Client:

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <pthread.h>
#define PORT 8445
#define BUFFER_SIZE 1024

void *receiveMessage(void *socket) {
   int sock = *(int *)socket;
   char message[BUFFER_SIZE];
   while(1) {
      int len = recv(sock, message,
BUFFER_SIZE - 1, 0);
      if(len > 0) {
         message[len] = '\0';
         printf("%s", message);
      } else {
         printf("Server disconnected or
recv error\n");
         exit(0); }}
   return NULL; }

int main() {
   int sock = socket(AF_INET,
SOCK_STREAM, 0);
   struct sockaddr_in server = {
      .sin_addr.s_addr = inet_addr("127.0.0.1"),
      .sin_family = AF_INET,
      .sin_port = htons(PORT),};

   if(connect(sock, (struct sockaddr *)&server,
sizeof(server)) < 0) {
      perror("connect failed");
      return 1;}
printf("Connected\n");
pthread_t recv_thread;
pthread_create(&recv_thread, NULL,
receiveMessage, &sock);
char message[BUFFER_SIZE];
   while(fgets(message, BUFFER_SIZE,
stdin) != NULL) {
      if(send(sock, message,
strlen(message), 0) < 0) {
         perror("send failed");
         break; // Exit or handle error}}
 close(sock);
 return 0;}
```

Server:

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <netinet/in.h>
#include <pthread.h>

#define PORT 8445
#define MAX_CLIENTS 10
#define BUFFER_SIZE 1024

int clients[MAX_CLIENTS];
int client_count = 0;
void broadcastMessage(char *msg, int sender) {
   for (int i = 0; i < client_count; i++) {
      if (clients[i] != sender) {
         send(clients[i], msg, strlen(msg), 0);} }}

void *clientHandler(void *socket_desc) {
 int sock = *(int *)socket_desc;
char buffer[BUFFER_SIZE];
 while (1) {
int read_size = recv(sock, buffer, BUFFER_SIZE
- 1, 0);
if (read_size <= 0) break;
buffer[read_size] = '\0';
broadcastMessage(buffer, sock);}
close(sock);
return NULL;}
int main() {
   int socket_desc, new_socket;
   struct sockaddr_in server, client;
   socket_desc = socket(AF_INET,
SOCK_STREAM, 0);
   if (socket_desc == -1) {
      perror("Could not create socket");
      return 1;}
server.sin_family = AF_INET;
server.sin_addr.s_addr = INADDR_ANY;
server.sin_port = htons(PORT);
if (bind(socket_desc, (struct sockaddr *)&server,
sizeof(server)) < 0) {
  perror("bind failed");
return 1;}
 listen(socket_desc, MAX_CLIENTS);
 puts("Waiting for incoming connections...");
while ((new_socket = accept(socket_desc, NULL,
NULL))) {
if (new_socket < 0) {
perror("accept failed");
```

```
continue;}
puts("Connection accepted");
if (client_count < MAX_CLIENTS) {
clients[client_count++] = new_socket;
pthread_t thread_id;
if (pthread_create(&thread_id, NULL,
clientHandler, (void *)&new_socket) != 0) {
perror("could not create thread");}
} else {char *message = "Max clients reached.
Try again later\n";
write(new_socket, message, strlen(message));
close(new_socket);}}
 return 0;}
```

Output:

Server:
Waiting for incoming connections...
Connection accepted
Connection accepted

Client1:
Connected
hi
hello
hhow are you

Client2:
Connected
hi
hello
hhow are you

Output:

server.c

```
Socket successfully created..
Socket successfully binded..
Server listening..
server accept the client...
From client: hello
        To client : hi
From client: how are you
        To client : fine
```

client.c

```
Socket successfully created..
connected to the server..
Enter the string : hello
From Server : hi
Enter the string : how are you
From Server : fine
```