Learning Outcome: Students will gain experience with Windows programming in C.

In lab 8, your task is to modify/improve the echo server and echo client program you wrote in class as follows.

- Server
  - Prints "Connecting Client <ID>".
  - Sends a welcome message: "Client <ID>, Thank you for connecting!", where ID is the client's identification.
  - The server assigns a unique ID to each client. The first client gets 0, and the second gets 1, and so on.

- Client
  - Prompts the user to enter a word and sends it.

- Server
  - Upon receiving a message, the server generates a random number `N` between 2 and 10, inclusive. The server sends the message back to the client N times separately.
  - Do your research on the `rand()` function in C.
  - Design a protocol regarding how the server and client communicate, so the client knows whether more messages will be coming from the server.

- Whenever the client receives a message (except for some control data that you might use), it should print the message to standard input for the user.

- Your client program must allow the user to enter data until the user enters "**quit**". Then the client should disconnect, and the server should display "Disconnecting Client <ID>" and disconnect.

- Your server program must be able to handle multiple client connections concurrently.
  - You will need to use the **CreateThread** function. Refer to https://docs.microsoft.com/en-us/windows/win32/api/processthreadsapi/nf-processthreadsapi-createthread

- The server program must have a function named **handleClient** that is to be executed by threads. You will need to determine the return type and the parameters for the function.

Deliverables:

a. Create a directory named **Lab8** under the **Week8** repository.
b. Place your source files, **Lab8Client.cpp** and **Lab8Server.cpp**, under the **Lab8** directory and upload it to GitHub before 6 PM, 3/31/2021.

NAME: must be written by hand prior any sign-offs being given.

Sign offs – Each signature is worth 1/N of your lab grade, where N is the number of signatures

- The student was able to write a basic server and client program, including socket creation, send/recv, proper error handling (if any operation fails), and closing connections.

- The student was able to use the **CreateThread** and **handleClient** functions to implement a multi-threaded server.

- The student could write a client program that interacts with the user and sends/receives messages.

- The student was able to design a simple protocol so that the server and the client send/receive messages a random number of times.