range(len(comparison_data["Criteria"])):
    print("{:<30} {:<20} {:<20}".format(
        comparison_data["Criteria"][i],
        comparison_data["Without Function"][i],
        comparison_data["With Function"][i]
    ))

print("\n" + "="*70)
print("RECOMMENDATION: Use function-based approach for production code")
print("="*70)
```

OUTPUT:



```
PS C:\Users\MEGHANA\OneDrive\Desktop\HPC Lab> & C:/Users/MEGHANA/AppData/Local/Programs/Python/Python313/python.exe c:/Users/MEGHANA/OneDrive/Desktop/ass.py
Enter a string: world
Reversed string: dlrow

======================================================================
COMPARISON: String Reversal Approaches
======================================================================

Criteria                       Without Function     With Function
----------------------------------------------------------------------
Code Clarity                   Low                  High
Reusability                    Poor                 Excellent
Debugging Ease                 Difficult            Easy
Maintainability                Low                  High
Large-Scale Suitability        Poor                 Excellent
Testing                        Hard to test         Easy to test
Documentation                  Limited              Full support

======================================================================
RECOMMENDATION: Use function-based approach for production code
======================================================================
PS C:\Users\MEGHANA\OneDrive\Desktop\HPC Lab>
Large-Scale Suitability        Poor                 Excellent
Testing                        Hard to test         Easy to test
Documentation                  Limited              Full support

======================================================================
RECOMMENDATION: Use function-based approach for production code
======================================================================
Large-Scale Suitability        Poor                 Excellent
Testing                        Hard to test         Easy to test
Documentation                  Limited              Full support

======================================================================
RECOMMENDATION: Use function-based approach for production code
Large-Scale Suitability        Poor                 Excellent
Testing                        Hard to test         Easy to test
Documentation                  Limited              Full support

Large-Scale Suitability        Poor                 Excellent
Testing                        Hard to test         Easy to test
```



```
Documentation                 Limited            Full support

======================================================================
RECOMMENDATION: Use function-based approach for production code
Large-Scale Suitability       Poor               Excellent
Testing                       Hard to test       Easy to test
Documentation                 Limited            Full support

Large-Scale Suitability       Poor               Excellent
Testing                       Hard to test       Easy to test
Documentation                 Limited            Full support

======================================================================
RECOMMENDATION: Use function-based approach for production code
======================================================================
Large-Scale Suitability       Poor               Excellent
Testing                       Hard to test       Easy to test
Documentation                 Limited            Full support

Large-Scale Suitability       Poor               Excellent
Testing                       Hard to test       Easy to test
Large-Scale Suitability       Poor               Excellent
Large-Scale Suitability       Poor               Excellent
Large-Scale Suitability       Poor               Excellent
Large-Scale Suitability       Poor               Excellent
Testing                       Hard to test       Easy to test
Testing                       Hard to test       Easy to test
Documentation                 Limited            Full support

======================================================================
RECOMMENDATION: Use function-based approach for production code
======================================================================
PS C:\Users\MEGHANA\OneDrive\Desktop\HPC Lab> 
```

JUSTIFICATION:

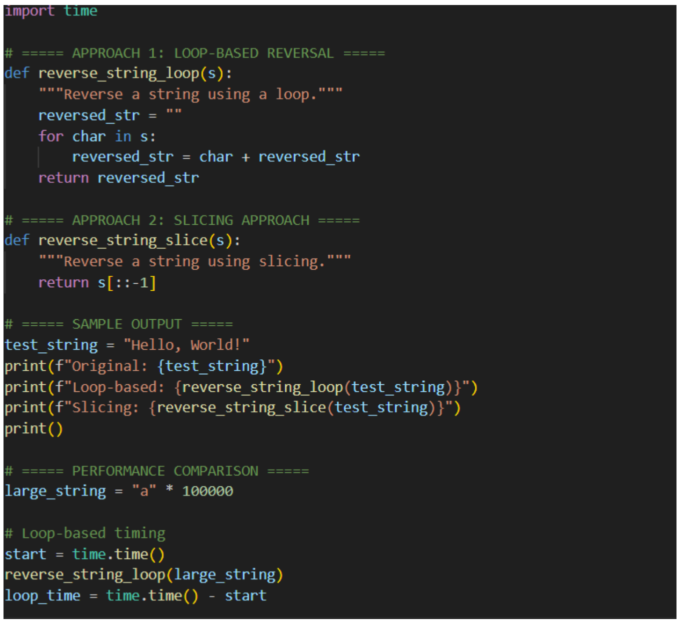
This prompt is justified because it teaches **both coding and evaluation skills**, helping learners understand not only *how* to solve a problem but also *why* one approach may be better than another in larger applications.

TASK-5

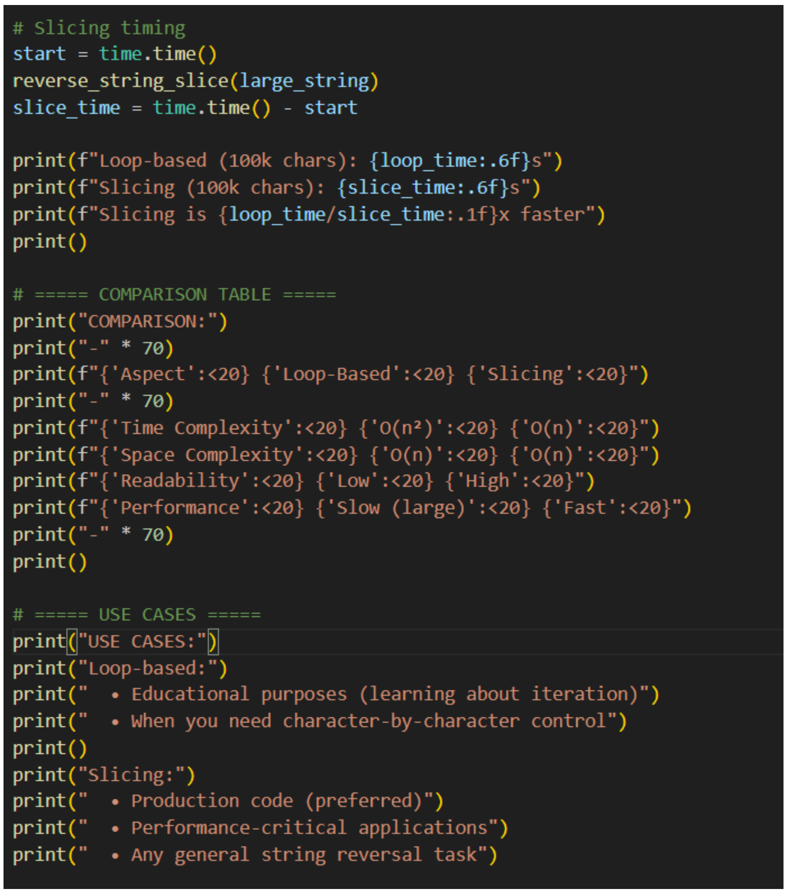
PROMPT:

Write Python code to reverse a string using both a loop-based approach and a slicing approach, with comments, sample output, and a comparison of execution flow, time complexity, performance for large inputs, and appropriate use cases

CODE:

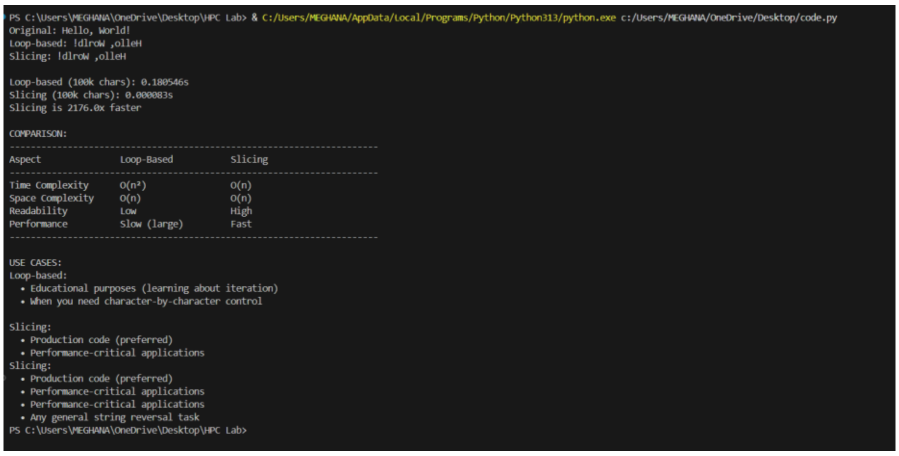


```python
import time

# ===== APPROACH 1: LOOP-BASED REVERSAL =====
def reverse_string_loop(s):
    """Reverse a string using a loop."""
    reversed_str = ""
    for char in s:
        reversed_str = char + reversed_str
    return reversed_str

# ===== APPROACH 2: SLICING APPROACH =====
def reverse_string_slice(s):
    """Reverse a string using slicing."""
    return s[::-1]

# ===== SAMPLE OUTPUT =====
test_string = "Hello, World!"
print(f"Original: {test_string}")
print(f"Loop-based: {reverse_string_loop(test_string)}")
print(f"Slicing: {reverse_string_slice(test_string)}")
print()

# ===== PERFORMANCE COMPARISON =====
large_string = "a" * 100000

# Loop-based timing
start = time.time()
reverse_string_loop(large_string)
loop_time = time.time() - start
```



```python
# Slicing timing
start = time.time()
reverse_string_slice(large_string)
slice_time = time.time() - start

print(f"Loop-based (100k chars): {loop_time:.6f}s")
print(f"Slicing (100k chars): {slice_time:.6f}s")
print(f"Slicing is {loop_time/slice_time:.1f}x faster")
print()

# ===== COMPARISON TABLE =====
print("COMPARISON:")
print("-" * 70)
print(f"{'Aspect':<20} {'Loop-Based':<20} {'Slicing':<20}")
print("-" * 70)
print(f"{'Time Complexity':<20} {'O(n²)':<20} {'O(n)':<20}")
print(f"{'Space Complexity':<20} {'O(n)':<20} {'O(n)':<20}")
print(f"{'Readability':<20} {'Low':<20} {'High':<20}")
print(f"{'Performance':<20} {'Slow (large)':<20} {'Fast':<20}")
print("-" * 70)
print()

# ===== USE CASES =====
print("USE CASES:")
print("Loop-based:")
print("  • Educational purposes (learning about iteration)")
print("  • When you need character-by-character control")
print()
print("Slicing:")
print("  • Production code (preferred)")
print("  • Performance-critical applications")
print("  • Any general string reversal task")
```

OUTPUT:



```
PS C:\Users\MEGHANA\OneDrive\Desktop\HPC Lab> & C:/Users/MEGHANA/AppData/Local/Programs/Python/Python313/python.exe c:/Users/MEGHANA/OneDrive/Desktop/code.py
Original: Hello, World!
Loop-based: !dlroW ,olleH
Slicing: !dlroW ,olleH

Loop-based (100k chars): 0.180546s
Slicing (100k chars): 0.000083s
Slicing is 2176.0x faster

COMPARISON:
----------------------------------------------------------------------
Aspect              Loop-Based          Slicing
----------------------------------------------------------------------
Time Complexity     O(n²)               O(n)
Space Complexity    O(n)                O(n)
Readability         Low                 High
Performance         Slow (large)        Fast
----------------------------------------------------------------------

USE CASES:
Loop-based:
  • Educational purposes (learning about iteration)
  • When you need character-by-character control

Slicing:
  • Production code (preferred)
  • Performance-critical applications
Slicing:
  • Production code (preferred)
  • Performance-critical applications
  • Performance-critical applications
  • Any general string reversal task
PS C:\Users\MEGHANA\OneDrive\Desktop\HPC Lab>
```

JUSTIFICATION:

This prompt is justified because it teaches coding and evaluation skills, helping learners write solutions while understanding efficiency, readability, and practical use. It builds a solid foundation for clean, efficient, and well-documented Python code.