**Different Inputs And Their Outcomes: (Step 5)**

Case 1: Put registered voters to an unreasonably large number. Eg. 100000000000000000. This results in a program that runs without any further input and gives 0% vote for either but says Neel wins.

Case 2: Put registered voters as a certain number such as 20, and then put the number for Jerry and Neel such that their sum exceeds the registered voters. The results then count over 100% but correctly attribute the rightful winner

Case 3: Make the results such that Neel and Jerry have equal amounts of voters between them such that it’s a 50/50 split. However the program still suggests one (Neel) winning over the other, because it does not include a “draw” option.

Case 4: In the case of negative integers, the result is different. When applied with negative registered voters, and negative votes for Jerry and Neel respectively, the one with a lower percentage of voters wins.

Case 5: Put the number of registered voters to a certain value and give both Jerry and Neel the exact same value such that both have 100% of voters. The result gives a winner (Neel) instead of an error in the voting process.

**Logic Errors: (Step 6)**

Error 1 and 2: The first two errors are both for the same candidate, Jerry. The amalgamation of the two errors is such that the output for those that voted for jerry is no longer a valid percentage., but is a sum followed by a multiplication of voters.

Error 3: This applies to the voters for Neel, in that instead of being 100 times the fraction, it is 10 such that the percentage of voters for Neel is down by a factor of 10, given an invalid overall percentage

Error 4: It is simply a mistake in the greater than or less than sign in the ‘if else’ statement such that if Jerry has a greater portion of voters for him, Neel will win.

**Compile Error: (Step 7)**

Error 1: In the very first output message of how many voters there should be, the error is only 1 ‘<’ sign instead of the required ‘<<’.

The error messages that pops up is as follows:

***Result of comparison against a string literal is unspecified (use strncmp instead)***

***Invalid operands to binary expression (‘ostream’ (aka ‘basic\_ostream<char>’) and ‘const char\*’)***

Error 2: In the “magic formula” the output is set using a ‘<<’ operand instead of the required ‘.’.

The error message that pops up is as follows:

***Use of undeclared identifier ‘setf’***

Error 3: Is simply a missing semicolon at the end of one of the statements.

The error message that pops up is as follows:

***Expected ‘;’ after expression***

Error 4: Is the misuse of ‘out’ instead of ‘cout’.

The error message that pops up is as follows:

***Use of undeclared identifier ‘out’***