Make sure the server and client files have same port number specified before compiling.
This can be modified at the field “MYPORTNUM” inside the source files.

To run client, execute following commands:

$ gcc wordlengthclient.c -lm -o client
$ ./client

To run server, execute the following commands:

$ gcc wordlengthserver.c -lm -o server
$ ./server

Description:

The server program runs continuously, listening for incoming connections. The client program once executed binds to the server. The client is now able to send messages to the server which then hashes it out and sends it back to the client.

The program uses the widely supported TCP protocol which has features that makes sure the recipient receives the message in whole in the unfortunate event of a packet loss. Its compatibility makes it useful in everyday networks and requires minimal configuration for it to work in network environments.

The testing was done on the lab computers in the Math Sciences building, either by physically being present there or through remotely connecting through SSH protocol and executed the files while updating source code on my local computer through a VPN connection. The program is able to encode incoming messages to the server by receiving, parsing it into a dictionary like data structure and sending the hash values back to the client. Unfortunately, after days of working on the assignment, the program is not able to keep state and the dictionary is renewed every new send. The algorithm works only for seq, and partly my own implementation, during tests done in a separate source file, without the network interface overhead.