------Input Files--------

1.User.txt (Username and Password)

------Functions in the Pseudocode------

A. User

1. User
2. ChangeUserDir

B.Server :-

1. createServer
2. LoadUsersData
3. AcceptConnections
4. ReceiveDataFromClient
5. SendDataToClient
6. AuthenticateUser
7. ListDirContents
8. ChangeDir
9. EditLine
10. ViewFile
11. SelectFile

C.Client :-

1. CreateSocket
2. ConnectToServer
3. AuthenticateUser
4. SendDataToServer
5. ReceiveDataFromServer
6. DisconnectClient
7. EditLine
8. ReceiveFile

-------Pseudocode---------

A. User

-----START-----

1. User
    a. Create a new user structure and update its values.
    b. Copy name to user->name.
    c. Copy password to user->password.
    d. Make a new dir string with default data as ./data/home.
    e. Append user->name to dir
    f. copy dir to user->dir.

2. ChangeUserDir
    a. If dir is equal to “”
    b. Change user->dir to default string “./data/home”
    c. Append user->name to user->dir.
    d. Else
    e. Copy dir to user->dir.
    f. End if

------END------

B. Server:

-----START-----

1. createServer:
    a. Create Socket
    b. Intialize Server Address
    c. Bind Socket to Server Address
    d. Listen for connection

2. Load User Data
    a. Load data in user.txt and store data in users array of server structure
3. Accept Connection from new Client
    a. Send confirmation message to client

4. Send Data to client (Connected to Server)

5. Receive Data From Client
    a. Store the data from client in a data variable
    b. Using strtok split and store in command, name and password

6. User
    a. Initialize name, password and dir of current user object

7. Authenticate User
    a. Check whether user is present using n variable of server structure
    b. Compare current username and password with users array of server structure
    c. If same
        send message AUTHENTICATED to client
    d. Else
        send message NOT AUTHENTICATED to client

8. ListDirContents
    a. Empty the data variable and store current user directory in dir variable

9. ChangeDir
    a. In data variable receive data from client and using strtok split and store commands in command variable.

10. Compare the value of command with ls,pwd,cd, select, print, edit, bye.
    If the command is ls
        List Dir Contents
        Open directory
            If null:
                Send NO FILES FOUND to client
            Else:
                Read directory
                Compare d_type value and classify it as Directory or file
                Add d_name along with d for directory and - for files to buffer
                If buffer is empty:
                    Buffer value will be empty directory
                Send buffer content to client
        Close Directory
    If the command is pwd
            Send current user dir stored in dir variable to client
    If the command cd
            Change Dir
            Using strtok split and store the directory name as new directory
            Open current user directory
              If opendir return Null
                Send DIRECTORY NOT FOUND to client
                Compare the value of d_type and d_name with new directory
                    Store current user directory/new directory in temp
                    Change user directory to temp
                    Send message directory changed to client
                    Close directory
                Compare new directory value with “..”
                    Store ./data/home and current username in temp
                    If current user directory and temp is same
                        Send DIRECTORY RESTRICTED to client
                        Close directory
                    Else
                        Traverse through the current user directory and
                        Replace last / with \0
                        Send DIRECTORY_CHANGED to client
                        Close directory
    If command is select
            Select File
            Store the filename using strtok
            Open dir
                If null
                  Send FILE_NOT_FOUND to client
                Else
                    Compare the d_type value and d_name with filename
                    Create a variable new and store dir-name/filename
                    Send FILE_SELECTED to client

If commond is print
Using strtok store start_line and end_line
ViewFile
Open file
If null
Send FILE NOT FOUND to client
Else
Count the line number
Compare end_line and start_line with line_number
If invalid then send INVALID LINE NUMBER to client
Else
Make a char array line with number and store the
File contents with numbering.
Then begin i from start_line to end_line
And Send data to client.
Close file
If command is edit
Using strtok store line number to be edited
Edit line
Open file
If null send FILE NOT FOUND to client
Else
Store contents of file in char array lines
Count the spaces if present in the line
Receive the edited line from the client
Replace that line in the char array lines
Using fputs function store the content in the file
Close File
If command is bye
Close the connection
If command is invalid
Send data to client INVALID COMMAND

------END------

C.Client :

------START------

1. CreateSocket
    a. Create Socket
    b. Intialize Server Address
    c. Bind Socket to Server Address
    d. Listen for connection

2. ConnectToServer
    a. Connect to server using the created socket.
    b. Change isConnected to 1.
    c. Return error if not connected.

3. Receive Data from Server (Connection Confirmation)
4. Accept Username and password

5. AuthenticateUser
    a. Check is the client is connected to server or not
    b. If connected create a string with username and password
    c. Send the string to server
    d. Receive message AUTHENTICATED and return 1 or NOT AUTHENTICATED and return 0.

6. Accept Commands and calculate arguments

7. SendDataToServer
    a. Send the buffer to server using send function
    b. Return 0 if successful else -1.

8. If command is edit - EditLine
    a. Receive buffer from server.
    b. If buffer is 0 print “file not selected” and return -1.
    c. Else if line number is invalid print “INVALID_LINE_NUMBER” and return -1.
    d. Else print the buffer.
    e. Input the user data into buffer and send to server. If no input is given send 0.
    f. Return 0

9. If command is print - ReceiveFile
    a. If file is not at the end change isNotEnd to 1.
    b. while(isNotEnd) do
    c. Receive buffer from server
    d. If buffer is 0
    e. Print “file not selected” and return -1
    f. End if
    g. Update isNotEnd = length of buffer
    h. Print buffer
    i. End while
    j. Reutrn 0

10. ReceiveDataFromServer
    a. Receive data in the buffer from server using recv function.
    b. Return buffer if successful else EXIT_FAILURE.

11. If command is bye - DisconnectClient
    a. Send bye command to server.
    b. Close the socket connection.

c. Change isConnected to 0.
d. Return 0

-----END-----