```
udpserver.c
=================
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <arpa/inet.h>

#define PORT 12345
#define BUFFER_SIZE 1024

int main() {
    int sockfd;
    char buffer[BUFFER_SIZE];
    struct sockaddr_in server_addr, client_addr;
    socklen_t addr_len = sizeof(client_addr);
    ssize_t recv_len;

    // Create UDP socket
    if ((sockfd = socket(AF_INET, SOCK_DGRAM, 0)) < 0) {
        perror("socket creation failed");
        exit(EXIT_FAILURE);
    }

    // Set up the server address struct
    memset(&server_addr, 0, sizeof(server_addr));
    server_addr.sin_family = AF_INET;
    server_addr.sin_addr.s_addr = INADDR_ANY;
    server_addr.sin_port = htons(PORT);

    // Bind the socket to the server address
    if (bind(sockfd, (struct sockaddr *)&server_addr, sizeof(server_addr)) < 0) {
        perror("bind failed");
        close(sockfd);
        exit(EXIT_FAILURE);
    }

    printf("UDP server listening on port %d\n", PORT);

    // Receive and print messages from clients
    while (1) {
        recv_len = recvfrom(sockfd, buffer, BUFFER_SIZE - 1, 0,
                            (struct sockaddr *)&client_addr, &addr_len);
        if (recv_len < 0) {
            perror("recvfrom failed");
            close(sockfd);
            exit(EXIT_FAILURE);
        }

        buffer[recv_len] = '\0';
        printf("Received message: %s\n", buffer);
    }

    close(sockfd);
    return 0;
}
```

```
udpclient.c
==============
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <arpa/inet.h>

#define PORT 12345
#define BUFFER_SIZE 1024

int main() {
    int sockfd;
    struct sockaddr_in server_addr;
    char buffer[BUFFER_SIZE];

    // Create UDP socket
    if ((sockfd = socket(AF_INET, SOCK_DGRAM, 0)) < 0) {
        perror("socket creation failed");
        exit(EXIT_FAILURE);
    }

    // Set up the server address struct
    memset(&server_addr, 0, sizeof(server_addr));
    server_addr.sin_family = AF_INET;
    server_addr.sin_addr.s_addr = inet_addr("127.0.0.1");  // Replace with server IP if different
    server_addr.sin_port = htons(PORT);

    printf("Enter message to send to server: ");
    fgets(buffer, BUFFER_SIZE, stdin);

    // Send message to server
    if (sendto(sockfd, buffer, strlen(buffer), 0,
                (struct sockaddr *)&server_addr, sizeof(server_addr)) < 0) {
        perror("sendto failed");
        close(sockfd);
        exit(EXIT_FAILURE);
    }

    printf("Message sent to server: %s", buffer);

    close(sockfd);
    return 0;
}
```