Chat Client :

```
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
#include<sys/types.h>
#include<sys/socket.h>
#include<unistd.h>
#include<arpa/inet.h>
#include<errno.h>
#include<fcntl.h>
void main(){

struct sockaddr_in server,client;
int sock,clientSocket;
char receivedBytes[1024],sendBytes[1024];
int bytes;
if((sock = socket(AF_INET,SOCK_STREAM,0)) == -
1){ perror("Invalid Socket Descriptor");
exit(1);
}
```

```
server.sin_family = AF_INET;
server.sin_port = htons(5005);
server.sin_addr.s_addr = INADDR_ANY;
bzero(&(server.sin_zero),8);

if(connect(sock,(struct sockaddr
*)&server,sizeof(server)) == -
1){ perror("Unable to connect");
exit(1);
}

while(1){
printf("\nClient: ");
gets(sendBytes);
bytes = send(sock,sendBytes,1024,0);
if(strcmp(sendBytes,"q")
== 0 || strcmp(sendBytes,"Q") == 0){
printf("\nClient exiting...");
close(sock);
exit(1);
}
bytes = recv(sock,receivedBytes,1024,0);
receivedBytes[bytes] = '\0';
printf("\nServer: %s",receivedBytes);
```

```
if(strcmp(receivedBytes,"q") == 0 ||
strcmp(receivedBytes,"Q") == 0){

printf("\nServer going off...");

close(sock);

break;

}

}

}
```

Chat Server :

```
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
#include<sys/types.h>
#include<sys/socket.h>
#include<unistd.h>
#include<arpa/inet.h>

#include<errno.h>
#include<fcntl.h>

void main(){

struct sockaddr_in server,client;
int sock,clientSocket;
char receivedBytes[1024],sendBytes[1024];
int bytes;
if((sock = socket(AF_INET,SOCK_STREAM,0)) == -
1){ perror("Invalid Socket Descriptor");
exit(1);
}
```

```
server.sin_family = AF_INET;
server.sin_port = htons(5005);
server.sin_addr.s_addr = INADDR_ANY;
bzero(&(server.sin_zero),8);

if(bind(sock,(struct sockaddr
*)&server,sizeof(server)) == -
1){ perror("Unable to bind");
exit(1);
}

if(listen(sock,5) == -1){
perror("Unable to listen");
exit(1);
}
printf("Server waiting for client...");

fflush(stdout);

while(1){
int len = sizeof(client);
clientSocket = accept(sock,(struct sockaddr
*)&client,&len); if(clientSocket == -1){
perror("Connection error");
```

```
exit(1);
}
printf("I recevied a connection from %s on
port
%d",inet_ntoa(client.sin_addr),ntohs(client.sin_por
t));

while(1){
bytes = recv(clientSocket,receivedBytes,1024,0);
receivedBytes[bytes] = '\0'; printf("\nClient:
%s",receivedBytes);
if(strcmp(receivedBytes,"q") == 0 ||
strcmp(receivedBytes,"Q") ==
0){ printf("\nClient want to exit...");
printf("\nWaiting for new client...");
close(clientSocket);
break;
}
printf("\nServer: ");
gets(sendBytes);
bytes = send(clientSocket,sendBytes,1024,0);
if(strcmp(sendBytes,"q") == 0 ||
strcmp(sendBytes,"Q") == 0){
printf("\nServer going off...");
close(clientSocket);
```

```
exit(1);
}}
```

Output