```c
#include <stdio.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <string.h>

int main() {
    int clientSocket;
    char buffer[1024];
    char buf[1024];
    struct sockaddr_in serverAddr;
    socklen_t addr_size;
    clientSocket = socket(AF_INET, SOCK_DGRAM, 0);
    if (clientSocket < 0) {
        perror("socket creation failed");
        return 1;
    }
    serverAddr.sin_family = AF_INET;
    serverAddr.sin_port = htons(2000);
    serverAddr.sin_addr.s_addr = inet_addr("127.0.0.1");
    memset(serverAddr.sin_zero, '\0', sizeof serverAddr.sin_zero);
    addr_size = sizeof serverAddr;

    while (1) {
        printf("Enter the message: ");
        fgets(buf, 1024, stdin);
        printf("Message sent to server\n");

        strcpy(buffer, buf);
        sendto(clientSocket, buffer, strlen(buffer), 0, (struct sockaddr *)&serverAddr,
addr_size);

        int recv_size = recvfrom(clientSocket, buffer, 1024, 0, NULL, NULL);
        if (recv_size < 0) {
            perror("recvfrom failed");
            return 1;
        }
        buffer[recv_size] = '\0';// Null-terminate the received data
        printf("Reply from server: %s\n", buffer);
    }
    return 0;
}
```

```c
#include <stdio.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <string.h>

int main(){
        int welcomeSocket;
        char buffer[1024];
        char buf[1024];
        struct sockaddr_in serverAddr, clientAddr;
        socklen_t addr_size;
        welcomeSocket = socket(AF_INET, SOCK_DGRAM, 0);
        if (welcomeSocket < 0){
                perror("socket creation failed");
                return 1;
        }
        serverAddr.sin_family = AF_INET;
        serverAddr.sin_family = AF_INET;
        serverAddr.sin_port = htons(2000);
        serverAddr.sin_addr.s_addr = inet_addr("127.0.0.1");
        memset(serverAddr.sin_zero,'\0',sizeof serverAddr.sin_zero);

        if (bind(welcomeSocket, (struct sockaddr *)&serverAddr, sizeof(serverAddr)) < 0){
                perror("bind failed");
                return 1;
        }

        printf("Listening...\n");
        addr_size = sizeof(clientAddr);

        while (1){
        int recv_size = recvfrom(welcomeSocket, buffer, 1024, 0, (struct sockaddr *)&clientAddr,
&addr_size);
        if (recv_size < 0) {
            perror("recvfrom failed");
            return 1;
        }
        buffer[recv_size] = '\0'; // Null-terminate the received data
        printf("Message from client: %s\n", buffer);

        printf("Enter the message: ");
        fgets(buf, 1024, stdin);
        printf("Message sent to client\n");
        sendto(welcomeSocket, buf, strlen(buf), 0, (struct sockaddr *)&clientAddr, addr_size);
    }
    return 0;
}
```

```
aadarsh@aadarsh-VirtualBox:~/CN/Exp3.2$ gcc client_udp.c -o client_udp.out
aadarsh@aadarsh-VirtualBox:~/CN/Exp3.2$ ./client_udp.out
Enter the message: Hey Server
Message sent to server
Reply from server: Ye ye client

Enter the message: au revoir
Message sent to server
Reply from server: peace out

Enter the message:
```

```
aadarsh@aadarsh-VirtualBox:~/CN/Exp3.2$ gcc server_udp.c -o server_udp.out
aadarsh@aadarsh-VirtualBox:~/CN/Exp3.2$ ./server_udp.out
Listening...
Message from client: Hey Server

Enter the message: Ye ye client
Message sent to client
Message from client: au revoir

Enter the message: peace out
Message sent to client
```