You are to write an Internet game program that will allow a user to play a game of Connect Four over the Internet with another user.

Connect Four is played on vertical grid of six rows and seven columns. By “vertical,” I mean that the grid stands upright on a tabletop; it does not lie flat. There are two players, red and blue. Red goes first. Each player takes turns dropping chips of his/her color into one of the seven slots along the top. Gravity causes the chip to drop to the lowest unfilled position. The winner is the first player to have four of his or her own color in a row, horizontally, vertically, or diagonally.

You are to write a server program and a client program. The server just sits there and waits for clients to connect. The server never prints anything to the screen and never reads anything from the keyboard. The client is what the players interact with. At any given time, there will be two copies of the client program running simultaneously, one for each player. Of course, these two client programs could be operating at opposite ends of the globe! The server should take no command-line arguments. The client should only take one: the host on which the server is running.

The first player that connects to the server is Red, and this player should be informed of this. The second player that connects is Blue, and this player should be informed of this. After Blue connects, Red should be prompted to make its move. The move MUST be a single digit in the range 1-7; your program should flag an error otherwise. Additionally, your program MUST check that a chip can legally be dropped into that column. If the input fails for any reason, your program MUST nag the user until s/he makes a valid move. Each player should receive a copy of the board BEFORE and AFTER s/he makes his/her move.

The program should detect when the game is over. Each player should be informed whether s/he won or lost or whether the game was a draw. The client programs should then terminate (not crash). The server program should NOT terminate. It should wait for two new players. The server should never terminate unless it is explicitly killed.

Some things to keep in mind:

1. This is a month-long major project. This program can be written by the due date, even understanding that Spring Break is your vacation. You probably will not be able to write the whole thing the night before, though. Work on it continuously over the period.
2. How the clients communicate with the server is up to you. I don't care who does the bulk of the work. Does the server do all the work and the clients just print messages? Do the clients do all the work and the server just passes messages back and forth? Is the work shared in some way between the three running programs? This is all up to you. You may want to plan strategy BEFORE writing the code. What would be the easiest way to do it?

3. Even though two clients are running, there is only one client source code. You can't have a RedClient and a BlueClient. The client program will somehow have to figure out whether it is Red or Blue. (I'd recommend having the server tell it.)

4. Break up the problem into tasks. Don't try to do it in one sitting. You might want to see if you can program the game without using the Internet first, and then adding the Internet calls later. Or maybe you want to set up some crude communication first and play an approximate game before really coding the rules. How you break it up is up to you.

Have fun!