LAB ASSIGNMENT – 1.5

Name : A .RamyaSri

2303A510I9

Batch:30

Task 1:

AI-Generated Logic Without Modularization (String Reversal Without Functions)

Prompt :

Generate a code for AI-Generated Logic Without Modularization (String Reversal Without Functions)

Code:

C: > Users > Ramya Sri > OneDrive > New folder > lab1.py > ...

```python
#String Reversal Without Functions
input_string = "Hello, World!"
reversed_string = ""
for char in input_string:
    reversed_string = char + reversed_string
print(reversed_string)
```

Output:

```
PS C:\Users\Ramya Sri\OneDrive\New folder\AI Assistant coding> & "C:/Users/Ramya Sri/AppData/Local/Programs/Python/Python313/python.exe" "c:/Users/Ramya Sri/OneDrive/New folder/lab1.py"
!dlroW ,olleH
PS C:\Users\Ramya Sri\OneDrive\New folder\AI Assistant coding>
```

TASK – 2:

Efficiency & Logic Optimization (Readability Improvement)

Prompt:

Generate a code for Efficiency & Logic Optimization (Readability Improvement)

Code:

```
C: > Users > Ramya Sri > OneDrive > New folder > lab1.py > ...
1   # Readability Improvement
2   input_string = "Hello, World!"
3   reversed_string = ""
4   for char in input_string:
5       reversed_string = char+reversed_string
6   print(reversed_string)
7
```

Output:

```
PS C:\Users\Ramya Sri\OneDrive\New folder\AI Assistant coding> & "C:/Users/Ramya Sri/AppData/Local/Programs/
Python/Python313/python.exe" "c:/Users/Ramya Sri/OneDrive/New folder/lab1.py"
!dlroW ,olleH
PS C:\Users\Ramya Sri\OneDrive\New folder\AI Assistant coding>
```

Task – 3:

Modular Design Using AI Assistance (String Reversal Using Functions)

Prompt :

Generate a code for Modular Design Using AI Assistance (String Reversal Using Functions)

Code:

```
C: > Users > Ramya Sri > OneDrive > New folder > lab1.py > ...
1   # String Reversal Using Functions
2   def reversed_string(S):
3       reversed_S = ""
4       for char in S:
5           reversed_S = char + reversed_S
6       return reversed_S
7   input_string = "Hello, World!"
8   reversed_string = reversed_string(input_string)
9   print(reversed_string)
10
11
```

Output:

```
PS C:\> & "C:/Users/Ramya Sri/AppData/Local/Programs/Python/Python313/python.exe" "c:/Users/Ramya Sri/OneDri
ve/New folder/lab1.py"
!dlroW ,olleH
PS C:\Users\Ramya Sri\OneDrive\New folder\AI Assistant coding>
```

Task – 4:

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

Generate a code for Comparative Analysis – Procedural vs Modular Approach (With vs Without Functions)

Code :

C: > Users > Ramya Sri > OneDrive > New folder > lab1.py > ...

```python
# With vs Withput Functions
def reversed_string(S):
    reversed_S = ""
    for char in S:
        reversed_S = char + reversed_S
    return reversed_S
input_string = "Hello, World!"
reversed_string = reversed_string(input_string)
print(reversed_string)

```

Output:

```
PS C:\> & "C:/Users/Ramya Sri/AppData/Local/Programs/Python/Python313/python.exe" "c:/Users/Ramya Sri/OneDri
ve/New folder/lab1.py"
!dlroW ,olleH
PS C:\Users\Ramya Sri\OneDrive\New folder\AI Assistant coding>
```

Task – 5:

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

Prompt :

Generate a code for AI-Generated Iterative vs Recursive Fibonacci Approaches (Different Algorithmic Approaches to String Reversal)

Code:

C: > Users > Ramya Sri > OneDrive > New folder > lab1.py > reverse_string_stack

```python
#Different Algorithmic Approaches to String Reversal
#Using Slicing
input_string = "Hello, World!"
reversed_string = input_string[::-1]
print(reversed_string)

#Using reversed() Built-in
input_String = "Hello, World!"
reversed_string = "".join(reversed(input_string))
print(reversed_string)

#using Recursion
def reverse_string_recursive(s):
    if len(s) == 0:
        return s
    return reverse_string_recursive(s[1:]) + s[0]
input_string = "Hello, World!"
reversed_string = reverse_string_recursive(input_string)
print(reversed_string)

#Using Stack
def reverse_string_stack(s):
    stack = [char for char in s]
    reversed_s = ""
    while stack:
        reversed_s += stack.pop()
    return reversed_s
input_string = "Hello, World!"
reversed_string = reverse_string_stack(input_string)
print(reversed_string)

```

Output:

```
Data/Local/Programs/Python/Python313/python.exe" "c:/Users/Ramya Sri/OneDrive/New folder/lab1.py"
!dlroW ,olleH
!dlroW ,olleH
!dlroW ,olleH
!dlroW ,olleH
```