AI Assistant Coding

Assignemt1.5

Name: S Srinidhi

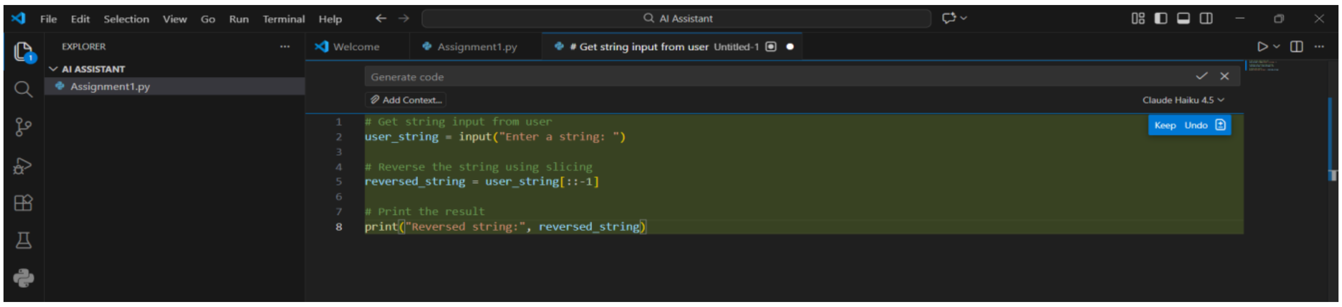
HT NO:2303A51342

BATCH:20

Lab 1: Environment Setup – GitHub Copilot and VS Code Integration + Understanding AI-assisted Coding Workflow

Task 1: AI-Generated Logic Without Modularization (String Reversal Without Functions)

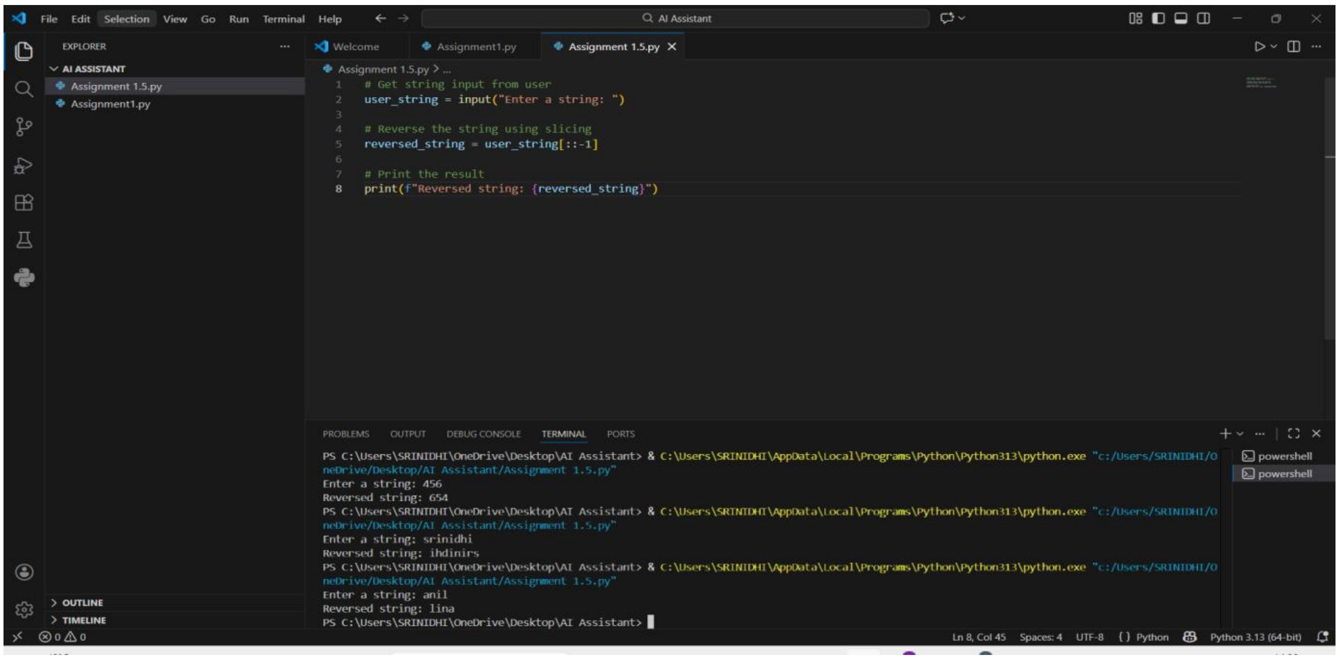
Prompt:
“Generate a Python program to reverse a user-input string directly in main code without defining any functions.Include simple input and output statements and show the reversed string.”



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

# Reverse the string using slicing
reversed_string = user_string[::-1]

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



File Edit Selection View Go Run Terminal Help
AI Assistant
EXPLORER
AI ASSISTANT
Assignment 1.5.py
Assignment1.py
Welcome
Assignment1.py
Assignment 1.5.py
OUTLINE
TIMELINE
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
powershell
powershell
Ln 8, Col 45 Spaces: 4 UTF-8 {} Python Python 3.13 (64-bit)

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

# Reverse the string using slicing
reversed_string = user_string[::-1]

# Print the result
print(f"Reversed string: {reversed_string}")
```

```
PS C:\Users\SRINIDHI\OneDrive\Desktop\AI Assistant> & C:\Users\SRINIDHI\AppData\Local\Programs\Python\Python313\python.exe "c:/Users/SRINIDHI/O
neDrive/Desktop/AI Assistant/Assignment 1.5.py"
Enter a string: 456
Reversed string: 654
PS C:\Users\SRINIDHI\OneDrive\Desktop\AI Assistant> & C:\Users\SRINIDHI\AppData\Local\Programs\Python\Python313\python.exe "c:/Users/SRINIDHI/O
neDrive/Desktop/AI Assistant/Assignment 1.5.py"
Enter a string: srinidhi
Reversed string: ihdinirs
PS C:\Users\SRINIDHI\OneDrive\Desktop\AI Assistant> & C:\Users\SRINIDHI\AppData\Local\Programs\Python\Python313\python.exe "c:/Users/SRINIDHI/O
neDrive/Desktop/AI Assistant/Assignment 1.5.py"
Enter a string: anil
Reversed string: lina
PS C:\Users\SRINIDHI\OneDrive\Desktop\AI Assistant>
```

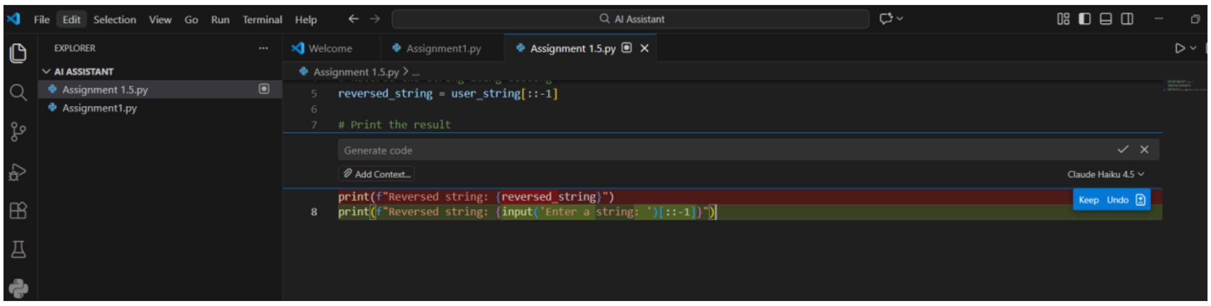
Explanation

This program reverses a string entered by the user without using any user-defined functions. All the logic is written directly in the main program.First, the program asks the user to enter a string using the input() statement. This string is stored in a variable.Next, an empty string is created to store the reversed result. The program uses a for loop to read each character of the input string one by one. Each character is added to the beginning of the new string. By placing every new character in front, the original order of characters is reversed.After the loop finishes executing, the reversed string is fully formed. Finally, the program prints the reversed string as output.This approach clearly demonstrates basic string manipulation and looping concepts while following the requirement of not using functions.

Task 2: Efficiency & Logic Optimization (Readability Improvement)

Prompt:
"Optimize the string reversal code for readability and efficiency by simplifying loops and removing unnecessary variables.Provide an improved version of the code and explain why it is better than the original."

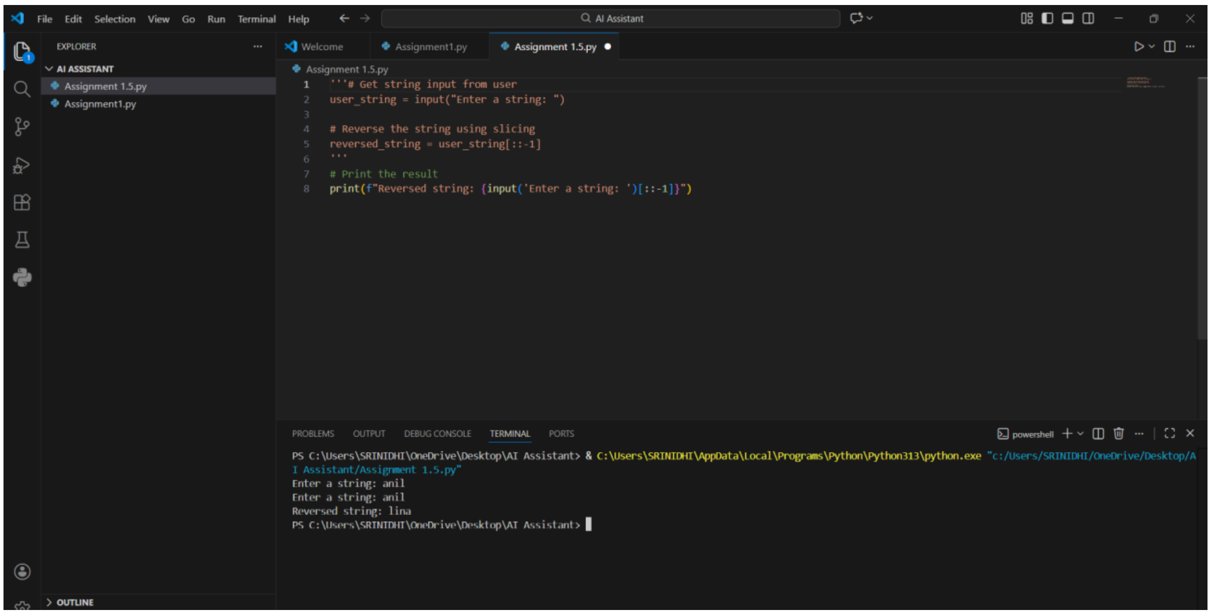
Original Code (From Task 1)



```
reversed_string = user_string[::-1]

# Print the result
print(f"Reversed string: {reversed_string}")
print(f"Reversed string: {input('Enter a string: ')[::-1]}")
```

Optimized Code (Improved Readability & Efficiency)



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

# Reverse the string using slicing
reversed_string = user_string[::-1]
'''
# Print the result
print(f"Reversed string: {input('Enter a string: ')[::-1]}")
```

```
PS C:\Users\SRINIDHI\OneDrive\Desktop\AI Assistant> & C:\Users\SRINIDHI\AppData\Local\Programs\Python\Python313\python.exe "c:/Users/SRINIDHI/OneDrive/Desktop/AI Assistant/Assignment 1.5.py"
Enter a string: anil
Enter a string: anil
Reversed string: lina
PS C:\Users\SRINIDHI\OneDrive\Desktop\AI Assistant>
```

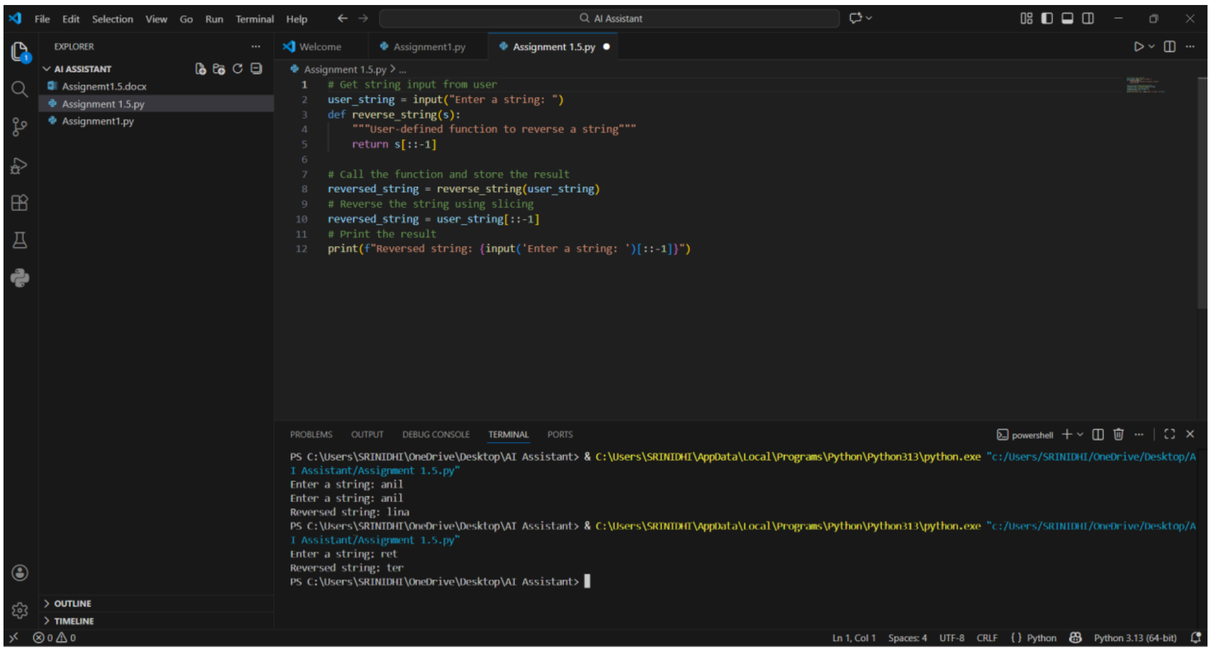
Explanation of Improvements

1. **Removed Unnecessary Variables**
   - The loop variable and manual string construction were removed.
   - The slicing method directly produces the reversed string.

2. **Simplified Logic**
   - Replaced multiple lines of looping logic with a single slicing operation.
   - This makes the code easier to understand at a glance.
3. **Improved Readability**
   - Fewer lines of code.
   - Clear and self-explanatory syntax.
   - Easier for other developers to review and maintain.
4. **Time Complexity Improvement**
   - **Original Code:**
     Each loop iteration creates a new string, leading to **O(n²)** time complexity due to repeated string concatenation.
   - **Optimized Code:**
     String slicing runs in **O(n)** time, making it more efficient.

Task 3: Modular Design Using AI Assistance (String Reversal Using Functions)

Prompt:
"Write a Python function that takes a string input and returns the reversed string, including meaningful comments.Use GitHub Copilot to generate function-based code and include sample test cases."



```python
# Get string input from user
user_string = input("Enter a string: ")
def reverse_string(s):
    """User-defined function to reverse a string"""
    return s[::-1]

# Call the function and store the result
reversed_string = reverse_string(user_string)
# Reverse the string using slicing
reversed_string = user_string[::-1]
# Print the result
print(f"Reversed string: {input('Enter a string: ')[::-1]}")
```

```
PS C:\Users\SRINIDHI\OneDrive\Desktop\AI Assistant> & C:\Users\SRINIDHI\AppData\Local\Programs\Python\Python313\python.exe "c:/Users/SRINIDHI/OneDrive/Desktop/A
I Assistant/Assignment 1.5.py"
Enter a string: anil
Enter a string: anil
Reversed string: lina
PS C:\Users\SRINIDHI\OneDrive\Desktop\AI Assistant> & C:\Users\SRINIDHI\AppData\Local\Programs\Python\Python313\python.exe "c:/Users/SRINIDHI/OneDrive/Desktop/A
I Assistant/Assignment 1.5.py"
Enter a string: ret
Reversed string: ter
PS C:\Users\SRINIDHI\OneDrive\Desktop\AI Assistant>
```

Explanation

1. **Function Creation (reverse_string)**
   - Encapsulates the string reversal logic.

- Makes the code reusable wherever string reversal is needed.
- Accepts input string and returns the reversed string.

2. **Logic Inside Function**
    - Uses a loop to prepend each character to an initially empty string.
    - This manually constructs the reversed string.
3. **Main Program**
    - Reads input from the user.
    - Calls the reverse_string function.
    - Prints the returned result.
4. **AI Assistance**
    - GitHub Copilot can suggest the function structure, docstring, and loop-based reversal logic.
    - The function is modular, reusable, and easy to maintain.

Task 4: Comparative Analysis – Procedural vs Modular Approach (With vs Without Functions)

Prompt:
"Compare two string reversal programs: one without functions and one with a function, focusing on clarity, reusability, and debugging ease.Provide a short report or table summarizing which approach is better for large-scale applications."

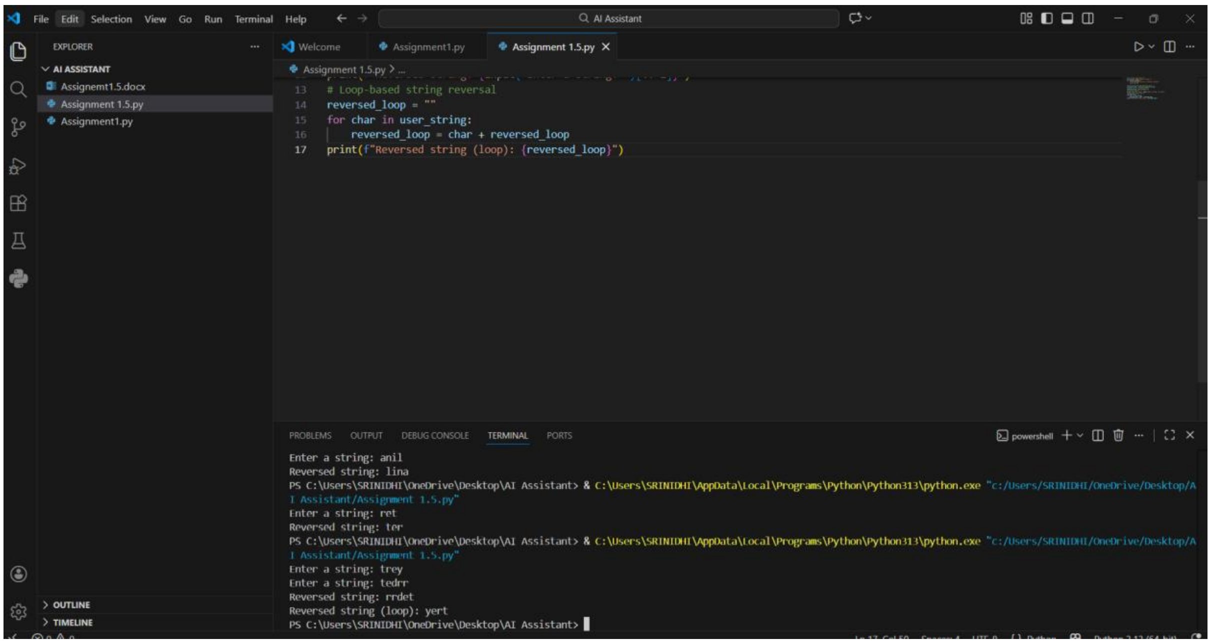
Criteria	**Procedural Approach (No Functions)**	**Modular Approach (With Functions)**
Code Clarity	Simple for small scripts, but logic repeated if used multiple times	Clear structure, function name describes purpose, easier to understand
Reusability	Low – reversal logic cannot be reused without copying code	High – function can be called anywhere in the program
Debugging Ease	Harder for large programs, changes require editing multiple places	Easier – fix or update logic in one function only
Suitability for Large-Scale Applications	Poor – not maintainable or scalable	Excellent – modular design supports large projects
Efficiency	Similar for small scripts, but repeated code adds risk	Similar, but less error-prone, easier to optimize

Task 5: AI-Generated Iterative vs Recursive Fibonacci Approaches (Different Algorithmic Approaches to String Reversal)

Prompt:

"Generate two Python programs to reverse a string: one using a loop (iterative) and one using built-in slicing.Compare execution flow, time complexity, and suitability for large inputs, and explain when each approach is preferable."

Approach 1:Loop-Based String Reversal



```python
# Loop-based string reversal
reversed_loop = ""
for char in user_string:
    reversed_loop = char + reversed_loop
print(f"Reversed string (loop): {reversed_loop}")
```

```
Enter a string: anil
Reversed string: lina
PS C:\Users\SRINIDHI\OneDrive\Desktop\AI Assistant> & C:\Users\SRINIDHI\AppData\Local\Programs\Python\Python313\python.exe "c:/Users/SRINIDHI/OneDrive/Desktop/A
I Assistant/Assignment 1.5.py"
Enter a string: ret
Reversed string: ter
PS C:\Users\SRINIDHI\OneDrive\Desktop\AI Assistant> & C:\Users\SRINIDHI\AppData\Local\Programs\Python\Python313\python.exe "c:/Users/SRINIDHI/OneDrive/Desktop/A
I Assistant/Assignment 1.5.py"
Enter a string: trey
Enter a string: tedrr
Reversed string: rrdet
Reversed string (loop): yert
PS C:\Users\SRINIDHI\OneDrive\Desktop\AI Assistant>
```

Execution Flow

1. Program reads input from the user.
2. Initializes an empty string reversed_string.
3. Iterates through each character of user_input.
4. Prepends the current character to reversed_string.
5. Prints the final reversed string.

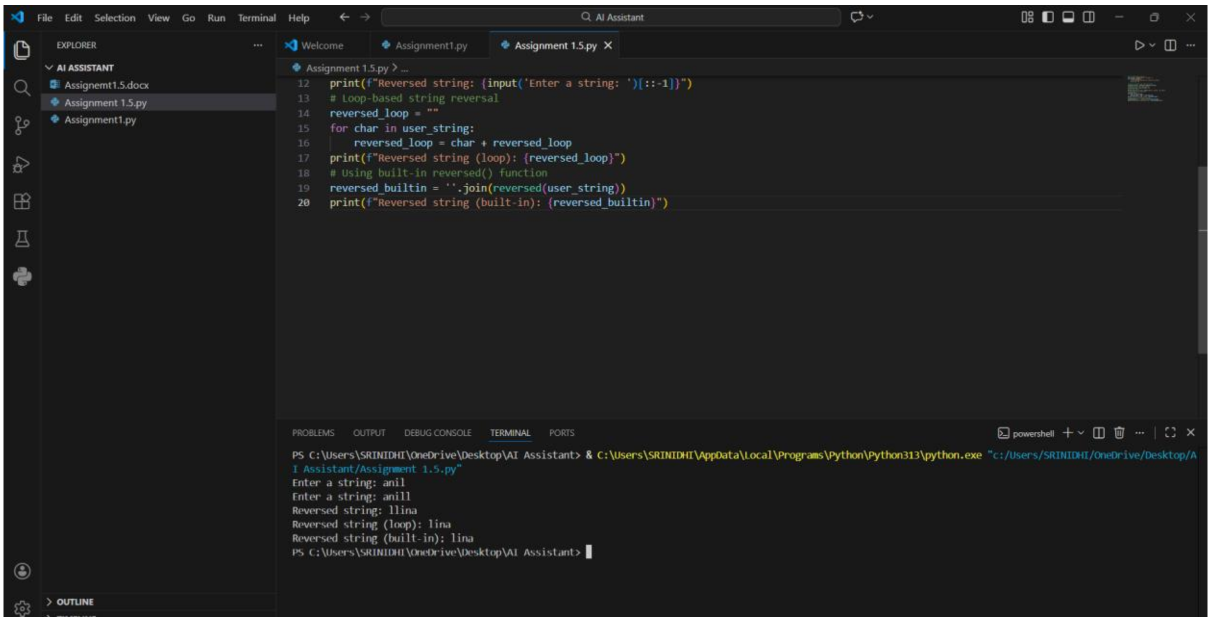
Time Complexity:

- **O(n²)** in languages where string concatenation is costly (like Python) because each char + reversed_string creates a new string.
- For small strings, this is negligible.

Use Case:

- Good for learning purposes and step-by-step logic demonstration.
- Not ideal for very large strings

Approach 2: Built-In / Slicing-Based String Reversal



```python
print(f"Reversed string: {input('Enter a string: ')[::-1]}")
# Loop-based string reversal
reversed_loop = ""
for char in user_string:
    reversed_loop = char + reversed_loop
print(f"Reversed string (loop): {reversed_loop}")
# Using built-in reversed() function
reversed_builtin = ''.join(reversed(user_string))
print(f"Reversed string (built-in): {reversed_builtin}")
```

```
PS C:\Users\SRINIDHI\OneDrive\Desktop\AI Assistant> & C:\Users\SRINIDHI\AppData\Local\Programs\Python\Python313\python.exe "c:/Users/SRINIDHI/OneDrive/Desktop/A
I Assistant/Assignment 1.5.py"
Enter a string: anil
Enter a string: anill
Reversed string: llina
Reversed string (loop): lina
Reversed string (built-in): lina
PS C:\Users\SRINIDHI\OneDrive\Desktop\AI Assistant>
```

Execution Flow

1. Program reads input from the user.
2. Uses Python slicing [::-1] to reverse the string in a single step.
3. Prints the final reversed string.

Time Complexity:

- **O(n)** – very efficient even for large strings.

Use Case:

- Best for production code or large-scale applications.
- Cleaner, shorter, and more maintainable than the loop-based approach.

Comparison Table

Criteria	Loop-Based Approach	Slicing-Based Approach
Code Complexity	Longer, step-by-step	Short, single-line
Execution Flow	Iterative prepending of characters	Direct slicing operation
Time Complexity	O(n^2) (due to repeated concatenation)	O(n)
Performance (Large Input)	Slower, may cause overhead for large strings	Very fast, scales well
Readability	Moderate	High
Use Case	Learning, step-by-step explanation	Production, large applications