Computer Science 31 Kitu Komya

Project 1 UID: 404-491-375

**original.cpp**

I did not modify the original.cpp file at all. What I will comment on instead are the results of some of its outputs I tested. In my first case, for all of the prompts, I entered in 0. Clearly that means that neither of the candidates won and that 0% of the votes were received on my end. However, the output reads that I have received -nan(ind)% of the votes. Upon closer examination of the code, we see that we have made the computer divide by zero, and since the code does not handle such a test case, our output is messed up. The output also says that I have won when clearly I have tied with my opponent.

I then entered in 5 for all of the values, hoping to get an actual percentage. This time the output read 100% of the votes needed to win have been received which is correct. However, like the past test case, the output again declares me the winner when I have only tied with my opponent. Thus it seems that our code defaults to me winning when there is a tie.

Next I tried a case where I received 10 electoral college votes, my opponent received 15, and inputted that 30 votes were needed to win. Surprisingly, the computer once more declared me as the winner (I promise I didn’t bribe the computer!) instead of saying that neither of us are the winner. I therefore have concluded that the code is not robust and does not handle all types of test cases.

I finally tried a case where I received 10 electoral college votes, my opponent received 20, and inputted that 15 were needed to win. I still win, when I am clearly below the cut-line while the opponent is not. I have accidentally mesmerized my computer, it seems, because no matter how I try, I always remain the winner. I have no complains, however. :)

**compile\_error.cpp**

In this, I have changed the case in “if” in the if statement to “If” as well as remove the last bracket “}” at the end of the code. Since C++ is case-sensitive and since it needs to close each open bracket, the code will not compile and therefore not run but rather give lots of error messages (not warnings, because it will not be able to compile and run).

**logic\_error.cpp**

I changed the declaration of pctYou from 100.0 \* votesReceived / votesToWin to simply 2. So now, although the code compiles and run, the output will always read that I have only received 2% of the votes, which will most likely be wrong for most test cases.