Lab Assignment – AI Assisted Coding

Name: M.Adarsh Reddy
Year/Sem: III/II
2303A51687

Task 0: Environment Setup – GitHub Copilot in VS Code

Question

Install and configure GitHub Copilot in VS Code.

Prompt

Install and configure GitHub Copilot in VS Code

Analysis

1. Open VS Code and go to Extensions.
2. Search for 'GitHub Copilot' and install it.
3. Sign in with your GitHub account.
4. Enable Copilot in settings.
5. Test by writing a comment and observing suggestions.

Output

Screenshots of Installation,Login and Activation.

Conclusion

Copilot successfully configured and ready to assist in coding tasks.

Task 1: AI-Generated Logic Without Modularization

Question

Reverse a string without using functions.

Prompt

Write a Python program to reverse a string without using functions

Code

```python
# Reverse a string without using functions

s = input("Enter a string: ")
reversed_string = ""

for i in range(len(s) - 1, -1, -1):
    reversed_string = reversed_string + s[i]

print("Reversed string:", reversed_string)
```

Analysis

Uses loop and indexing. No user-defined functions. Readability is moderate, but correct.

Output

```
Reversed string: yddar
PS C:\Users\adars> & C:/Users/adars/AppData/Local/Programs/Python/Python312/python.exe "c:/Users/adars/1.5_ AI&AC.py"
Enter a string: adarsh
Reversed string: hsrada
```

Conclusion

Procedural approach works correctly but is verbose and less readable.

Task 2: Efficiency & Logic Optimization

Question

Optimize Task 1 code for readability.

Prompt

Simplify this string reversal code

Code

```
# Optimized string reversal for better readability
s = input("Enter a string: ")
reversed_string = ""
for ch in s:
    reversed_string = ch + reversed_string
print("Reversed string:", reversed_string)
```

Analysis

Uses Python slicing. Removes unnecessary variables and loops. Time complexity remains O(n), readability improves.

Output

```
PS C:\Users\adars> & C:/Users/adars/AppData/Local/Programs/Python/Python312/python.exe "c:/Users/adars/1.5_ AI&AC.py"
Enter a string: adarsh
Reversed string: hsrada
Enter a string: reddy
Reversed string: ydder
```

Conclusion

Optimized code is concise, efficient, and easier to maintain.

Task 3: Modular Design Using Functions

Question

Reverse a string using a function.

Prompt

Write a Python function to reverse a string with comments

Code

```
# Function to reverse a string
def reverse_string(text):
    # This function reverses the given string
    reversed_text = ""
    for ch in text:
        reversed_text = ch + reversed_text
    return reversed_text
s = input("Enter a string: ")
result = reverse_string(s)

print("Reversed string:", result)
```

Analysis

Encapsulates logic in a reusable function. Easier debugging and reusability. AI-generated comments improve clarity.

Output

```
PS C:\Users\adars> & C:/Users/adars/AppData/Local/Programs/Python/Python312/python.exe "c:/Users/adars/1.5_ AI&AC.py"
Enter a string: assistant coding
Reversed string: gnidoc tnatsissa
```

Conclusion

Modular design improves maintainability and supports reuse across applications.

Task 4: Comparative Analysis – Procedural vs Modular

Question

Compare Task 1 (procedural) vs Task 3 (modular).

Prompt

Compare procedural vs modular string reversal approaches

Analysis

Comparison of procedural and modular approaches based on clarity, reusability, debugging, and scalability.

Aspect	Procedural (Task 1)	Modular (Task 3)
Code Clarity	Moderate	High

Aspect	Procedural (Task 1)	Modular (Task 3)
Reusability	Low	High
Debugging	Difficult	Easy
Scalability	Poor	Good
Maintainability	Low	High

Conclusion

Modular approach is superior for scalability, clarity, and reusability.

Task 5: Iterative vs Slicing Approaches

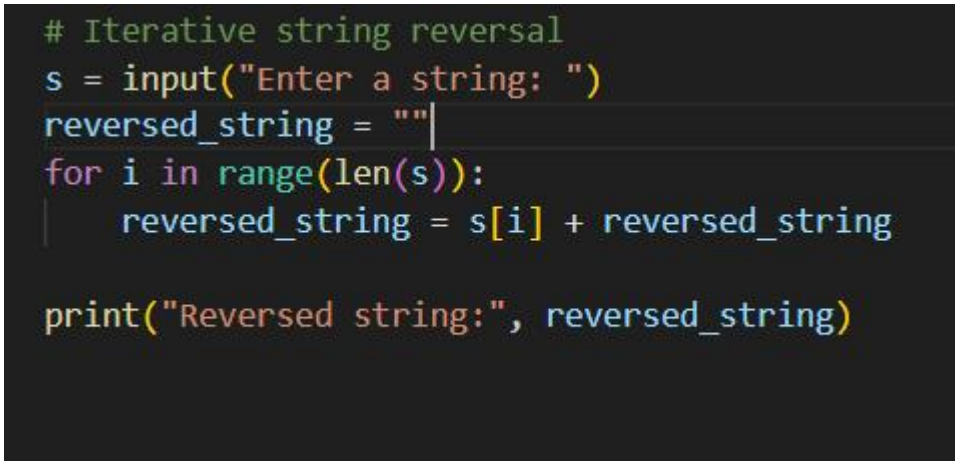
Question

Generate loop-based and slicing-based string reversal.

Prompt

Generate iterative and slicing-based string reversal approaches in Python

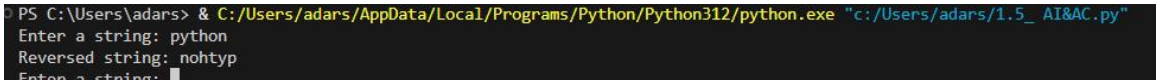
Code



Analysis

Iterative builds reversed string step by step. Slicing uses Python's built-in slicing. Both O(n), but slicing is faster and more concise.

Output



Conclusion

Iterative approach is useful for learning, slicing is preferred in production for efficiency and readability.