Purpose: This assignment is to gain an in-depth understanding of a TCP client-server program.

You will build a connection-oriented time client and an iterative connection-oriented time server using the techniques found in Chapter 10 of the text. (You will use TCP.)

Submission: You must submit a print out of both the client and the server. In addition you must place the source code for both the client and the server in your home directory in files called `Ttimec.c` and `Ttimed.c`, respectively. I will access these files to check the correctness of your programs.

Port numbers: As in the previous assignment, you must use your personal well-known-port number from the `get_port` procedure.

THE TIME SERVER

Your time server will listen on your well known TCP port. Whenever it accepts a connect from a client it will call `gettimeofday` and return the `tv_sec` and `tv_usec` values to the client. You are returning 8 bytes; you will need an 8 byte array.

Sequence:

1) Get a request.
2) Use the `gettimeofday` call.
3) Print (to the screen) the values of `tv_sec` and `tv_usec` in hexadecimal.
4) htonl the `tv_sec` and `tv_usec`, go ahead and put them right back into the structure.
5) Use two calls to `memcpy`, to copy the sec into the array starting at position 0 and the usec into the array starting at position 4.
6) Send the array (8 bytes).

THE TIME CLIENT

Your time client will connect to server and get the time on the server's machine. The client will connect (it will NOT send a message) and receive as a reply the time on the server's machine given as the two `gettimeofday` numbers (`tv_sec` and `tv_usec`).

You are receiving 8 bytes; you will need an 8 byte array.

You are receiving a time of day you are required to use a `timeval` structure to contain the final answer.

Sequence:

1) Connect to the server.
2) Read and reassemble the reply.
3) Use two calls to `memcpy` to copy the sec and usec into the (`tv_sec` and `tv_usec`) fields of a `timeval` structure.
4) ntohl the `tv_sec` and `tv_usec`, go ahead and but them right back into the structure.
5) Print (to the screen) the values of `tv_sec` and `tv_usec` in hexadecimal.

Recommendations: (not requirements)

Server) This is a merge of the TCP server from Chapter 10 and your time server from the previous homework. A good starting point is Comer's code, copy in the time, htonl and memcpy from your previous homework.

Client) This is a merge of your UDP client from your homework and your TCP reassembly loop from the TCP client homework. Start with your TCP client from your homework, copy in the memcpy and htonl stuff after the reassembly loop. Remember, it's 8 bytes this time so change your array size and loop constant.