Eduardo Ramirez Torres

UID: 205-317-677

CS31

**Report**

Step 5:

The given code runs as it was intended. However, there are some inputs that are still accepted and do not make sense. For example, inputting 100 for people surveyed and stating that 5000000000000000 people supported impeachment while 38428948349834 were against it. It still produced percentages, but they were way above 100%, which does not make sense in the context of the question. Running the same inputs on the Linux server, the program does not read the input for people against impeachment; rather, it skipped that input and gave it a very negative percentage. Either way, the numbers would produce a nonsensical output.

Step 6:

The logic error I introduced to the original source code was adding an extra 0 to the end of the percentage calculation for people supporting impeachment. When the user inputs logically sound numbers into the program, they will realize that the percentage of people for impeachment is 10 times too big. The code was able to compile, but no matter what the user inputs, the output will not be as intended originally.

Step 7:

On line 13 of the code, I removed a semicolon in order to prevent the program from compiling. By removing the semicolon, the compiler does not recognize the end of the line of code and will keep on reading until another semicolon is detected. This results in a compilation error. Another way of producing a compiling error is by changing the arrows for “cout” on line 17. Normally, all output streams require two less than (<<) symbols to actually print out the consecutive output on the console. However, by reversing the symbols (>>), the compiler detects the mistake and will not compile unless it is fixed. These are two distinct mistakes that result in compiling errors.