The input I provided in original.cpp that caused the program to output incorrect results was “100” in response to the prompt “How many UK citizens were surveyed?”, “50” in response to the prompt “How many of them want to remain in the EU?”, and “50” in response to “ How many of them want to leave the EU?”. The result was the correct output that 50% are Remainers and 50% are Leavers, but also the incorrect output that more people want to leave than remain. This logic error occurred because the source code did not account for the possibility that the number of people who want to leave and the number of people who want to remain are equal to each other. The else statement incorrectly assumed that the only other possibility to having more people wanting to remain than leave is for more people wanting to leave than remain.

The error I introduced in logic\_error.cpp was to change the calculation of the percentage of UK citizens who want to remain in the EU by altering the 100 in the source code so that “double pctRemainers” was instead equivalent to “80.0 \* numberOfRemainers / numberOfLeavers”. This made the calculation of the percentage value incorrect. For example, when I inputted the values, 100, 90, 10 at the respective prompts, the output was that 72% wanted to remain when it should have been 90%. The change in the source code of that line of computation still allowed the program to build and run successfully, but the results were incorrect given reasonable input.

The first error I introduced in compile\_error.cpp was initializing the integer variable numberSurveyed as “this is not an int”. Xcode gave me an error of “Cannot initialize a variable of type 'int' with an lvalue of type 'const char [19]'”. This means that the integer variable cannot be initialized to hold a string of characters. The second error I introduced was removing a semicolon at the end of the statement “cout << "How many UK citizens were surveyed?"” in the source code. This gave me the error of “Expected ';' after expression,” meaning that the compiler needs a semicolon after that statement to know that that statement is finished and that it can move on to the next new statement. Both errors resulted in the program failing to compile.