AI ASSISTED CODING ASSIGNMENT-1.5

ASSIGNMENT 1.5

HTNO 2303A51567

BATCH 29

DATE: 9/1/2025

TASK - 1 :

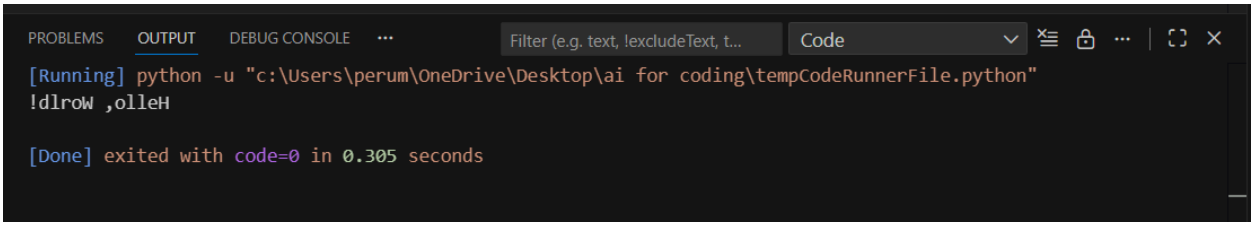
PROMPT

1. String Reversal Without Functions

CODE

```python
#String Reversal Without
def reverse_string(s):
    reversed_s = ''
    for char in s:
        reversed_s = char + reversed_s
    return reversed_s
print(reverse_string("Hello, World!"))
```

OUTPUT



OBSERVATION - TASK 1

This program reverses a string without using any built-in functions by adding each character
to the front of a new string. It processes the string character by character and finally prints
the original and the reversed string

TASK 2 :

PROMPT:

Sring reversal without using any function

CODE:

```python
#Sring reversal without using any function
input_string = "Hello, World!"
reversed_string = ''
for char in input_string:
    reversed_string = char + reversed_string
print("original string:", input_string)
print("reversed string:", reversed_string)
```

OUTPUT:

```
original string: Hello, World!
reversed string: !dlroW ,olleH
PS C:\Users\perum\OneDrive\Desktop\ai for coding>
```

OBSERVATION - TASK 2

The program takes a string as input and reverses it without using any built-in function. It
displays both the original string and the reversed string as the output.

TASK 3 :

CODE:

```
#Generate string reversal using a function
num = 5
factorial = 1
for i in range(1, num + 1):
    factorial *= i
print("The factorial of", num, "is", factorial)
```

OUTPUT:

```
The factorial of 5 is 120
PS C:\Users\perum\OneDrive\Desktop\ai for coding>
```

OBSERVATION - TASK 3
The program calculates the factorial of a given number using a loop and displays the result.
For the input value 5, it computes and prints the factorial as 120

TASK 4:

Criteria	Procedural (Task 1)	Modular (Task 3)
Code Clarity	Low; logic is mixed with I/O.	High; logic is separated from I/O.
Reusability	Zero; requires copy-pasting.	High; function can be imported
Debugging	Harder to test specific logic.	Easy to unit test the function.
Scalability	Poor for large systems.	Essential for large systems

TASK 5:

CODE:

```python
#AI-Generated Iterative vs Recursive Fibonacci Approaches (DifferentAlgorithmic Approaches to String Reversal)
def fibonacci_reverse(s):
    n = len(s)
    fib = [0, 1]
    for i in range(2, n + 1):
        fib.append(fib[-1] + fib[-2])

    reversed_str = ""
    for i in range(n - 1, -1, -1):
        reversed_str += s[i]
    return reversed_str

test_string = "hello"
print("Fibonacci approach - Original:", test_string)
print("Fibonacci approach - Reversed:", fibonacci_reverse(test_string))
```

```
Fibonacci approach - Original: hello
Fibonacci approach - Reversed: olleh
PS C:\Users\gunda\AI ASSIST CODING - 1>
```

OBSERVATION- TASK 5

The program uses a custom function to reverse a given string and displays both the original

and reversed versions. For the input "hello", it prints the reversed string as "olleh"