**TASK – 1**

**Fibonacci program**

def fibonacci(n):

This line defines a function named **fibonacci** that takes one argument **n**. This function is designed to generate the Fibonacci sequence up to the nth term.

if n <= 0:

return []

These are the base cases of the Fibonacci sequence. If **n** is less than or equal to 0, an empty list is returned. If **n** is 1, **[0]** is returned, and if **n** is 2, **[0, 1]** is returned. These cases handle the initial conditions of the sequence.

elif n == 1:

return [0]

elif n == 2:

return [0, 1]

else:

If **n** is greater than 2, this block of code is executed. It initializes the Fibonacci sequence **fib\_sequence** with the first two terms **[0, 1]**. Then, it iterates from the third term (**i=2**) up to **n-1** using a **for** loop.

fib\_sequence = [0, 1]

for i in range(2, n):

fib\_sequence.append(fib\_sequence[-1] + fib\_sequence[-2])

In each iteration, it calculates the next Fibonacci number by summing the last two numbers in the sequence (**fib\_sequence[-1]** and **fib\_sequence[-2]**) and appends the result to **fib\_sequence**. Finally, the complete Fibonacci sequence is returned.

return fib\_sequence

# Example usage:

n = 10

fib\_sequence = fibonacci(n)

It sets **n** to 10 and calls the **fibonacci** function with **n** as an argument. The resulting Fibonacci sequence is stored in the variable **fib\_sequence**, which is then printed.

print(fib\_sequence)

**Output: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377**