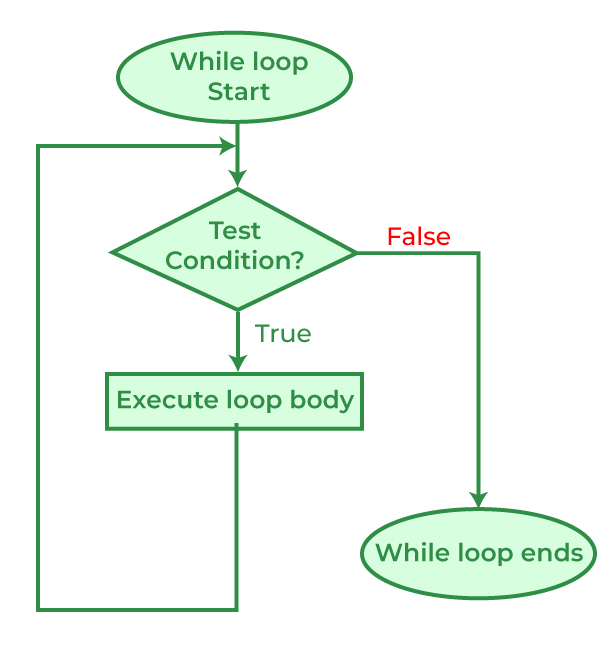
Exercise 84:

**Operating Mechanism of a while Loop**

A while loop in Python continues to execute a block of code as long as a specified condition is True. The loop checks the condition before each iteration, and if the condition becomes False, the loop stops.

**Flowchart of the while Loop**



**Explanation of the while Loop Mechanism**

* **Condition check**: The loop begins by checking the condition.
* **If the condition is True**, the code inside the loop (statements) is executed.
* **If the condition is False**, the loop exits, and the program continues with the next part of the code.
* The loop condition is often updated within the loop to prevent it from running infinitely.

**Example of a while Loop**

# Example: Countdown using a while loop

count = 5

while count > 0:

print(f"Counting down: {count}")

count -= 1

print("Blast off!")

# Output:

# Counting down: 5

# Counting down: 4

# Counting down: 3

# Counting down: 2

# Counting down: 1

# Blast off!

**Explanation of the Example:**

* **Initial Condition**: The loop starts with count = 5 and checks if count > 0. Since this condition is True, the loop enters the block of code.
* **Statements**: Inside the loop, it prints the current value of count and decreases count by 1 (count -= 1).
* The loop repeats this process until count becomes 0.
* **Loop ends**: Once count becomes 0, the condition count > 0 is False, so the loop stops, and the program prints "Blast off!".