CSC209H Worksheet: Select: `netcat`

On this worksheet, you will write a simple version of a client called `netcat`. `netcat` must listen to both `stdin` (for input from the user) and the socket, so it needs `select`.

Your program should take two command-line arguments: the IP address and port of the machine you want to connect to. It should open a socket and connect to the given IP address at the given port, and then simply echo the data back and forth. Whenever the user types something on the keyboard, the program sends this to the socket, and whenever information arrives on the socket, the program echoes this data to the user on standard output. You may assume each time you read data that it fits into a buffer of size `MAXSIZE`.

```
int main(int argc, char **argv) {
    char buf[MAXSIZE];

    // create the socket








    // set up the sockaddr_in struct for connecting
    struct sockaddr_in peer;
    peer.sin_family = AF_INET;


    peer.sin_port =


    if (inet_pton(                                                      ) < 0) {

        perror("inet_pton");
        close(soc);
        exit(1);
    }


    // connect the socket







    // anything else to do before the infinite loop (on next page)
    while (1) {
        // use an fd_set and select to wait for keyboard or server input



        // If there is data from stdin, send to server



        // If there is data from the server, print to stdout



        // Any other cleanup/setup, if necessary



    }
    return 0;
}
```