Name: Marc Luzuriaga

Student ID #: 205916585

**Step 5:**

* If I input a negative number into one of the inputs in the original program, the program returns a negative percentage of the electoral college who voted for a person. For instance, if I input that there were 10 electors in the electoral college; and, out of the 10 electors, -5 voted for the republican party candidate and 5 voted for the democratic party candidate, the program would return that -50.0% of the 10 electors voted for the Republican party candidate and 50.0% of the 10 electors voted for the Democratic party candidate. Yet, this is impossible because negative electors to vote for a candidate is not possible. Therefore, this is a nonsensical input. A solution to this problem could be to add a positive range of possible numbers, and the program would keep the user at the question until their input matches a positive number.
* Input:
  + How many Electoral College electors are there? 10
  + How many of these electors voted for the Republican party candidate? -5
  + How many of these electors voted for the Democratic party candidate? 5
* Output:
  + -50% of the 10 electors voted for the Republican party candidate.
  + 50% of the 10 electors voted for the Democratic party candidate.
  + Looks like the Democratic party candidate won!
  + Look like some of these data values don’t make sense

**Step 6:**

* An error that someone could make would be forgetting to convert one of the votes to a percentage. In this step, I removed the “100.0” from the “double pctRepub” variable. This made the line in the program be as such, “double pctRepub = (republicanVotes / totalVoters;”. As a result, when I ran the program and entered valid inputs, the votes would result in a “0.0% of the 10 electors voted for the Republican party candidate,” no matter what the input was for “how many of those electors voted for the Republican party candidate.” For instance, if I inputted that 9 electors voted for the Republican party candidate while 1 of them voted for the Democratic party candidate, the output of the program would be “0.0% of the 10 electors voted for the Republican party candidate.” Consequently, this output showed that the democratic party candidate won even though the republican party was supposed to win in this case.
* Input:
  + How many Electoral College electors are there? 10
  + How many of these electors voted for the Republican party candidate? 9
  + How many of these electors voted for the Democratic party candidate? 1
* Output:
  + 0.0% of the 10 electors voted for the Republican party candidate.
  + 10.0% of the 10 electors voted for the Democratic party candidate.
  + Looks like the Democratic party candidate won!

**Step 7:**

* Error 1: An error that I introduced to the program was removing the double less-than sign in “cout << “How many Electoral College electors are there? “;. Therefore, the current line is written as, “cout < “How many Electoral College electors are there? “;. This resulted in a “build failed” output. In order for the program to run, a double less-than sign must be introduced after “cout.” This may be a common error while writing a program because of the attention to detail of adding another less-than sign.
* Error 2: The second error that I made was not including a semicolon at the end of the statement, “double pctDemoc = (100.0 \* democraticVotes) / totalVoters”. This resulted in a “build failed” output. Semicolons are used in C++ to let the compiler know that it has reached the end of the command. Because there was no semicolon, the compiler was not able to recognize that it had reached the end of the command.