**Assignment 5: Implement a Telnet Client using C Sockets**

Objective:

Create a telnet client program in C that utilizes sockets to establish a connection with a remote server. The program should allow users to execute commands on the server and receive output in real-time.

Requirements:

1. Create a socket using the socket() function.

2. Set the server address using the struct sockaddr\_in structure.

3. Establish a connection to the server using the connect() function.

4. Implement a loop to continuously read user input from the console.

5. Send the user input to the server using the send() function.

6. Receive output from the server using the recv() function.

7. Print the received output to the console.

8. Handle the "quit" command to terminate the connection.

Note:

- Use the C standard library and socket API only.

- Ensure the program is well-structured, readable, and includes proper error handling.

Submission:

Submit your C source code file (.c) and a brief report (PDF or Word document) explaining your implementation approach and any challenges faced.

Assessment:

Your code will be evaluated by attempting to establish telnet connections to five distinct remote servers that can be accessed using a standard telnet client. If your code successfully connects to all five servers, you will receive 5 marks.

Following a successful connection, we will execute a set of five commands on the remote servers using your code. These commands will be identical to those executed using a standard telnet client. If your code successfully executes all five commands, you will receive an additional 5 marks.

In total, you can earn up to 10 marks by demonstrating a fully functional telnet client that can establish connections and execute commands on remote servers.