Assignment_1.5

Name : Rishik.P

Hall Ticket No: 2303a51778

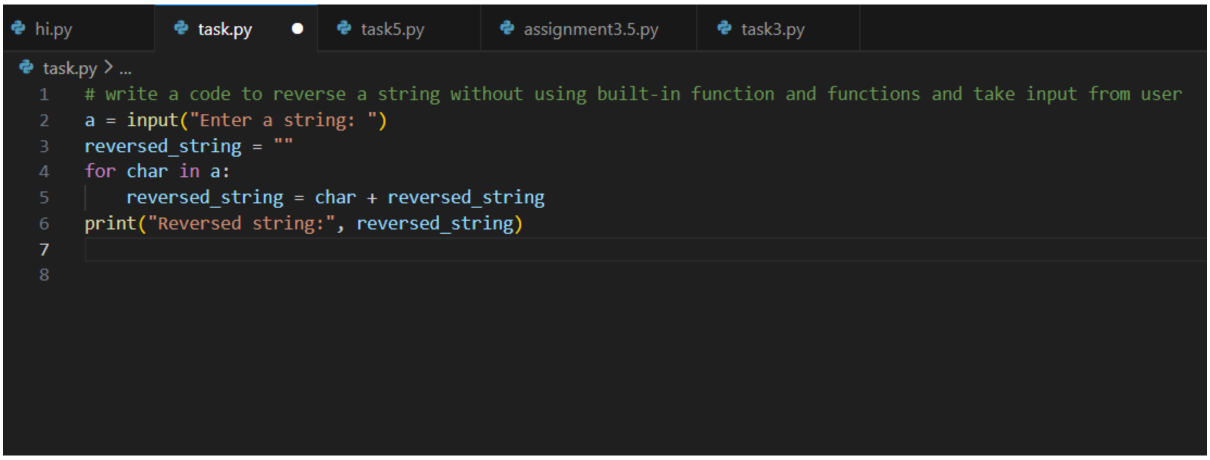
Batch : 12

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

PROMPT:

write a code to reverse a string without using built-in function and functions and take input from user

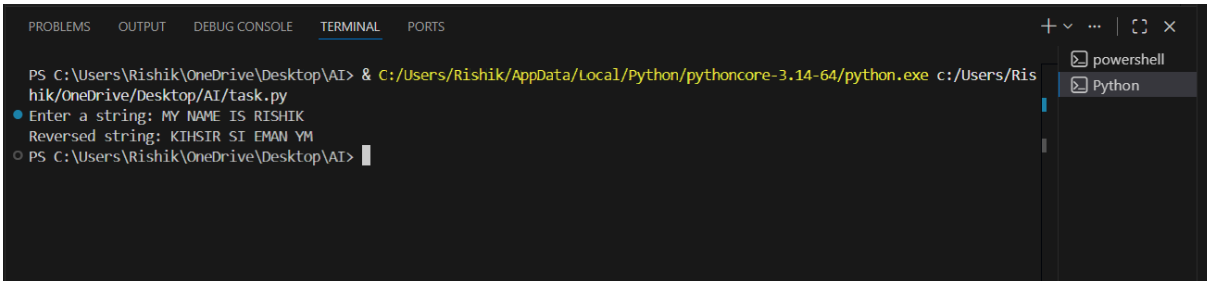
CODE:



```python
# write a code to reverse a string without using built-in function and functions and take input from user
a = input("Enter a string: ")
reversed_string = ""
for char in a:
    reversed_string = char + reversed_string
print("Reversed string:", reversed_string)

```

OUTPUT:



```
PS C:\Users\Rishik\OneDrive\Desktop\AI> & C:/Users/Rishik/AppData/Local/Python/pythoncore-3.14-64/python.exe c:/Users/Ris
hik/OneDrive/Desktop/AI/task.py
Enter a string: MY NAME IS RISHIK
Reversed string: KIHSIR SI EMAN YM
PS C:\Users\Rishik\OneDrive\Desktop\AI> 
```

EXPLANATION:

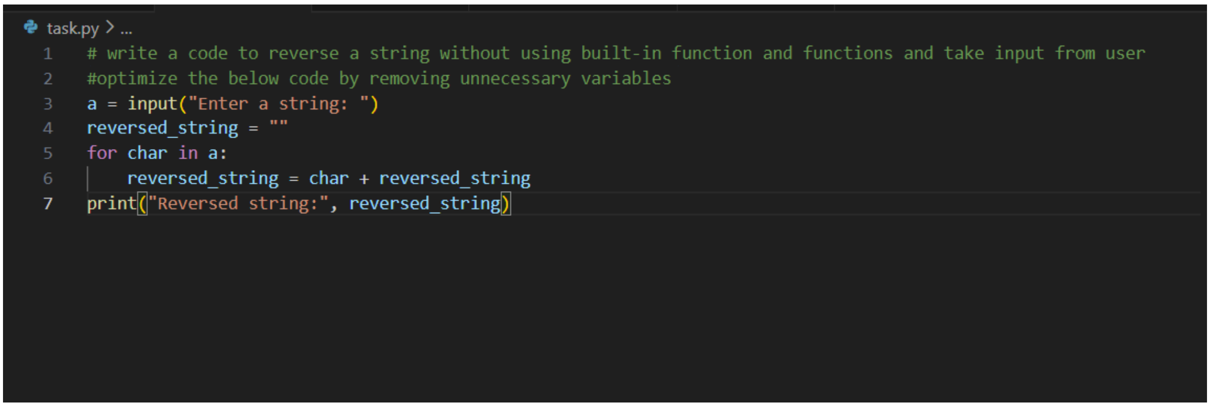
In this task, I understood that the main goal is to reverse a string **without using any user-defined functions**. The entire logic must be written directly in the main part of the program. This helps in understanding how string reversal works at a basic level without hiding the logic inside a function.

The program first takes input from the user. Then, using a loop, it starts reading the string from the last character and keeps adding each character to a new string. By doing this repeatedly, the string gets reversed.

Task 2: Efficiency & Logic Optimization (Readability Improvement)

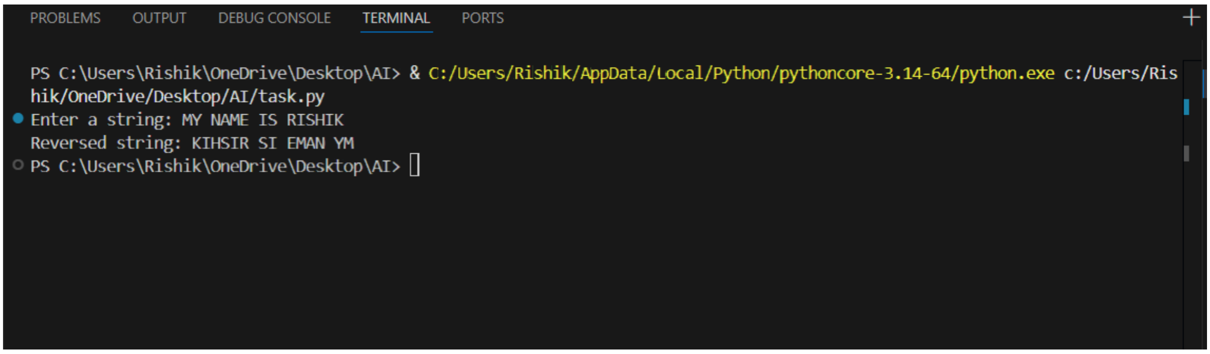
PROMPT: #optimize the below code by removing unnecessary variables

CODE:



```python
# write a code to reverse a string without using built-in function and functions and take input from user
#optimize the below code by removing unnecessary variables
a = input("Enter a string: ")
reversed_string = ""
for char in a:
    reversed_string = char + reversed_string
print("Reversed string:", reversed_string)
```

OUTPUT:



```
PS C:\Users\Rishik\OneDrive\Desktop\AI> & C:/Users/Rishik/AppData/Local/Python/pythoncore-3.14-64/python.exe c:/Users/Ris
hik/OneDrive/Desktop/AI/task.py
Enter a string: MY NAME IS RISHIK
Reversed string: KIHSIR SI EMAN YM
PS C:\Users\Rishik\OneDrive\Desktop\AI> 
```

EXPLANATION:

I analyzed the code generated in Task 1 and tried to make it more readable and simple. I realized that some variables were not really necessary and the logic could be written in a cleaner way.

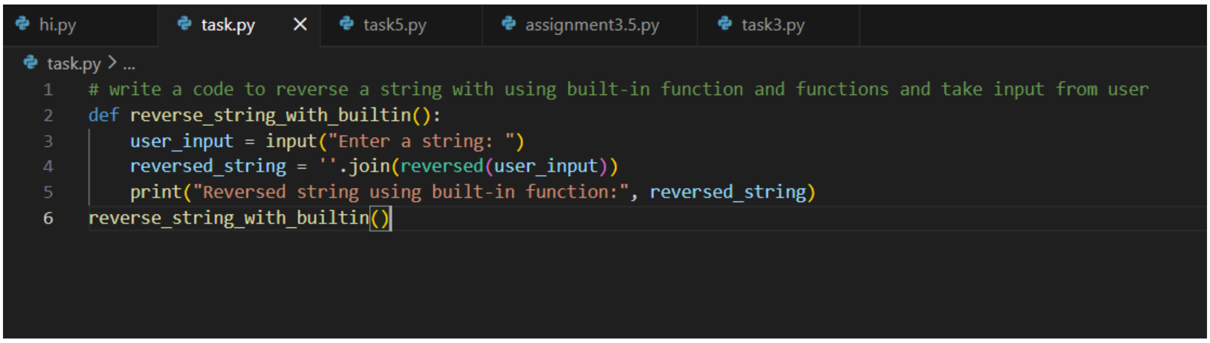
Instead of using indexes and calculating the length of the string, the optimized code directly iterates over each character. This makes the code easier to understand for anyone who reads it later. Even though the time complexity remains the same

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

PROMPT:

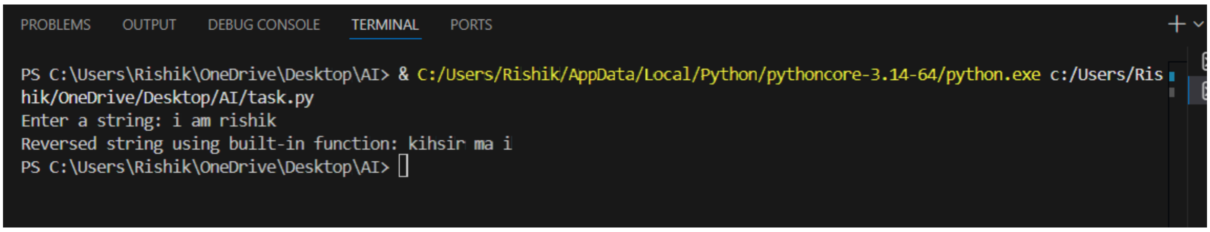
write a code to reverse a string with using built-in function and functions and take input from user

CODE:



```
# write a code to reverse a string with using built-in function and functions and take input from user
def reverse_string_with_builtin():
    user_input = input("Enter a string: ")
    reversed_string = ''.join(reversed(user_input))
    print("Reversed string using built-in function:", reversed_string)
reverse_string_with_builtin()
```

OUTPUT:



```
PS C:\Users\Rishik\OneDrive\Desktop\AI> & C:/Users/Rishik/AppData/Local/Python/pythoncore-3.14-64/python.exe c:/Users/Ris
hik/OneDrive/Desktop/AI/task.py
Enter a string: i am rishik
Reversed string using built-in function: kihsir ma i
PS C:\Users\Rishik\OneDrive\Desktop\AI> 
```

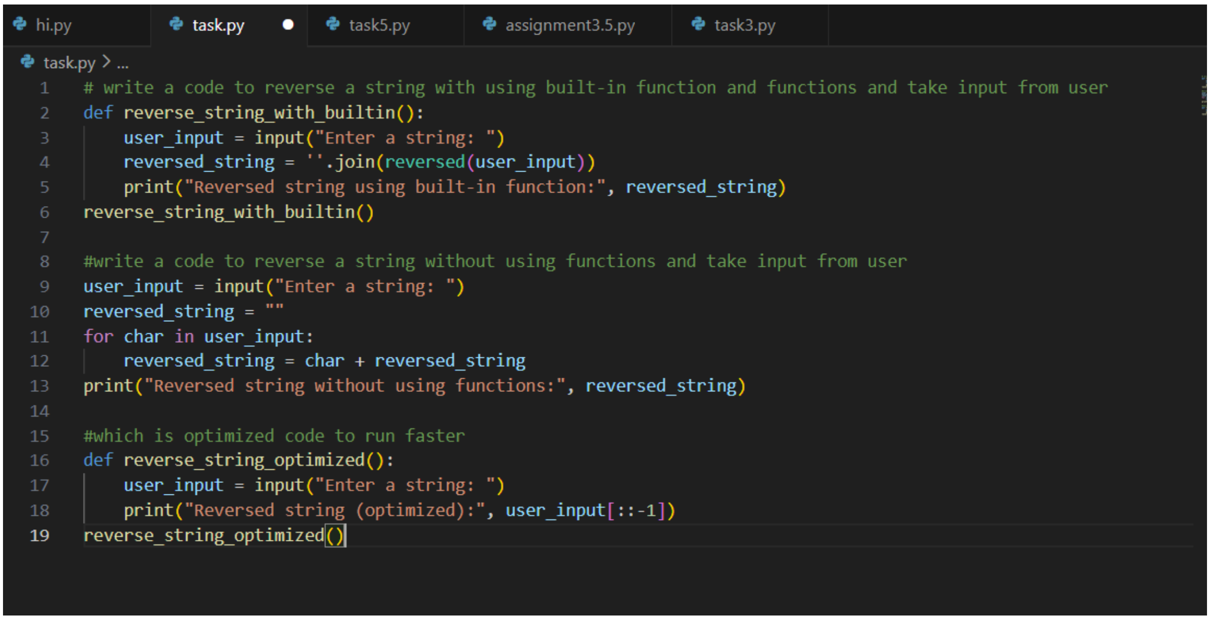
EXPLANATION:

I learned why functions are important in programming. The string reversal logic is written inside a user-defined function, which returns the reversed string. This approach is very useful because the same function can be used in multiple places without rewriting the logic again and again. If there is any mistake, we only need to fix it in one place. This task helped me understand how modular programming improves reusability, clarity, and structure of the program.

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

PROMPT: #which is optimized code to run faster

CODE:

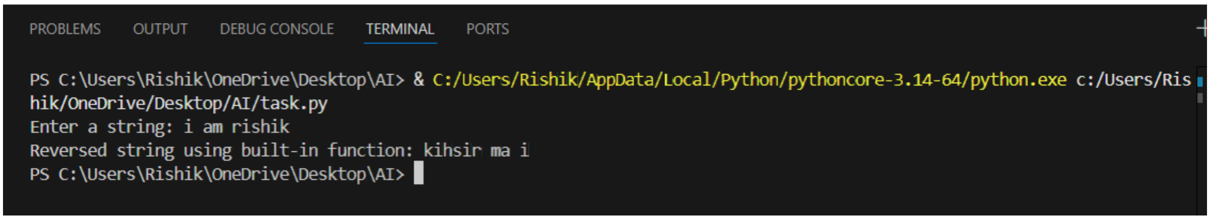


```python
# write a code to reverse a string with using built-in function and functions and take input from user
def reverse_string_with_builtin():
    user_input = input("Enter a string: ")
    reversed_string = ''.join(reversed(user_input))
    print("Reversed string using built-in function:", reversed_string)
reverse_string_with_builtin()

#write a code to reverse a string without using functions and take input from user
user_input = input("Enter a string: ")
reversed_string = ""
for char in user_input:
    reversed_string = char + reversed_string
print("Reversed string without using functions:", reversed_string)

#which is optimized code to run faster
def reverse_string_optimized():
    user_input = input("Enter a string: ")
    print("Reversed string (optimized):", user_input[::-1])
reverse_string_optimized()
```

OUTPUT:



```
PS C:\Users\Rishik\OneDrive\Desktop\AI> & C:/Users/Rishik/AppData/Local/Python/pythoncore-3.14-64/python.exe c:/Users/Ris
hik/OneDrive/Desktop/AI/task.py
Enter a string: i am rishik
Reversed string using built-in function: kihsir ma i
PS C:\Users\Rishik\OneDrive\Desktop\AI>
```

EXPLANATION:

I compared the programs written **with functions and without functions**. From my understanding, writing code without functions is okay for small programs, but it becomes difficult to manage when the program grows. The function-based approach is much better because it improves code clarity and makes debugging easier. Functions allow us to separate logic and make the program well-organized.

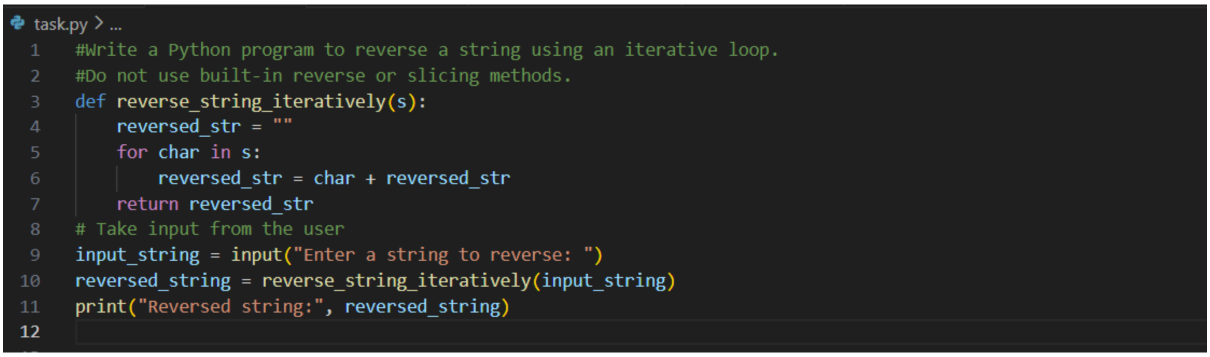
Task 5: AI-Generated Iterative vs Recursive Fibonacci Approaches (Different Algorithmic Approaches to String Reversal)

PROMPT:

#Write a Python program to reverse a string using an iterative loop.

#Do not use built-in reverse or slicing methods.

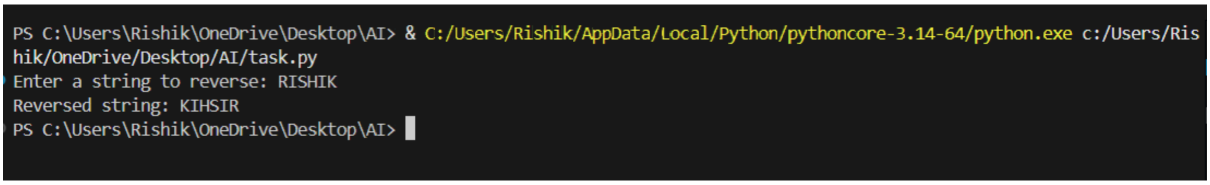
CODE:



```python
#Write a Python program to reverse a string using an iterative loop.
#Do not use built-in reverse or slicing methods.
def reverse_string_iteratively(s):
    reversed_str = ""
    for char in s:
        reversed_str = char + reversed_str
    return reversed_str
# Take input from the user
input_string = input("Enter a string to reverse: ")
reversed_string = reverse_string_iteratively(input_string)
print("Reversed string:", reversed_string)

```

OUTPUT:



```
PS C:\Users\Rishik\OneDrive\Desktop\AI> & C:/Users/Rishik/AppData/Local/Python/pythoncore-3.14-64/python.exe c:/Users/Ris
hik/OneDrive/Desktop/AI/task.py
Enter a string to reverse: RISHIK
Reversed string: KIHSIR
PS C:\Users\Rishik\OneDrive\Desktop\AI> 
```