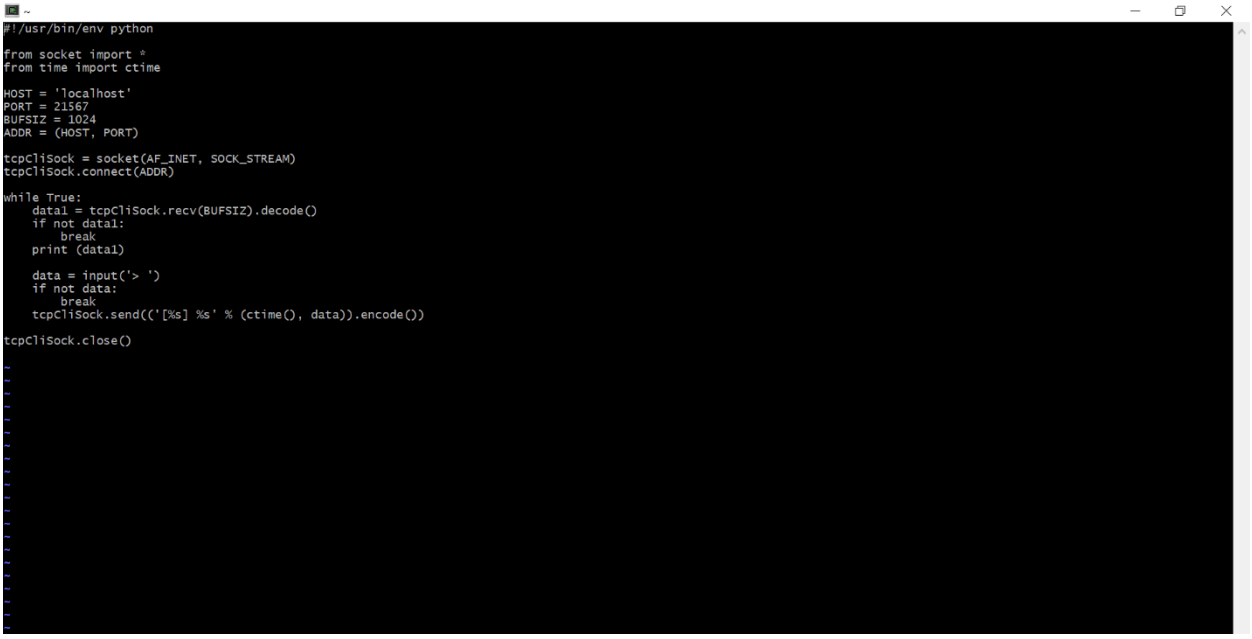
Programs:

TsClient.py



```
#!/usr/bin/env python

from socket import *
from time import ctime

HOST = 'localhost'
PORT = 21567
BUFSIZ = 1024
ADDR = (HOST, PORT)

tcpCliSock = socket(AF_INET, SOCK_STREAM)
tcpCliSock.connect(ADDR)

while True:
    data1 = tcpCliSock.recv(BUFSIZ).decode()
    if not data1:
        break
    print (data1)

    data = input('> ')
    if not data:
        break
    tcpCliSock.send(('[%s] %s' % (ctime(), data)).encode())

tcpCliSock.close()
```

TsServer.py

```
#!/usr/bin/env python

from socket import *
from time import ctime

HOST = ''
PORT = 21567
BUFSIZ = 1024
ADDR = (HOST, PORT)

tcpSerSock = socket(AF_INET, SOCK_STREAM)
tcpSerSock.bind(ADDR)
tcpSerSock.listen(5)

while True:
    print ('waiting for connection...')
    tcpCliSock, addr = tcpSerSock.accept()
    print ('...connected from:', addr)

    while True:
        data = input('> ')
        if not data:
            break
        tcpCliSock.send(('[%s] %s' % (ctime(), data)).encode())

        data1 = tcpCliSock.recv(BUFSIZ).decode()
        if not data1:
            break
        print (data1)

    tcpCliSock.close()
tcpSerSock.close()
```

TsClient.py :-

```
#!/usr/bin/env python

from socket import *
from time import ctime

HOST = 'localhost'
PORT = 21567
BUFSIZ = 1024
ADDR = (HOST, PORT)

tcpCliSock = socket(AF_INET, SOCK_STREAM)
tcpCliSock.connect(ADDR)

while True:
    data1 = tcpCliSock.recv(BUFSIZ).decode()
    if not data1:
        break
    print (data1)

    data = input('> ')
    if not data:
        break
    tcpCliSock.send(('[%s] %s' % (ctime(), data)).encode())

tcpCliSock.close()
```

TsServer.py :-

```
#!/usr/bin/env python

from socket import *
from time import ctime

HOST = ''
PORT = 21567
BUFSIZ = 1024
ADDR = (HOST, PORT)

tcpSerSock = socket(AF_INET, SOCK_STREAM)
tcpSerSock.bind(ADDR)
tcpSerSock.listen(5)

while True:
   print ('waiting for connection...')
   tcpCliSock, addr = tcpSerSock.accept()
   print ('...connected from:', addr)

   while True:
      data = input('> ')
      if not data:
         break
      tcpCliSock.send(('[%s] %s' % (ctime(), data)).encode())

      data1 = tcpCliSock.recv(BUFSIZ).decode()
      if not data1:
         break
      print (data1)

   tcpCliSock.close()
tcpSerSock.close()
```

Execution Result