Server side

```
#include"sys/socket.h"
#include"netinet/in.h"
#include"stdio.h"
#include"string.h"
#include"stdlib.h"
#include <fcntl.h>
#include <unistd.h>

int main()
{
char buf[100];
int k;
socklen_t len;
int sock_desc,temp_sock_desc;

struct sockaddr_in server,client;
sock_desc=socket(AF_INET,SOCK_STREAM,0);

if(sock_desc==-1)
{
  printf("error in socket creation");
  exit(EXIT_FAILURE);
}

server.sin_family=AF_INET;
server.sin_addr.s_addr=INADDR_ANY;
server.sin_port=3003;



k=bind(sock_desc,(struct sockaddr*)&server,sizeof(server));
if(k==-1){
 printf("error in binding");
 exit(EXIT_FAILURE);
  }

 k=listen(sock_desc,5);
 if(k==-1){
   printf("error in listening");
   exit(EXIT_FAILURE);
  }

len=sizeof(client);
temp_sock_desc=accept(sock_desc,(struct sockaddr*)&client,&len);
if(temp_sock_desc==-1){
  printf("error in temporary socket creation");
  exit(EXIT_FAILURE);
```

```
  }


 while(1)
 {
k=recv(temp_sock_desc,buf,sizeof(buf),0);
if(k==-1) {
  printf("error in receiving");
  exit(EXIT_FAILURE);
  }

  printf("message from the client is : %s",buf);

  if(strcmp(buf,"exit\n")==0){
     printf("exiting the program.\n");
     break;

  }


  printf("enter data to client(type'exit' to quit):");
  fgets(buf,sizeof(buf),stdin);
  k=send(temp_sock_desc,buf,strlen(buf),0);
  if(k==-1)
  {
  printf("error in sending");
  exit(EXIT_FAILURE);
  }
  }
  close(temp_sock_desc);
  close(sock_desc);
  return 0;
}
```

Client Side

```
#include "sys/socket.h"
#include "netinet/in.h"
#include "stdio.h"
#include "string.h"
#include <stdlib.h>
#include <fcntl.h>
#include <unistd.h>

int main()
{
      char buf[100];
```

```
      int k;
      int sock_desc;
      struct sockaddr_in server;

      sock_desc = socket(AF_INET, SOCK_STREAM, 0);
      if (sock_desc == -1) {
      perror("Error in socket creation");
      exit(EXIT_FAILURE);
      }

      server.sin_family = AF_INET;
      server.sin_addr.s_addr = INADDR_ANY;
      server.sin_port = 3003;

      k = connect(sock_desc, (struct
sockaddr*)&server,sizeof(server));
      if (k == -1) {
      perror("Error in connecting to server");
      exit(EXIT_FAILURE);
      }

      while (1) {
      printf("\nEnter data to Server (type 'exit' to quit): ");
      fgets(buf, sizeof(buf), stdin);

      // Send data to server
      k = send(sock_desc, buf, strlen(buf), 0);
      if (k == -1) {
            perror("Error in sending");
            exit(EXIT_FAILURE);
      }

      if (strcmp(buf, "exit\n") == 0) {
            printf("Exiting the program.\n");
            break;
      }

      // Receive data from server
      k = recv(sock_desc, buf, sizeof(buf), 0);
      if (k == -1) {
            perror("Error in receiving");
            exit(EXIT_FAILURE);
      }

      printf("Message from server is: %s", buf);
      }
      close(sock_desc);
      return 0;
```

}

```
ubuntu@ubuntu-H81M-S:~$ gcc tcpc.c -o client
ubuntu@ubuntu-H81M-S:~$ ./client

Enter data to Server (type 'exit' to quit): hello
Message from server is: computer

Enter data to Server (type 'exit' to quit): hiii
Message from server is: laptop
r

Enter data to Server (type 'exit' to quit): exit
Exiting the program.
```

```
ubuntu@ubuntu-H81M-S:~$ ./server
message from the client is : hello
enter data to client(type'exit' to quit):computer
message from the client is : hiii
ter
enter data to client(type'exit' to quit):laptop
message from the client is : exit
p
enter data to client(type'exit' to quit):exit
message from the client is : exit
exiting the program.
```