

## Lab Assignment – 01

**Name:G.Harika**

**HT.NO:2303A51612**

**Batch:05**

### **Task-01:**

#### **Prompt:**

Find Fibanocci series upto n terms without using user defined functions

#### **Code:**

```
n = int(input("Enter the number of terms: "))
a, b = 0, 1
count = 0
if n <= 0:
    print("Please enter a positive integer")
elif n == 1:
    print("Fibonacci sequence upto", n, "term:")
    print(a)
else:
    print("Fibonacci sequence upto", n, "terms:")
    while count < n:
        print(a, end=' ')
```

`c = a + b`

`a = b`

`b = c`

`count += 1`

### **Output:**

Enter the number of terms: 5

Fibonacci sequence upto 5 terms:

0 1 1 2 3

### **Implementation:**

1. The program asks for n and starts the Fibonacci series with  $a = 0$  and  $b = 1$ .
2. If n is 0 or negative, it shows an error; if  $n == 1$ , it prints only 0.
3. Otherwise, it runs a while loop, printing a each time and updating the values using  $c = a + b$  then  $a = b$ ,  $b = c$  and count increases by 1.
4. The loop stops after printing n numbers.



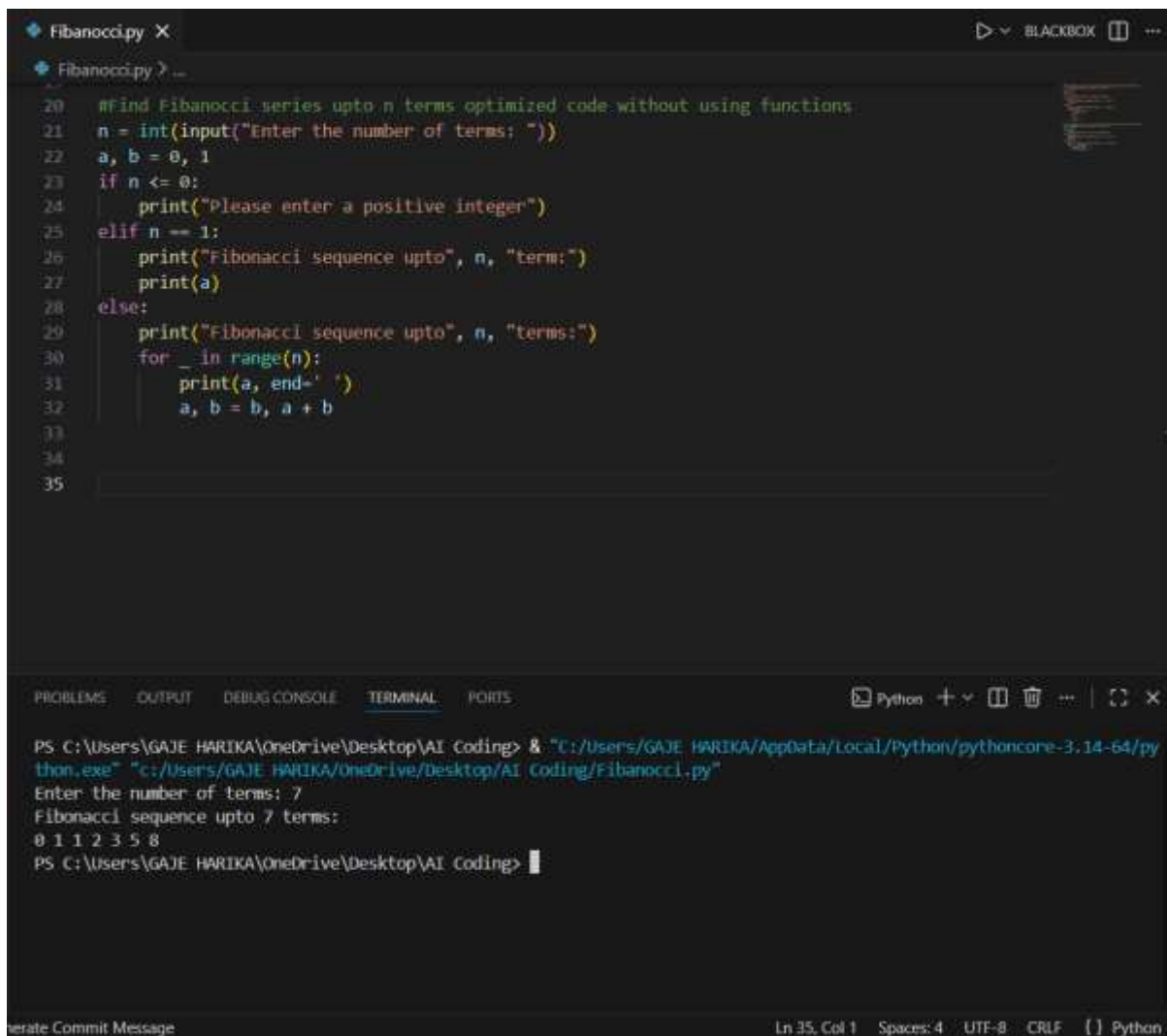
```
    print(a)
else:
    print("Fibonacci sequence upto", n, "terms:")
    for _ in range(n):
        print(a, end=' ')
        a, b = b, a + b
```

### **Output:**

Enter the number of terms: 7  
Fibonacci sequence upto 7 terms:  
0 1 1 2 3 5 8

### **Implementation:**

1. The program takes n from the user and starts with a = 0 and b = 1.
2. If n is not positive, it shows an error message.
3. Otherwise, a for loop runs n times, printing a each time.
4. Inside the loop, the next Fibonacci number is generated using a, b = b, a + b.



```
Fibonacci.py X
Fibonacci.py > ...
20 #find Fibonacci series upto n terms optimized code without using functions
21 n = int(input("Enter the number of terms: "))
22 a, b = 0, 1
23 if n <= 0:
24     print("Please enter a positive integer")
25 elif n == 1:
26     print("Fibonacci sequence upto", n, "term:")
27     print(a)
28 else:
29     print("Fibonacci sequence upto", n, "terms:")
30     for _ in range(n):
31         print(a, end=' ')
32         a, b = b, a + b
33
34
35
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\GAJE HARIKA\OneDrive\Desktop\AI Coding> & "C:/Users/GAJE HARIKA/AppData/Local/Python/pythoncore-3.14-64/python.exe" "C:/Users/GAJE HARIKA/OneDrive/Desktop/AI Coding/Fibonacci.py"
Enter the number of terms: 7
Fibonacci sequence upto 7 terms:
0 1 1 2 3 5 8
PS C:\Users\GAJE HARIKA\OneDrive\Desktop\AI Coding>
```

Generate Commit Message Ln 35, Col 1 Spaces: 4 UTF-8 CRLF Python

## Task-03:

### Prompt:

Find Fibonacci series using Functions returns or prints the sequence upto n terms

### Code:

```
def fibonacci_series(n):
```

```
    a, b = 0, 1
```

```
    series = []
```

```
    for _ in range(n):
```

```
        series.append(a)
        a, b = b, a + b
    return series

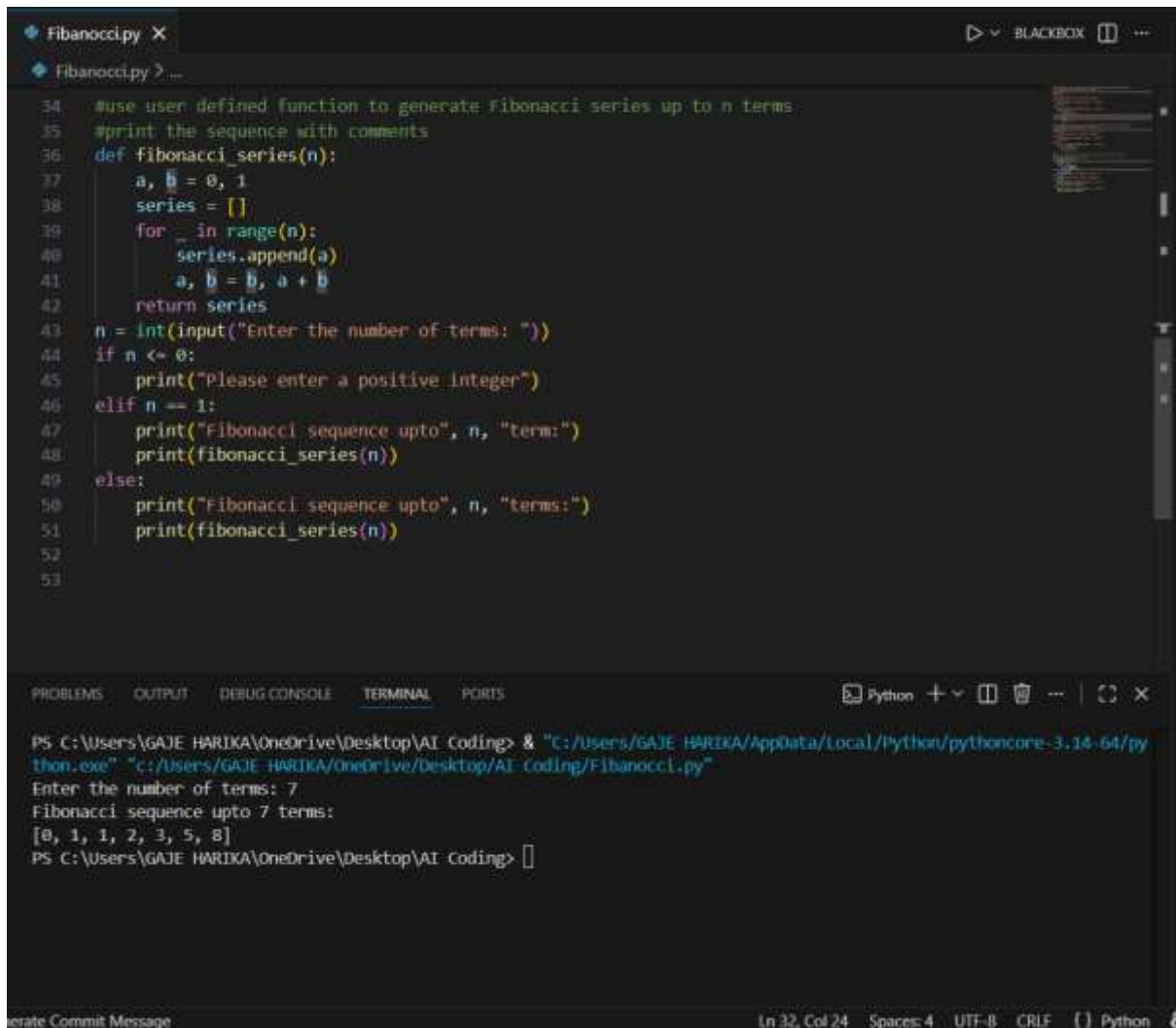
n = int(input("Enter the number of terms: "))
if n <= 0:
    print("Please enter a positive integer")
elif n == 1:
    print("Fibonacci sequence upto", n, "term:")
    print(fibonacci_series(n))
else:
    print("Fibonacci sequence upto", n, "terms:")
    print(fibonacci_series(n))
```

### **Output:**

Enter the number of terms: 7  
Fibonacci sequence upto 7 terms:  
[0, 1, 1, 2, 3, 5, 8]

### **Implementation:**

1. The function starts with 0 and 1 as the first Fibonacci numbers.
2. It repeats n times to create the next numbers.
3. Each number is added to a list.
4. The list is returned and printed as the Fibonacci sequence.



```
34 #use user defined function to generate Fibonacci series up to n terms
35 #print the sequence with comments
36 def fibonacci_series(n):
37     a, b = 0, 1
38     series = []
39     for _ in range(n):
40         series.append(a)
41         a, b = b, a + b
42     return series
43 n = int(input("Enter the number of terms: "))
44 if n <= 0:
45     print("Please enter a positive Integer")
46 elif n == 1:
47     print("Fibonacci sequence upto", n, "term:")
48     print(fibonacci_series(n))
49 else:
50     print("Fibonacci sequence upto", n, "terms:")
51     print(fibonacci_series(n))
52
53
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\GAJE HARIKA\OneDrive\Desktop\AI Coding> & "C:/Users/GAJE HARIKA/AppData/Local/Python/pythoncore-3.14-64/python.exe" "C:/Users/GAJE HARIKA/OneDrive/Desktop/AI Coding/Fibanoccl.py"
Enter the number of terms: 7
Fibonacci sequence upto 7 terms:
[0, 1, 1, 2, 3, 5, 8]
PS C:\Users\GAJE HARIKA\OneDrive\Desktop\AI Coding>
```

Ln 32, Col 24 Spaces: 4 UTF-8 CRLF Python

## **Task-04:**

### **Prompt:**

# Fibonacci series with Procedural vs Modular Fibonacci Code  
AI code with and without

# Procedural approach

### **Code:**

```
def fibonacci_series(n):
```

```
    a, b = 0, 1
```

```
    series = []
```

```
    for _ in range(n):
        series.append(a)
        a, b = b, a + b
    return series

n = int(input("Enter the number of terms: "))

if n <= 0:
    print("Please enter a positive integer")
elif n == 1:
    print("Fibonacci sequence upto", n, "term:")
    print(fibonacci_series(n))
else:
    print("Fibonacci sequence upto", n, "terms:")
    print(fibonacci_series(n))
```

### **Output:**

```
Enter the number of terms: 5
Fibonacci sequence upto 5 terms:
[0, 1, 1, 2, 3]
```

### **Prompt:**

```
# Modular approach
```

### **Code:**

```
def get_fibonacci_series(n):
```



```
a, b = 0, 1
series = []
for _ in range(n):
    series.append(a)
    a, b = b, a + b
return series

def main():
    n = int(input("Enter the number of terms: "))
    if n <= 0:
        print("Please enter a positive integer")
    elif n == 1:
        print("Fibonacci sequence upto", n, "term:")
        print(get_fibonacci_series(n))
    else:
        print("Fibonacci sequence upto", n, "terms:")
        print(get_fibonacci_series(n))

if __name__ == "__main__":
    main()
```

**Output:**

Enter the number of terms: 7

Fibonacci sequence upto 7 terms:

[0, 1, 1, 2, 3, 5, 8]

### Implementation:

```
Fibonacci.py X
Fibonacci.py > main
72 # Modular approach
73 def get_fibonacci_series(n):
74     a, b = 0, 1
75     series = []
76     for _ in range(n):
77         series.append(a)
78         a, b = b, a + b
79     return series
80 def main():
81     n = int(input("Enter the number of terms: "))
82     if n <= 0:
83         print("Please enter a positive integer")
84     elif n == 1:
85         print("Fibonacci sequence upto", n, "term:")
86         print(get_fibonacci_series(n))
87     else:
88         print("Fibonacci sequence upto", n, "terms:")
89         print(get_fibonacci_series(n))
90 if __name__ == "__main__":
91     main()
92

PS C:\Users\GAJE HARIKA\OneDrive\Desktop\AI Coding> & "C:/Users/GAJE HARIKA/AppData/Local/Python/pythoncore-3.14-64/python.exe" "C:/Users/GAJE HARIKA/OneDrive/Desktop/AI Coding/Fibonacci.py"
Enter the number of terms: 7
Fibonacci sequence upto 7 terms:
[0, 1, 1, 2, 3, 5, 8]
PS C:\Users\GAJE HARIKA\OneDrive\Desktop\AI Coding>
```

### Task-05:

**Prompt:**

## #AI-Generated Iterative vs Recursive Fibonacci Approaches (Different Algorithmic Approaches for Fibonacci Series)

## # Iterative approach

**Code:**

```
def fibonacci_iterative(n):
```

 $a, b = 0, 1$

```
series = []
for _ in range(n):
    series.append(a)
    a, b = b, a + b
return series

n = int(input("Enter the number of terms: "))
if n <= 0:
    print("Please enter a positive integer")
elif n == 1:
    print("Fibonacci sequence upto", n, "term:")
    print(fibonacci_iterative(n))
else:
    print("Fibonacci sequence upto", n, "terms:")
    print(fibonacci_iterative(n))
```

### **Code:**

Enter the number of terms: 8

Fibonacci sequence upto 8 terms:

[0, 1, 1, 2, 3, 5, 8, 13]

### **Implementation:**

- 1.The function is defined to generate Fibonacci numbers.
2. a and b are initialized to 0 and 1.
3. A loop runs n times.

4. Each Fibonacci number is added to a list.

5.The list is returned and printed.

The screenshot shows a Visual Studio Code window titled "Fibonacci.py X". The editor displays a Python script with the following code:

```
93 #AI-Generated Iterative vs Recursive Fibonacci Approaches (Different  
94 #Algorithmic Approaches for Fibonacci Series)  
95 # Iterative approach  
96 def fibonacci_iterative(n):  
97     a, b = 0, 1  
98     series = []  
99     for _ in range(n):  
100         series.append(a)  
101         a, b = b, a + b  
102     return series  
103 n = int(input("Enter the number of terms: "))  
104 if n <= 0:  
105     print("Please enter a positive integer")  
106 elif n == 1:  
107     print("fibonacci sequence upto", n, "term:")  
108     print(fibonacci_iterative(n))  
109 else:  
110     print("Fibonacci sequence upto", n, "terms:")  
111     print(fibonacci_iterative(n))  
112  
113
```

The bottom panel shows the TERMINAL tab with the following output:

```
PS C:\Users\GAJE HARIKA\OneDrive\Desktop\AI Coding> & "C:/Users/GAJE HARIKA/AppData/Local/Python/pythoncore-3.14-64/py  
thon.exe" "c:/Users/GAJE HARIKA/OneDrive/Desktop/AI Coding/Fibonacci.py"  
Enter the number of terms: 8  
Fibonacci sequence upto 8 terms:  
[0, 1, 1, 2, 3, 5, 8, 13]  
PS C:\Users\GAJE HARIKA\OneDrive\Desktop\AI Coding>
```

**Prompt:**

## # Recursive approach

**Code:**

```
def fibonacci_recursive(n):
```

```
if n <= 0:
```

```
return []
```

```
elif n == 1:
    return [0]
elif n == 2:
    return [0, 1]
else:
    series = fibonacci_recursive(n - 1)
    series.append(series[-1] + series[-2])
    return series

n = int(input("Enter the number of terms: "))
if n <= 0:
    print("Please enter a positive integer")
elif n == 1:
    print("Fibonacci sequence upto", n, "term:")
    print(fibonacci_recursive(n))
else:
    print("Fibonacci sequence upto", n, "terms:")
    print(fibonacci_recursive(n))
```

### **Output:**

Enter the number of terms: 6  
Fibonacci sequence upto 6 terms:  
[0, 1, 1, 2, 3, 5]

### **Implementation:**

1. The function `fibonacci_recursive(n)` is called.
2. If `n` is 1 or 2, it returns the base Fibonacci values.
3. If `n` is greater than 2, the function calls itself with `n-1`.
4. The last two numbers are added to get the next Fibonacci number.
5. The updated list is returned and printed.

```
Fibonacci.py X
Fibonacci.py > ...
113 #Recursive approach
114 def fibonacci_recursive(n):
115     if n <= 0:
116         return []
117     elif n == 1:
118         return [0]
119     elif n == 2:
120         return [0, 1]
121     else:
122         series = fibonacci_recursive(n - 1)
123         series.append(series[-1] + series[-2])
124         return series
125 n = int(input("Enter the number of terms: "))
126 if n <= 0:
127     print("Please enter a positive integer")
128 elif n == 1:
129     print("Fibonacci sequence upto", n, "term:")
130     print(fibonacci_recursive(n))
131 else:
132     print("Fibonacci sequence upto", n, "terms:")
133     print(fibonacci_recursive(n))
134
135
PS C:\Users\GAJE HARIKA\OneDrive\Desktop\AI Coding> & "C:/Users/GAJE HARIKA/AppData/Local/Python/pythoncore-3.14-64/python.exe" "C:/Users/GAJE HARIKA/OneDrive/Desktop/AI Coding/Fibonacci.py"
Enter the number of terms: 6
Fibonacci sequence upto 6 terms:
[0, 1, 1, 2, 3, 5]
PS C:\Users\GAJE HARIKA\OneDrive\Desktop\AI Coding>
```

