Tcp server:

```
#include<stdio.h>
#include<netinet/in.h>
#include<stdlib.h>
#include<string.h>
#include<sys/socket.h>
#include<netdb.h>
#include<sys/types.h>
#define MAX 80
#define PORT 8080
#define SA struct sockaddr

void func(int connfd)
{
char buff[MAX];
int n;
for(;;) {
bzero(buff, MAX);
//read the message from client and copy it in buffer
read(connfd,buff,sizeof(buff));
//print buffer which contains client contents
printf("From Client:%s\t To client: ", buff);
bzero(buff, MAX);
n=0;
//copy server message in the buffer
while((buff[n++]=getchar())!='\n')
;
//send that buffer to client
write(connfd,buff,sizeof(buff));
//if msg contains "Exit" then server exit
if(strncmp("exit",buff,4)==0) {
printf("Server Exit...\n");
break;
}
}
}
int main()
{
 int sockfd,connfd,len;
struct sockaddr_in servaddr, cli;

//socket create and verification
sockfd=socket(AF_INET,SOCK_STREAM,0);
if(sockfd==-1) {
printf("socket creation failed....\n");
exit(0);
}
else
printf("socket Successfully created..\n");
bzero(&servaddr,sizeof(servaddr));
//assign IP, PORT
servaddr.sin_family=AF_INET;
servaddr.sin_addr.s_addr=htonl(INADDR_ANY);
servaddr.sin_port=htons(PORT);
//binding newly created socket to given IP
if((bind(sockfd,(SA*)&servaddr,sizeof(servaddr)))!=0)
{ printf("socket bind failed...\n");
exit(0);
}
else
printf("socket successfully binded..\n");
// now server is ready to listen and verification
if((listen(sockfd,5))!=0)
{
printf("listen failed..\n");
exit(0);
}
else
printf("Server listening...\n");
len=sizeof(cli);
//accpet data packet from client and verify
connfd=accept(sockfd,(SA*)&cli,&len);
if(connfd<0){
printf("server accept failed..\n");
exit(0);
}
else
printf("aserver accept the client...\n");
//function for chatting between clent and server
func(connfd);
//close socjket
close(sockfd);
}
```

Tcp client:

```
#include<netdb.h>
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
#include<sys/socket.h>
#define MAX 80
#define PORT 8080
#define SA struct sockaddr
void func(int sockfd)
{
char buff[MAX];
int n;
for(;;){
bzero(buff,sizeof(buff));
printf("enter the string:");
n=0;
while((buff[n++]=getchar())!='\n')
;
write(sockfd,buff,sizeof(buff));
bzero(buff,sizeof(buff));
read(sockfd,buff,sizeof(buff));
printf("From server: %s",buff);
if((strncmp(buff,"exit",4))==0) {
```

```
printf("Client exit....\n");
break;
}
}
}
int main()
{
int sockfd,connfd;
struct sockaddr_in servaddr,cli;
//socket create and verification
sockfd=socket(AF_INET,SOCK_STREAM,0);
if(sockfd==-1){
printf("socket creation failed....\n");
exit(0);
}
else
printf("socket successfully created...\n");
bzero(&servaddr, sizeof(servaddr));
//assign IP,PORT
servaddr.sin_family=AF_INET;
servaddr.sin_addr.s_addr=inet_addr("127.0.0.1");
servaddr.sin_port=htons(PORT);

//connect the client socket to server socket
if(connect(sockfd,(SA*)&servaddr,sizeof(servaddr))!=0) {
printf("connection with server failed...\n");
exit(0);
}
else
printf("connected to server...\n");
//function for chat
func(sockfd);
//close socket
close(sockfd);
}
```

<u>Udp server:</u>

```
#include<stdio.h>
#include<stdlib.h>
#include<unistd.h>
#include<string.h>
#include<sys/types.h>
#include<sys/socket.h>
#include<arpa/inet.h>
#include<netinet/in.h>
#define PORT 8080
#define MAXLINE 1024

//driver code
int main()
{
int sockfd;
char buffer[MAXLINE];
char *hello="hello from server";
struct sockaddr_in servaddr,cliaddr;
//creating socket file descriptor
if((sockfd=socket(AF_INET,SOCK_DGRAM,0))<0) {
perror("socket creation dfailed");
exit(EXIT_FAILURE);
}
memset(&servaddr,0,sizeof(servaddr));
memset(&cliaddr,0,sizeof(cliaddr));
//filling server information
servaddr.sin_family=AF_INET;
servaddr.sin_addr.s_addr=INADDR_ANY;
servaddr.sin_port=htons(PORT);
//bind socket with server addredd
if(bind(sockfd,(const struct
sockaddr*)&servaddr,sizeof(servaddr))>0)
{
perror("bind failed");
exit(EXIT_FAILURE);
}
int len,n;
len=sizeof(cliaddr);
n=recvfrom(sockfd,(char*)buffer,MAXLINE,MSG_WAITALL,(struct
sockaddr*)&cliaddr,&len);
buffer[n]='\0';
printf("client:%s\n",buffer);
sendto(sockfd,(const char*)hello,strlen(hello),MSG_CONFIRM,(const
struct sockaddr*)&cliaddr,len);
printf("hello message sent\n");
return 0;
}
```

<u>**Udp client:**</u>

```
#include<stdio.h>
#include<stdlib.h>
#include<unistd.h>
#include<string.h>
#include<sys/types.h>
#include<sys/socket.h>
#include<arpa/inet.h>
#include<netinet/in.h>
#define PORT 8080
#define MAXLINE 1024
//Driver code
int main()
{
int sockfd;
char buffer[MAXLINE];
char *hello="Hwello from client";
struct sockaddr_in servaddr;
//creating socketfile descriptor
if((sockfd=socket(AF_INET,SOCK_DGRAM,0))<0)
{
perror("socket creation failed");
exit(EXIT_FAILURE);
}
memset(&servaddr,0,sizeof(servaddr));
//filling server information
servaddr.sin_family=AF_INET;
servaddr.sin_port=htons(PORT);
servaddr.sin_addr.s_addr=INADDR_ANY;

int n,len;
sendto(sockfd,(const char*)hello,strlen(hello),MSG_CONFIRM,(const
struct sockaddr*)&servaddr,sizeof(servaddr));
printf("hello message sent:\n");
n=recvfrom(sockfd,(char*)buffer,MAXLINE,MSG_WAITALL,(struct
sockaddr*)&servaddr,&len);
buffer[n]='\0';
printf("server:%s\n",buffer);
close(sockfd);
return 0;
}
```