Ayush Goyal
190905522 CSE D 62

Distributed Systems Week 4: Lab 4: Socket Programming using Python

Lab Excercise:

1. Write a UDP time server to display the current time and day.

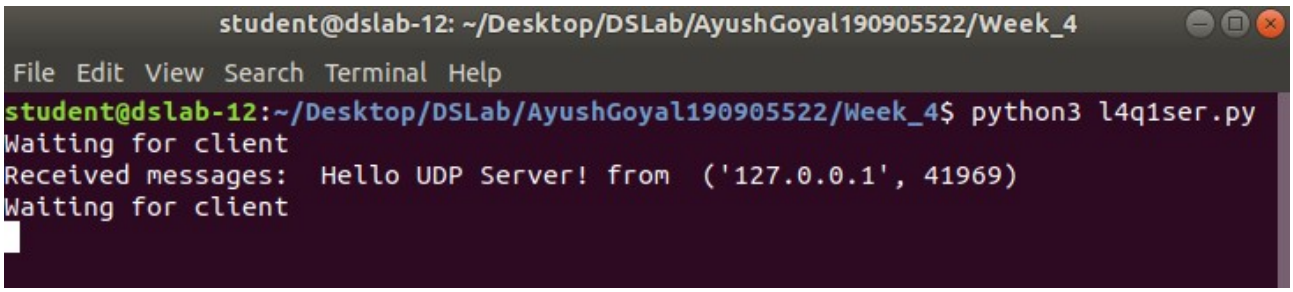
Time Server: “l4q1ser.py”:

```
import socket
import time

sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
udp_host = socket.gethostname()
udp_port = 12345

sock.bind((udp_host, udp_port))

while True:
        print("Waiting for client")
        data,addr = sock.recvfrom(1024)
        print("Received messages: ", data.decode(),"from ", addr)

        # Sending a reply to client
        currentTime = time.ctime(time.time()) + "\r\n"
        bytesToSend = str.encode(currentTime)
        sock.sendto(bytesToSend, addr)
```

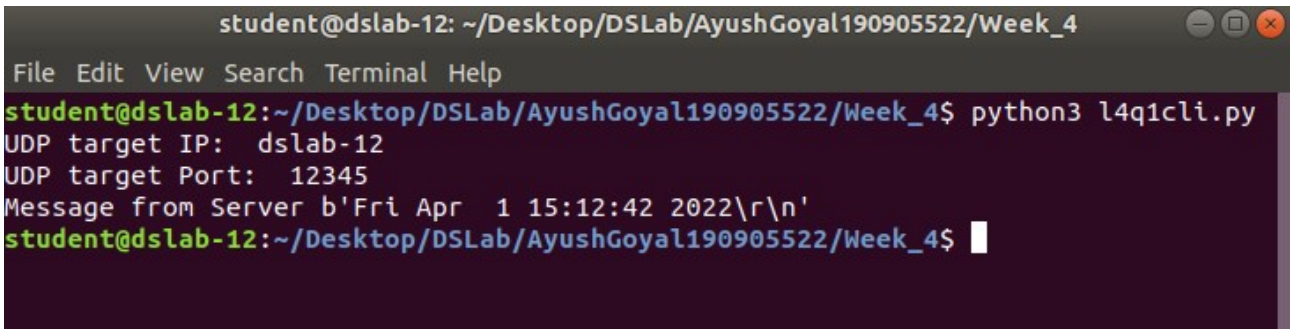
Client: “l4q1cli.py”:

```
import socket

sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)

udp_host = socket.gethostname()
udp_port = 12345

msg = "Hello UDP Server!"
print("UDP target IP: ", udp_host)
print("UDP target Port: ", udp_port)

sock.sendto(msg.encode(), (udp_host, udp_port))
msgFromServer = sock.recvfrom(1024)
msg = "Message from Server {}".format(msgFromServer[0])
print(msg)
```

Server Side Terminal Output:



Client Side Terminal Output:



2. Write a UDP simple chat program for message send and receive.

Server: "l4q2ser.py":

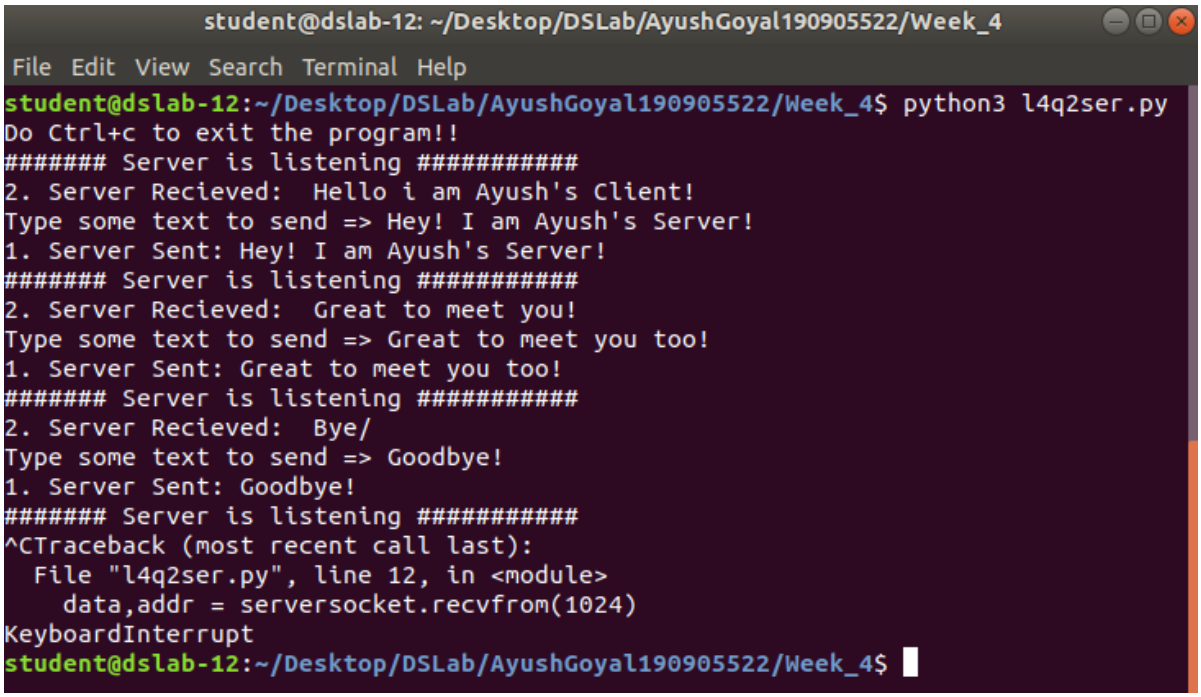
```
import socket

serversocket = socket.socket(socket.AF_INET,socket.SOCK_DGRAM)

udp_host = socket.gethostname()
udp_port = 12345

serversocket.bind((udp_host,udp_port))
print('Do Ctrl+c to exit the program!!')
while True:
        print("######## Server is listening ##########")
        data,addr = serversocket.recvfrom(1024)
        print('2. Server Recieved: ', data.decode())
        message = input('Type some text to send => ')
        serversocket.sendto(message.encode(),addr)
        print('1. Server Sent:', message)
```

Client: "l4q2cli.py":

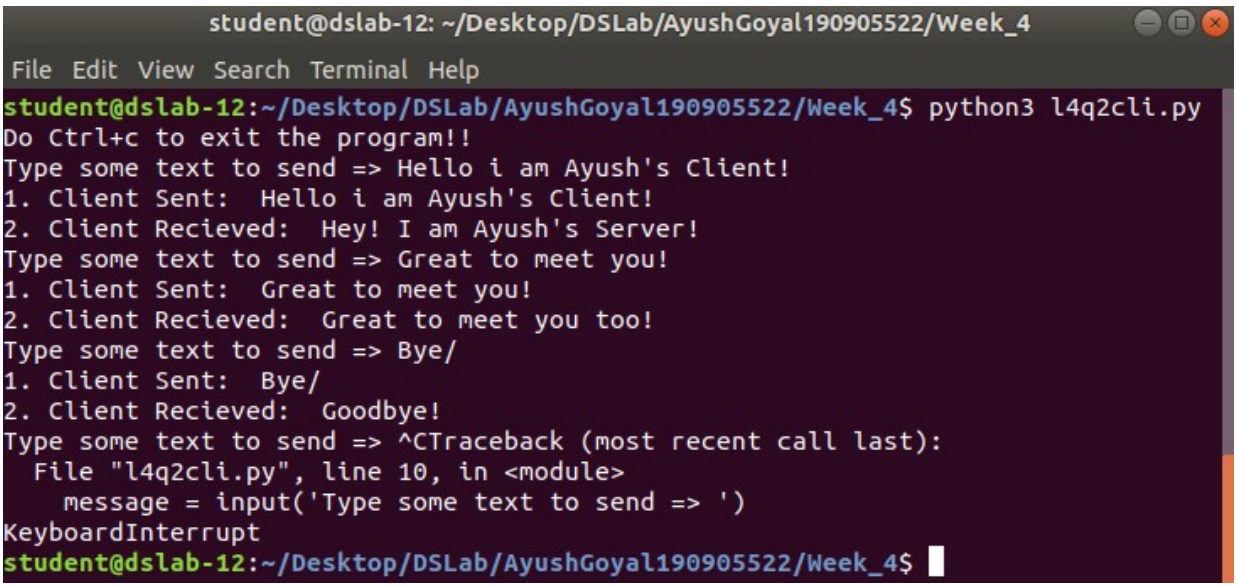
```
import socket

s = socket.socket(socket.AF_INET,socket.SOCK_DGRAM)
```

```
udp_host = socket.gethostname()
udp_port = 12345

print('Do Ctrl+c to exit the program!!')
while True:
        message = input('Type some text to send => ')
        s.sendto(message.encode(),(udp_host,udp_port))
        print('1. Client Sent: ',message)
        data = s.recv(1024)
        print('2. Client Recieved: ', data.decode())
```

Server Side Terminal Output:



Client Side Terminal Output:



3. Write a TCP/UDP peer to peer chat system between two different machines.

(Me and my friend, Dipesh Singh(190905520) have used peer to peer chat system using my pc as server and his pc as client and have shown output for inter device peer to peer chat system.)

1st PC: Ayush Goyal (190905522) : 172.16.58.44/24
2nd PC: Dipesh Singh (190905520) : 172.16.58.153/24

We can find my PC's IP Address by executing **ip addr**:

```
student@dslab-12:~/Desktop/DSLab/AyushGoyal190905522/Week_4$ ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen
1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp3s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group def
ault qlen 1000
    link/ether 8c:dc:d4:4f:c7:bf brd ff:ff:ff:ff:ff:ff
    inet 172.16.58.44/24 brd 172.16.58.255 scope global dynamic noprefixroute enp3s0
       valid_lft 378sec preferred_lft 378sec
    inet6 fe80::e64:4bce:7df1:a2bb/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
3: virbr0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group
default qlen 1000
    link/ether 52:54:00:89:63:fc brd ff:ff:ff:ff:ff:ff
    inet 192.168.122.1/24 brd 192.168.122.255 scope global virbr0
       valid_lft forever preferred_lft forever
4: virbr0-nic: <BROADCAST,MULTICAST> mtu 1500 qdisc fq_codel master virbr0 state DOWN g
roup default qlen 1000
    link/ether 52:54:00:89:63:fc brd ff:ff:ff:ff:ff:ff
student@dslab-12:~/Desktop/DSLab/AyushGoyal190905522/Week_4$
```

Similarly we can find Dipesh's PC's IP Address too:

```
student@dslab-12:~/Desktop/190905520/DS-LAB/lab3/q3$ ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp3s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 8c:dc:d4:4c:a5:ac brd ff:ff:ff:ff:ff:ff
    inet 172.16.58.153/24 brd 172.16.58.255 scope global dynamic noprefixroute enp3s0
       valid_lft 429sec preferred_lft 429sec
    inet6 fe80::a158:554b:cf6b:65f4/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
student@dslab-12:~/Desktop/190905520/DS-LAB/lab3/q3$
```

Server: "l4q3ser.py":

```
import socket

host = '172.16.58.44'
port = 3000
s = socket.socket()
s.bind((host, port))
s.listen(5)
conn, addr = s.accept()
print(conn, addr)
```

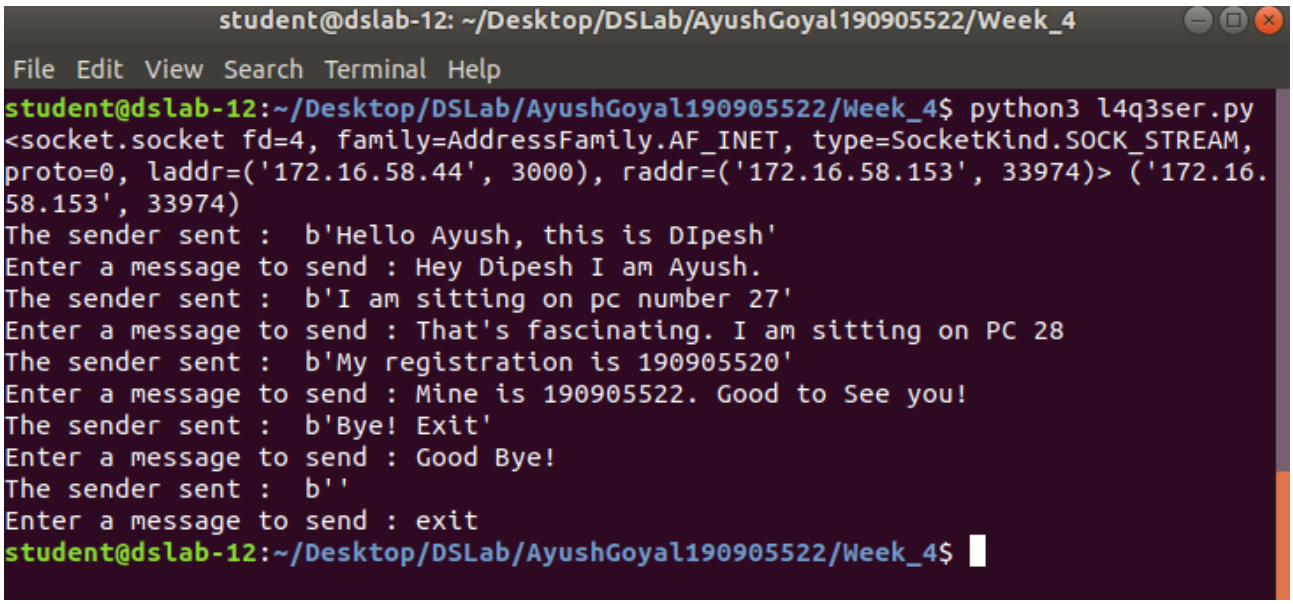
```
while True:
        data = conn.recv(1024)
        if str(data).find('exit') != -1:
                break
        print("The sender sent : ", str(data))
        msg = input("Enter a message to send : ")
        if msg.find('exit') != -1:
                break
        conn.sendall(str.encode(msg, encoding='utf-8'))
        conn.close()
```

Client: “l4q3cli.py”:

```
import socket
host = '172.16.58.44'
port = 3000
s = socket.socket()
s.connect((host, port))
while True:
        msg = input("Enter a message to send : ")
        if msg.find('exit') != -1:
                break
        s.sendall(str.encode(msg, encoding='utf-8'))
        data = s.recv(1024)
        if str(data).find('exit') != -1:
                break
        print("The sender sent : ", str(data))
s.close()
```

When my PC is the server and Dipesh’s PC is the client:

Server Side Terminal Output(My PC) :



Client Side Terminal Output (Dipesh's PC) :

```
student@dslab-12:~/Desktop/190905520/DS-LAB/lab3/q3$ python3 client.py
Enter a message to send : Hello Ayush, this is DIpesh
The sender sent :  Hey Dipesh I am Ayush.
Enter a message to send : I am sitting on pc number 27
The sender sent :  That's fascinating. I am sitting on PC 28
Enter a message to send : My registration is 190905520
The sender sent :  Mine is 190905522. Good to See you!
Enter a message to send : Bye! Exit
The sender sent :  Good Bye!
Enter a message to send : exit
student@dslab-12:~/Desktop/190905520/DS-LAB/lab3/q3$
```

When we switch the PC and now my PC is the client and Dipesh's PC is the server:

Server Side Terminal Output(Dipesh's PC) :

```
student@dslab-12:~/Desktop/190905520/DS-LAB/lab3/q3$ python3 server.py
<socket.socket fd=4, family=AddressFamily.AF_INET, type=SocketKind.SOCK_STREAM, proto=0, laddr=('172.16.58.153', 3000),
 raddr=('172.16.58.44', 32818)> ('172.16.58.44', 32818)
The sender sent :  Hey we meet again!
Enter a message to send : This time you, Ayush, are the client, then what am I?
The sender sent :  You, Dipesh, are the server this time.
Enter a message to send : That is really interesting, what is your IP address?
The sender sent :  My IP Address is 172.16.58.44. What's yours?
Enter a message to send : Mine is 172.16.58.153, you should know this alreasy since you connected to me first :P
The sender sent :  That's right haha. this was nice! I'll see you around!
Enter a message to send : Bye!
The sender sent :  Good Bye!\
Enter a message to send : exit
student@dslab-12:~/Desktop/190905520/DS-LAB/lab3/q3$
```

Client Side Terminal Output (My PC) :

```
student@dslab-12: ~/Desktop/DSLab/AyushGoyal190905522/Week_4
File  Edit  View  Search  Terminal  Help
student@dslab-12:~/Desktop/DSLab/AyushGoyal190905522/Week_4$ python3 l4q3cli.py
Enter a message to send : Hey we meet again!
The sender sent :  b'This time you, Ayush, are the client, then what am I?'
Enter a message to send : You, Dipesh, are the server this time.
The sender sent :  b'That is really interesting, what is your IP address?'
Enter a message to send : My IP Address is 172.16.58.44. What's yours?
The sender sent :  b'Mine is 172.16.58.153, you should know this alreasy since y
ou connected to me first :P'
Enter a message to send : That's right haha. this was nice! I'll see you around!
The sender sent :  b'Bye!'
Enter a message to send : Good Bye!\
The sender sent :  b''
Enter a message to send : exit
student@dslab-12:~/Desktop/DSLab/AyushGoyal190905522/Week_4$
```

4. Try to debug the error in the code and execute it.

Wrong Codes:

Client:

```
import socket

serverIP = 'localhost'
serverPort = 16000

clientSock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
clientSock.connect((serverIP, serverPort))

message = raw_input("Input integers with space in between: ")
message2 = raw_input("Enter the length of the set: ")

clientSock.send(message)
clientSock.send(message2)

data = clientSock.recv(1024)
temp = [float(x) for x in data.split(' ')]

print("The total of all numbers is: "+str(temp[0]))
print("The lowest number is: "+str(temp[1]))
print("The highest number is: "+str(temp[2]))
print("The mean is: "+str(temp[3]))

clientSock.close()
```

Server:

```
import socket

serverIP = 'localhost'
serverPort = 16000
serverSock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
serverSock.bind((serverIP, serverPort))
serverSock.listen(1)
print("TCP Serverhas started and is ready to receive!")

while 1:
        connection, addr = serverSock.accept()
        data = connection.recv(1024)
        if not data: break
        temp = [float(x) for x in data.split(' ')]
        print "Received data : ", temp
```

```
        length = len(temp)
        maximum = max(temp)
        minimum = min(temp)
        total = sum(temp)
        mean = total/length
        msg = str(total) + " " + str(minimum) + ' ' + str(maximum) + " " + str(mean)
        connection.send(str(msg))
```

Expected Output:

- Below is an example input from clientTCP side:
- `Input integers with space in between: 5 4 6 9 3.4`
- Next, the user will enter the length of the above set when prompted from clientUDP side:
- `Enter the length of the set: 5`
- Next, you will notice a screen output on the serverTCP side as follows:
- `TCP server has started and is ready to receive`
- `Received data: [5.0, 4.0, 6.0, 9.0, 3.4]`
- Finally you will should see an output on the clientTCP side as follows:
- `The total of all numbers is: 27.4`
- `The lowest number is: 3.4`
- `The highest number is: 9.0`

`The mean is: 5.48`

Changes:

In Server:

Move this out of the while loop:

```
connection, addr = serverSock.accept()
```

Changed from 1 to True:

```
while True:
```

Add decode method:

```
temp = [float(x) for x in data.decode('utf-8').split(' ')]
```

Put paranthesis:

```
print("Received data : ", temp)
```

Add str.encode()

```
connection.send(str.encode(msg, 'utf-8'))
```

In Client:

Change to input and remove message2:

```
message = input("Input integers with space in between: ")
```

Add encode:

```
clientSock.send(str.encode(message, 'utf-8'))
```

Add decode:

```
temp = [float(x) for x in data.decode('utf-8').split(' ')]
```

Corrected Codes:

Client:

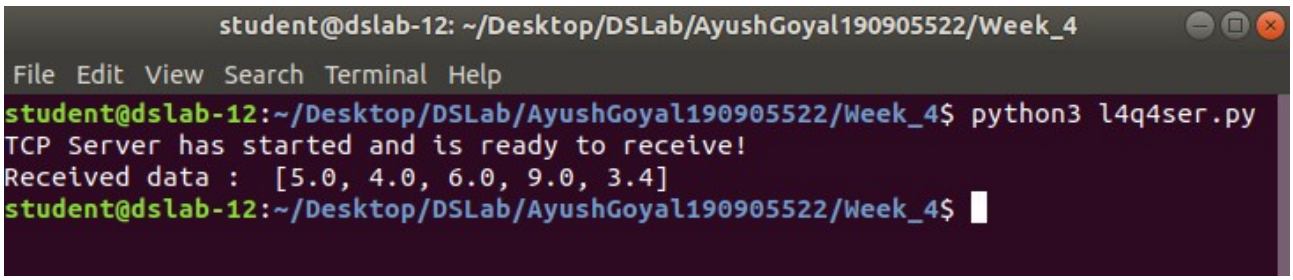
```
import socket

serverIP = 'localhost'
serverPort = 16000

clientSock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

clientSock.connect((serverIP, serverPort))

message = input("Input integers with space in between: ") # change to input and remove message2
clientSock.send(str.encode(message, 'utf-8')) #add encode

data = clientSock.recv(1024)

temp = [float(x) for x in data.decode('utf-8').split(' ')] #add decode

print("The total of all numbers is: "+str(temp[0]))
print("The lowest number is: "+str(temp[1]))
print("The highest number is: "+str(temp[2]))
print("The mean is: "+str(temp[3]))

clientSock.close()
```

Server:

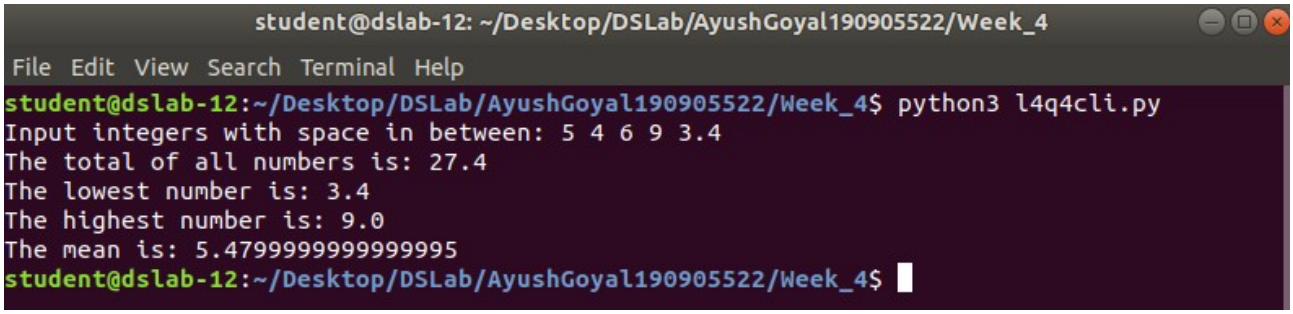
```
import socket

serverIP = 'localhost'
serverPort = 16000
```

```
serverSock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
serverSock.bind((serverIP, serverPort))
serverSock.listen(1)

print("TCP Server has started and is ready to receive!")
connection, addr = serverSock.accept() # move this out of the while loop

while True: # changed from 1 to True
        data = connection.recv(1024)
        if not data: break
        temp = [float(x) for x in data.decode('utf-8').split(' ')] # add decode method
        print("Received data : ", temp) # put parentheses
        length = len(temp)
        maximum = max(temp)
        minimum = min(temp)
        total = sum(temp)
        mean = total/length
        msg = str(total) + " " + str(minimum) + ' ' + str(maximum) + " " + str(mean)
        connection.send(str.encode(msg, 'utf-8')) # add str.encode
```

Correct Server Output:



Correct Client Output:



THE END