Bug #1 - The division for the average is integer division, not floating point, and the double on the left doesn't change that.

Description:

The program calculated the average incorrectly because the division used integer division instead of floating-point

Steps to reproduce:

1. Compile the program
2. Run it with the numbers that do not average to a whole number (10, 20, 25)
3. Observe that the displayed average is truncated

Expected behavior: The average should include decimal precision when needed.

Actual behavior: The average loses precision and always displays an integer value instead of the correct decimal result

Bug Resolution

Root Cause: The expression (num1 + num2 + num3) /3 was evaluated using integer division because all operands were integers, causing the fraction to be dropped.

Fix Summary: Changed the divisor to 3.0 from 3 to force floating-point division

Bug #2 – Incorrect Values in the sum_even for loop

Description : The loop calculates that calculates the sum of even numbers did not match the description; first, it included 0(which is not positive) and also did not include 6, when the message says <=6

Steps to Reproduce:

1. Compile and run the program
2. Look at the output for " Sum of positive even numbers <= 6:
3. Manually compute 2 + 4 + 6

Expected Behavior: The program should sum all positive even numbers <=6 (2 + 4 + 6 = 12) and display 12

Actual Behavior: The loop only iterated from 0 to 5 so it included 0 and excluded 6, The Printed sum did not reflect the printed statement

Bug Resolution

Root Cause: The loop bounds were incorrect, it started at 0(which is not positive) and stopped at 5, which excluded 6. This made the logic inconsistent with the requirement to sum positive even numbers up to and including 6.

Fix Summary: Updated the loop to start at 2 and end at 6 (for(int i = 2; i <= 6; i++)) so it correctly sums 2,4, and 6 as the positive even numbers up to and equal to 6.

Bug #3 – Incorrect Terms for finding Positive Value

Description: The program incorrectly classifies 0 as a positive number

Steps to Reproduce:

1. Set value = 0 or run the program with 0
2. Observe the output printed as positive

Expected Behavior: 0 should not be considered positive. Only numbers greater than 0 are positive so with a value of 0 the program should print “0 is not positive”

Actual Behavior: The condition used for positive numbers includes 0 “|| value ==0)” so it will print “0 is positive”.

Bug Resolution

Root Cause: The condition for the positive statement incorrectly included 0 in the set of positive values

Fix Summary: Removed the “|| value == 0” portion of the condition so that it only lets values greater than 0 through the condition.