**Step 5 – Reasonable Inputs, Nonsensical Outputs:**

I inputted 0 for each of the inputs (total surveyed voters, Newsom voters, and Cox voters).



As a result, the program output was undefined as the percent calculations for each candidate involve dividing by the total number of surveyed voters. Dividing by 0 yields a nonsensical, undefined answer (nan%) as the output.

I also inputted negative numbers when asked to input how many voters voted for each of the candidates.



By inputting negative numbers for the number of voters for each candidate (-5 – Newsom, -9 – Cox) and keeping the total surveyed voters positive (50), the percent of voters for each candidate are negative which is illogical since you can’t have a negative percentage of a group.

**Step 6 – Logic Errors:**

I changed the program to assign the user inputs to the wrong variables. The number the user inputs for how many voters will vote for Newsom is stored in the variable forCox and the number the user inputs for how many voters will vote for Cox is stored in the variable forNewsom. As a result of this logical error, the program flips the results and wrongly predicts the winner relative to the user input.



In addition to this, I also changed the greater than sign to a less than sign in the if-statement that checks to see if Newsom had more votes than Cox. As a result, the program predicts Newsom to win if Newsom has less votes than Cox (forNewsom < forCox) which is an illogical conclusion.

**Step 7 – Compile Errors:**

The first compile error I introduced was to remove the int identifier for numberSurveyed, forNewsom, and forCox. This mistake yields the error: “use of undeclared identifier” for each of the variables and fails to compile as variables must have an identifier to be interpreted properly by the C++ compiler.



The second compile error I introduced was switching the alligator brackets for the cout and cin commands. Since this is invalid syntax, the compiler is unable to properly interpret the cout and cin commands and exits with an error that invalid operands were used in the code.

The third compile error I introduced was removing a semicolon at the end of a line of code. Semicolons are statement separators that indicate the end of an expression, and without them, the compiler is unable to distinguish when an expression is completed. As a result, the compiler exits with an error indicating that a semicolon is expected.