Practical 12

AIM:

1. Implement chat client server using TCP/UDP sockets. ALGORITHM:

**Chat Server**

# Start the Server:

* + - Create a socket (like a phone line).
    - Bind it to a specific address and port (set your phone number).
    - Listen for incoming connections (wait for calls).

# Accept Connections:

* + - When a new client connects:
      * Add the client to a list of connected clients.
      * Start a new process to talk to this client (like talking on a different line).

# Receive Messages:

* + - For each connected client:
      * Keep checking for new messages.
      * When a message is received, show it on the server screen.
      * Send this message to all other clients (share the message with everyone).

# Handle Disconnections:

* + - If a client disconnects:
      * Remove that client from the list.
      * Stop talking to that client.

# Keep Running:

* + - Repeat the process until you stop the server.

**Chat Client**

# Connect to the Server:

* + Create a socket (your phone).
  + Connect it to the server’s address and port (call the server).

# Receive Messages:

* + Start a process to listen for messages from the server:
    - Whenever a message arrives, show it on your screen.

# Send Messages:

* + Keep asking the user for new messages:
    - If the user types a message, send it to the server.
    - If the user types 'exit', disconnect from the server and close the client.

# Keep Running:

* + Repeat the process until the user decides to exit.

OUTPUT:

