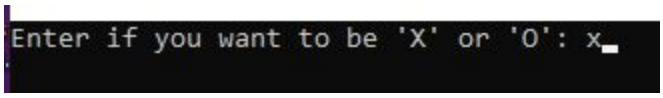
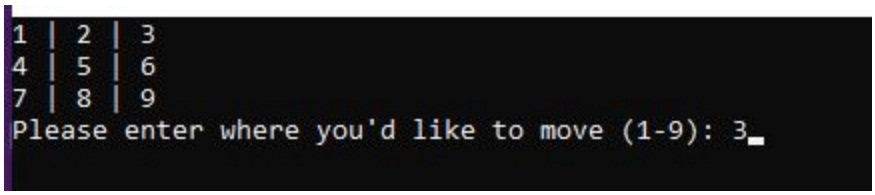
# Tic-Tac-Toe (C++)

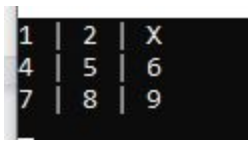
Your final CS1B Project is to write a program that enables the user to play *Tic-Tac-Toe* with the computer. On paper this sounds simple, as you might be thinking you can just make the computer move to a random spot every time. However, your job is to make it so the computer will be "smart" and actually be a worthy opponent. This means that it must be able to stop you from winning if it sees you're about to, or if it sees it is about to win then it will make the necessary move to achieve victory, and it can't move somewhere that's already occupied. You are free to decide how you want to implement this program, and you can use anything we've learned from CS 1A or CS 1B. Once you are done, you will demo it in front of me and show me what you've accomplished.

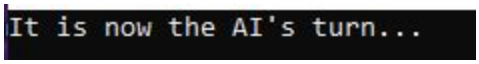
***This will be due on 07/19/2020 by 11:59 PM***

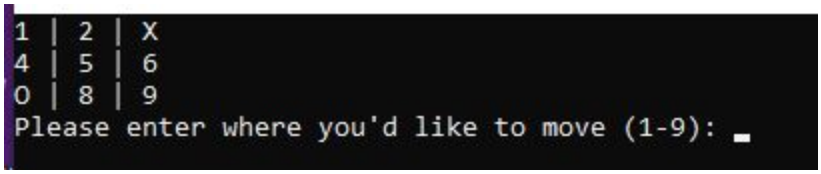
Here is a run-through of the code: (**PLEASE NOTE THAT WHENEVER THERE IS A NEW SCREENSHOT THAT MEANS THE TERMINAL SCREEN CLEARED)**



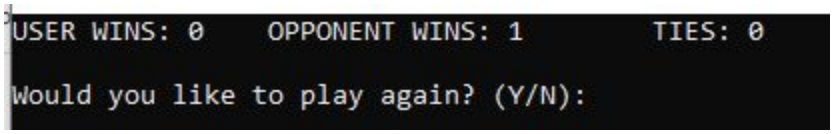






**CONTINUE THE GAME…**


**(I lose)**




You will be responsible for keeping track of how many wins, losses, and ties there are. And you will prompt the user to play again as well. Additionally, you will store the "HIGH SCORE" in a .bin file which will consist of their win ratio. (wins divided by total matches). If they beat their former high score, you will alert them. If they don't, the program proceeds as normal.

This program should be error-proof, meaning that if I type in "RUHAUFHAUFH38FH9484SGH" it'll prompt me for input again. Or if I try to move to a spot that's already occupied, it'll handle that as well, or if I try to move to an invalid spot like I input (25), it'll deal with that as well.

Lastly, you will need to randomize if the user or the AI starts the game. Meaning that some matches you will make the first move, and others the AI will make the first move. This is so you can't exploit the program and start at a good spot every time. You'll also be required to ask the user if they want to be 'X' or 'O', and whatever piece they pick, the AI picks the opposite.

**I will run the program several times in front of you so you can fully see how it should operate, as these screenshots can be difficult to follow.**