// Author: Patrick Laverty

// - Server communication

// - Based on code from geeksforgeeks.com

// Will handle up to 3 clients

/*

Responsibilities: Server-client architecture including handling of user ersponse, game processes and instructions with Client

Completed: Unused early iteration server code below that was referenced for final server

Earlier iteration server-client architecture

Other: Synchronization and debugging

This is just a basic setup for server communication and setup.

Also need to complete scoreboard.....

#include "Main.h"

void server()

{

int sockfd, newsockfd, port_num;

socklen_t clilen;

struct sockaddr_in serv_addr, cli_addr;

char buffer[MAX];

int n;

// Socket creation, port number

sockfd = socket(AF_INET, SOCK_STREAM, 0);

```
bzero((char*)&serv_addr, sizeof(serv_addr));
port_num = atoi(PORT);

// Assigning IP, PORT
serv_addr.sin_family = AF_INET;
ser_addr.sin_addr.s_addr = inet_addr("127.0.0.1");
serv_addr.sin_port = htons(port_num);

// Binds socket
bind(sockfd, (struct sockaddr*)&serv_addr, sizeof(serv_addr));
clilen = sizeof(cli_addr);

int pid;
while (1)
{
	new_sockfd = accept(sockfd, (struct sockaddr*)&cli_addr, &clilen);

	pid = fork();

	// Child process, will contain game logic?
	if (pid == 0) {
		close(sockfd);
		n = read(new_sockfd, buffer, MAX-1);
		if (n < 0)
		{
			printf("Error reading from socket.");
		}
		printf("Message: %s\n", buffer);
		close(new_sockfd);
```

```
                }
                // Parent process for what?
                if (pid > 0)
                {
                        close(new_sockfd);
                }
        }
}

// Game starts
if(playerTurn(newSocket) == 0)
{
        // SCOREBOARD METHOD HERE, MAKE SCOREBOARD METHOD AND PUT IT ABOVE
        // singlePlayerScoreboard();
        // NEEDS TO SEND CLIENT SCORE OF PLAYER AND COMPUTER
        // NEEDS TO LET CLIENT KNOW IF THEY WERE ADDED TO SINGLE PLAYER SCOREBOARD FILE
        // IF PLAYER HAS HIGHER SCORE THAN COMPUTER ADD THEM TO SINGLE PLAYER SCOREBOARD
FILE

}



int displaySingleplayerScoreboard() {
        welp
}
*/
```