

EECS 348 Assignment 8 Bug report

12/2/2025

Vincent Vordtriede

Bug documentation

1. Bug #1 lies in the first block of the source code where integer division is performed instead of floating-point division. This can cause incorrect answers if the average of 3 numbers does not result in a perfect integer, as the resulting decimal number will be rounded down to the nearest integer.
 - Steps to reproduce: First, change the three numbers to anything that would result in an average that results in a non-integer (Ex: change num3 to 32).
 - Expected behavior: The average of 10,20, and 32 is 20.6667
 - Actual behavior: The program prints out that the average is 20, which is incorrect.
2. Bug #2 lies in the second block of the source code where the incorrect number of ints is looped through in the for loop. The for loop only goes up to 5, but it is supposed to loop up to 6.
 - Steps to reproduce: Run the original source code
 - Expected behavior: Sum of positive even numbers <= 6: 12
 - Actual behavior: Sum of positive even numbers <= 6: 6
3. Bug #3 lies in the third block of the source code where the logic in the if statement for checking positive numbers is incorrect. The code checks if the given value is both greater than zero or if it is equal to zero. This is incorrect as inputting a value of zero would cause the code to say '0 is positive'
 - Steps to reproduce: Change the inputted value to 0 and run the original source code.
 - Expected behavior: 0 is not positive
 - Actual behavior: 0 is positive

Bug Resolution

1. The underlying issue of bug #1 is that integer division was performed instead of floating-point division. This causes non-integer averages to be rounded down to the nearest integer instead of displaying the actual correct average.
 - This can be fixed by replacing '\3' with '\3.0' in the code

-
2. The underlying issue of bug #2 is that the for loop's condition is incorrect for looping through all numbers from 0-6 (inclusive). This causes the incorrect even sum to be computed.
 - This can be fixed by changing the for loops ending condition from 'i <= 5' to 'i <= 6'
3. The underlying issue of the final bug is that the Boolean statement in that is checked in the if statement is logically incorrect and causes an inputted value of 0 to be displayed as a positive number, which is incorrect.
 - This can be fixed by changing the comparison from 'if (value > 0 || value == 0)' to 'if (value > 0)'