```
udpserver.c
    ===========
   #include <stdio.h>
    #include <stdlib.h>
    #include <string.h>
 6
    #include <unistd.h>
 7
    #include <arpa/inet.h>
8
9
    #define PORT 12345
10
    #define BUFFER SIZE 1024
11
12
   int main() {
13
         int sockfd;
14
         char buffer[BUFFER SIZE];
15
         struct sockaddr in server addr, client addr;
         socklen t addr len = sizeof(client addr);
16
17
         ssize_t recv_len;
18
19
         // Create UDP socket
20
         if ((sockfd = socket(AF INET, SOCK DGRAM, 0)) < 0) {
21
             perror("socket creation failed");
22
             exit(EXIT FAILURE);
23
         }
24
25
         // Set up the server address struct
26
        memset(&server addr, 0, sizeof(server addr));
27
         server_addr.sin_family = AF_INET;
28
         server_addr.sin_addr.s_addr = INADDR_ANY;
29
        server_addr.sin_port = htons(PORT);
30
         // Bind the socket to the server address
31
32
         if (bind(sockfd, (struct sockaddr *)&server addr, sizeof(server addr)) < 0) {
33
             perror("bind failed");
34
             close(sockfd);
35
             exit(EXIT_FAILURE);
36
         }
37
38
         printf("UDP server listening on port %d\n", PORT);
39
40
         // Receive and print messages from clients
41
         while (1) {
42
             recv len = recvfrom(sockfd, buffer, BUFFER SIZE - 1, 0,
43
                                 (struct sockaddr *) &client addr, &addr len);
44
             if (recv len < 0) {
45
                 perror("recvfrom failed");
46
                 close(sockfd);
47
                 exit(EXIT FAILURE);
48
             }
49
50
             buffer[recv_len] = '\0';
51
             printf("Received message: %s\n", buffer);
52
         }
53
54
        close(sockfd);
55
         return 0;
56
   }
57
58
   udpclient.c
59
    =========
60
    #include <stdio.h>
61
    #include <stdlib.h>
62
    #include <string.h>
63 #include <unistd.h>
64 #include <arpa/inet.h>
65
66 #define PORT 12345
4define BUFFER SIZE 1024
68
69
    int main() {
```

```
int sockfd;
 71
         struct sockaddr_in server_addr;
 72
         char buffer[BUFFER SIZE];
 73
 74
         // Create UDP socket
 75
         if ((sockfd = socket(AF_INET, SOCK_DGRAM, 0)) < 0) {</pre>
              perror("socket creation failed");
 76
 77
              exit(EXIT FAILURE);
 78
 79
 80
         // Set up the server address struct
         memset(&server addr, 0, sizeof(server addr));
 81
 82
          server addr.sin family = AF INET;
 83
          server addr.sin addr.s addr = inet addr("127.0.0.1"); // Replace with server IP if
          different
 84
          server addr.sin port = htons(PORT);
 85
 86
         printf("Enter message to send to server: ");
 87
         fgets(buffer, BUFFER SIZE, stdin);
 88
 89
         // Send message to server
 90
         if (sendto(sockfd, buffer, strlen(buffer), 0,
 91
                     (struct sockaddr *)&server addr, sizeof(server addr)) < 0) {</pre>
 92
              perror("sendto failed");
 93
              close(sockfd);
 94
              exit(EXIT FAILURE);
 95
          }
 96
 97
         printf("Message sent to server: %s", buffer);
 98
99
         close(sockfd);
100
         return 0;
101
     }
102
103
```