**190905514** **MOHAMMAD TOFIK** **BATCH : C3**
**ROLLNO : 62** **SEM : 6th** **SECTION : C**

---

# DS-LAB -4:

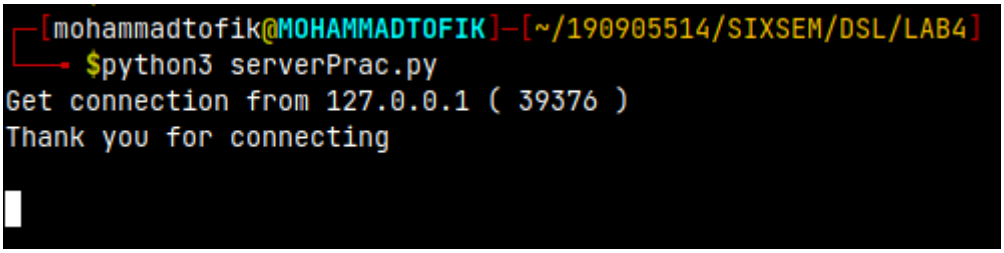
## PRACTICE :

### serverPrac.py

```
import socket
import os
host = socket.gethostname()
port = 9113
conn = socket.socket()
conn.bind((host, port))
conn.listen(5)
(connClient, clientAddr) = conn.accept()
print('Get connection from', clientAddr[0], '(', clientAddr[1], ')')
print("Thank you for connecting \n")
while True:
msg = connClient.recv(1024)
if not msg: break
conn.close()
```

### clientPrac.py

```
import socket
import os

host = socket.gethostname()
port = 9113
conn = socket.socket()
conn.connect((host, port))
conn.sendall(b'Welcome User !')
msg = conn.recv(1024)
conn.close()
print(repr(msg))
```



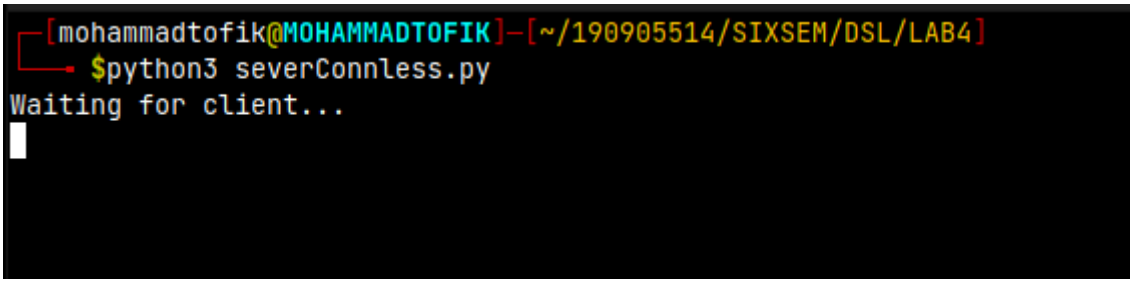


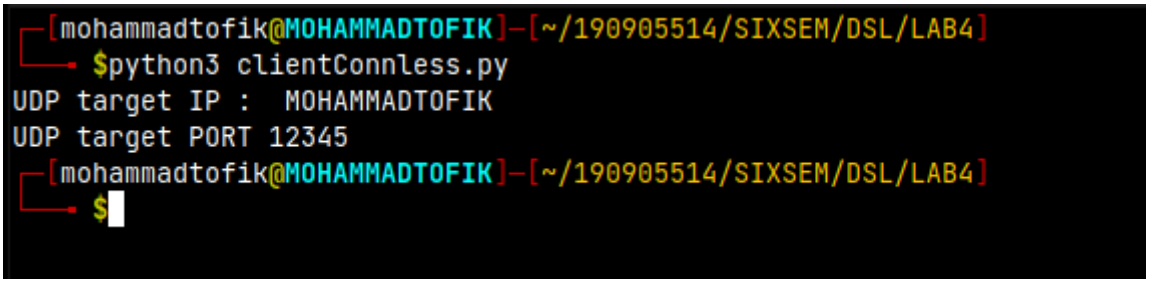
**serverConnless.py**

```
import socket
import os

connSocket = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
udpHost = socket.gethostname()
udpPort = 1234
connSocket.bind((udpHost, udpPort))
while True:
print("Waiting for client ... ")
data, clientAddr = connSocket.recvfrom(1024)
print("Received Messages : ", data.decode(), "from", clientAddr)
```

**clientConnless.py**

```
import socket
import os

connSocket = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
udpHost = socket.gethostname()
udpPort = 12345
msg = "UDP Program"
print("UDP target IP : ", udpHost)
print("UDP target PORT", udpPort)
connSocket.sendto(msg.encode(), (udpHost, udpPort))
```



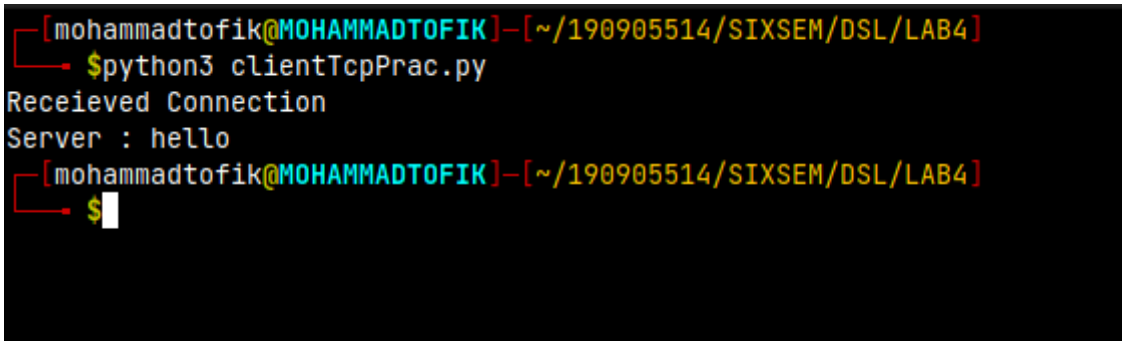




**serverTcpPrac.py**

```
import socket
import os

HOST = 'localhost'
PORT = 2020
with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as conn:
conn.bind((HOST, PORT))
conn.listen()
clientConn, clientAddr = conn.accept()
while conn:
print("Connected by", clientAddr)
while True:
data = clientConn.recv(1024)
if data:
print("Client : ", data.decode())
data = input("Enter the message to client : ")
if not data:
break
clientConn.sendall(bytearray(data, 'utf-8'))
conn.close()
```

**cientTcpPrac.py**

```
import socket
import os

HOST = 'localhost'
PORT = 2020
with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as conn:
conn.connect((HOST, PORT))
conn.sendall(b'Hello World')
data = conn.recv(1024)
print('Receieved Connection')
print('Server :', data.decode())
```

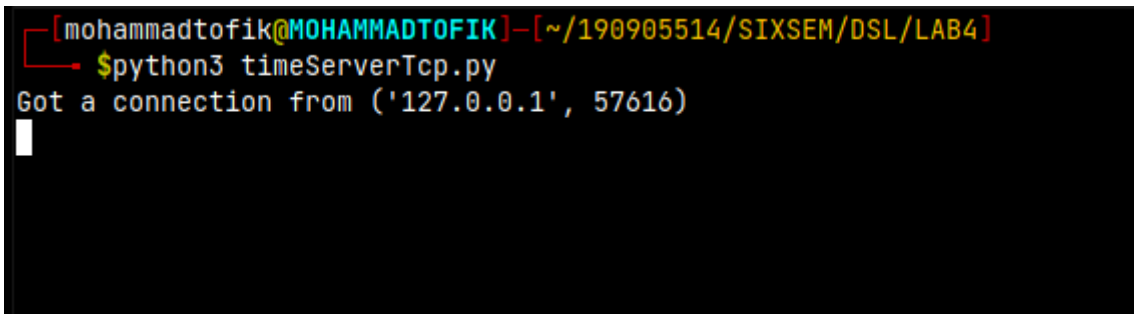

**timeServer.py**

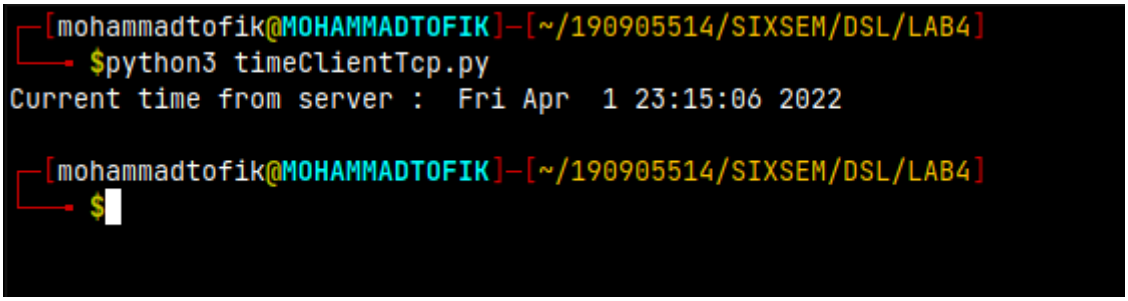
```
import socket
import time

connSocket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
HOST = socket.gethostname()
PORT = 9991
connSocket.bind((HOST, PORT))
connSocket.listen(5)
while True:
connClient, clientAddr = connSocket.accept()
print("Got a connection from %s" % str(clientAddr))
currentTime = time.ctime(time.time()) + "\r\n"
```

```
connClient.send(currentTime.encode('ascii'))
connClient.close()
```

**timeClient.py**

```
import socket

connSocket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
HOST = socket.gethostname()
PORT = 9991
connSocket.connect((HOST, PORT))
clientTime = connSocket.recv(1024)
print("Current time from server : ", clientTime.decode())
connSocket.close()
```




**serverTcpChat.py**

```
import socket

HOST = 'localhost'
PORT = 1999
connSocket = socket.socket()
connSocket.bind((HOST, PORT))
connSocket.listen()
print("Waiting for incoming connection ... \n")
clientConn, clientAddr = connSocket.accept()
print("Received connection from : ", clientAddr[0], "(", clientAddr[1],
")\n")
s_name = clientConn.recv(1024)
```

```
s_name = s_name.decode()
print(s_name, "Has connected to the chat room\nEnter [e] to exit chat
room\n")
name = input(str("Enter your name : "))
clientConn.send(name.encode())
while True:
msg = input(str("Me : "))
if msg == "[e]":
msg = "Left Chat room"
clientConn.send(msg.encode())
print("\n")
break
clientConn.send(msg.encode())
msg = clientConn.recv(1024)
msg = msg.decode()
print(s_name, ":", msg)
```

**clientTcpChat.py**

```
import socket

HOST = 'localhost'
PORT = 1999
connSocket = socket.socket()
name = input("Enter your name : ")
print("\n Trying to connect to ", HOST, "(", PORT, ")\n")
connSocket.connect((HOST, PORT))
print("Connected ... \n")
connSocket.send(name.encode())
s_name = connSocket.recv(1024)
s_name = s_name.decode()
print(s_name, "Has joined to the chat room\nEnter [e] to exit chat room\
n")
while True:
msg = connSocket.recv(1024)
msg = msg.decode()
print(s_name, ":", msg)
msg = input(str('Me : '))
if msg == "[e]":
msg = "Left the room"
connSocket.send(msg.encode())
print("\n")
connSocket.send(msg.encode())
```

```
┌─[mohammadtofik@MOHAMMADTOFIK]─[~/190905514/SIXSEM/DSL/LAB4]
└──╼ $python3 serverTcpChat.py
Waiting for incoming connection...

Received connection from :  127.0.0.1 ( 49706 )

mohammad Has connected to the chat room
Enter [e] to exit chat room

Enter your name : khan
Me : hi
mohammad : yes
Me : [e]
```

```
┌─[mohammadtofik@MOHAMMADTOFIK]─[~/190905514/SIXSEM/DSL/LAB4]
└──╼ $python3 clientTcpChat.py
Enter your name : mohammad

 Trying to connect to  localhost ( 1999 )

Connected...

khan Has joined to the chat room
Enter [e] to exit chat room

khan : hi
Me : yes
khan : Left Chat room
Me : [e]
```

## EXCERCISE:

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

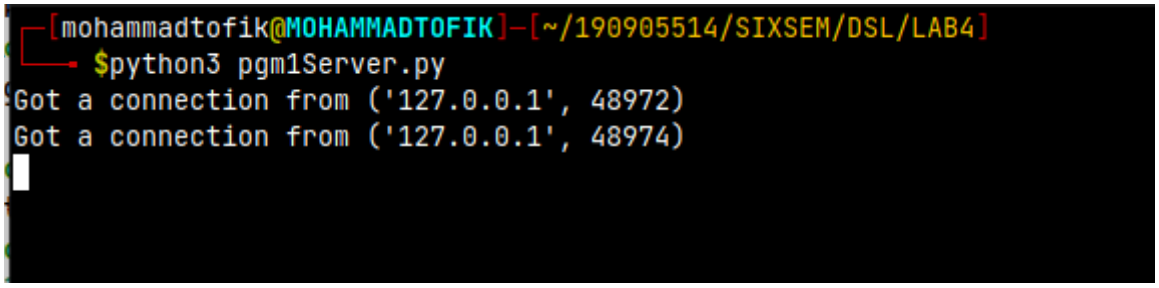
**Pgm1Server.py**

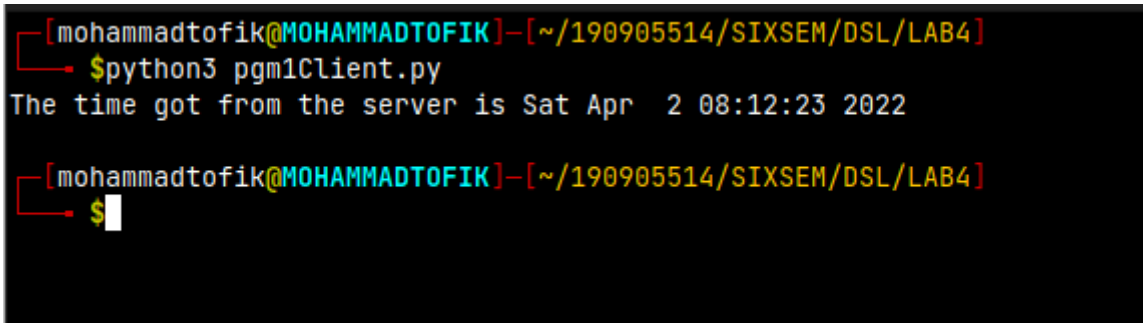
```
import socket
import time
# create a socket object
serversocket = socket.socket(
socket.AF_INET, socket.SOCK_STREAM)
# get local machine name
host = socket.gethostname()

port = 9999
# bind to the port
```

```
serversocket.bind((host, port))
# queue up to 5 requests
serversocket.listen(5)
while True:
# establish a connection
clientsocket,addr = serversocket.accept()
print("Got a connection from %s" % str(addr))
currentTime = time.ctime(time.time()) + "\r\n"
clientsocket.send(currentTime.encode('ascii'))
clientsocket.close()
```

**pgm1Client.py**

```
import socket
# create a socket object
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
# get local machine name
host = socket.gethostname()
port = 9999
# connection to hostname on the port.
s.connect((host, port))
# Receive no more than 1024 bytes
tm = s.recv(1024)
s.close()
print("The time got from the server is %s" % tm.decode('ascii'))
```



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

**pgm2Server.py**

```
import socket


def server_program():
# get the hostname
host = socket.gethostname()
port = 5000 # initiate port no above 1024
server_socket = socket.socket() # get instance
# look closely. The bind() function takes tuple as argument
server_socket.bind((host, port)) # bind host address and port together
# configure how many client the server can listen simultaneously
server_socket.listen(2)
conn, address = server_socket.accept() # accept new connection
print("Connection from: " + str(address))
while True:
# receive data stream. it won't accept data packet greater than 1024
bytes
data = conn.recv(1024).decode()
if not data:
# if data is not received break
break
print("from connected user: " + str(data))
data = input(' → ')
conn.send(data.encode()) # send data to the client
conn.close() # close the connection
if __name__ == '__main__':
server_program()
```

**pgm2Client.py**

```
import socket


def client_program():
host = socket.gethostname() # as both code is running on same pc
port = 5000 # socket server port number

client_socket = socket.socket() # instantiate
client_socket.connect((host, port)) # connect to the server

message = input(" → ") # take input
```

```
while message.lower().strip() ≠ 'bye':
client_socket.send(message.encode()) # send message
data = client_socket.recv(1024).decode() # receive response

print('Received from server: ' + data) # show in terminal

message = input(" -> ") # again take input

client_socket.close() # close the connection


if __name__ == '__main__':
client_program()
```

```
┌─[mohammadtofik@MOHAMMADTOFIK]─[~/190905514/SIXSEM/DSL/LAB4]
└──╼ $python3 pgm2Server.py
Connection from: ('127.0.0.1', 35968)
from connected user: hello
 -> how are you
from connected user: i am fine and how you
 -> i am too good
from connected user: ok
 -> done
```

```
┌─[mohammadtofik@MOHAMMADTOFIK]─[~/190905514/SIXSEM/DSL/LAB4]
└──╼ $python3 pgm2Client.py
 -> hello
Received from server: how are you
 -> i am fine and how you
Received from server: i am too good
 -> ok
Received from server: done
 ->
```

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

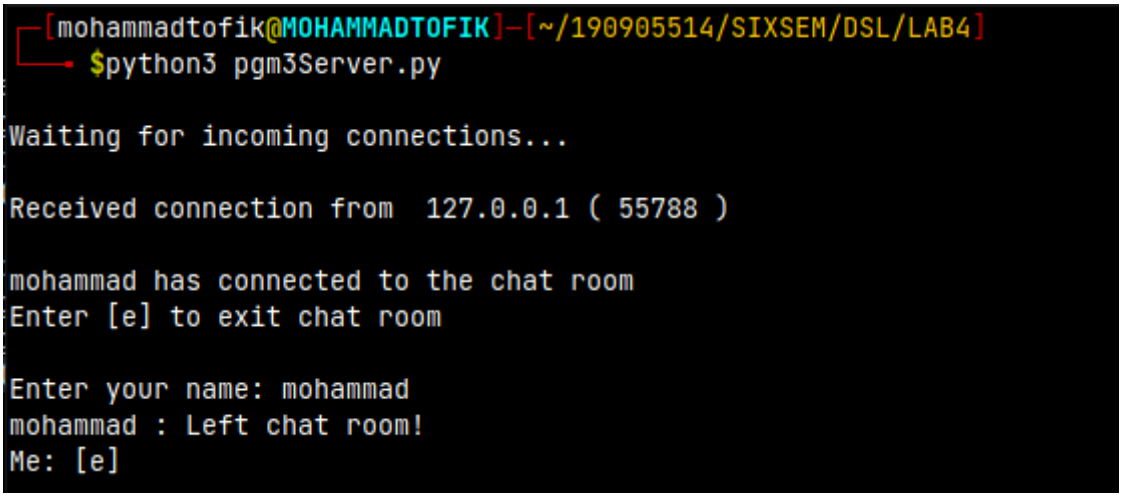
**Pgm3Server.py**

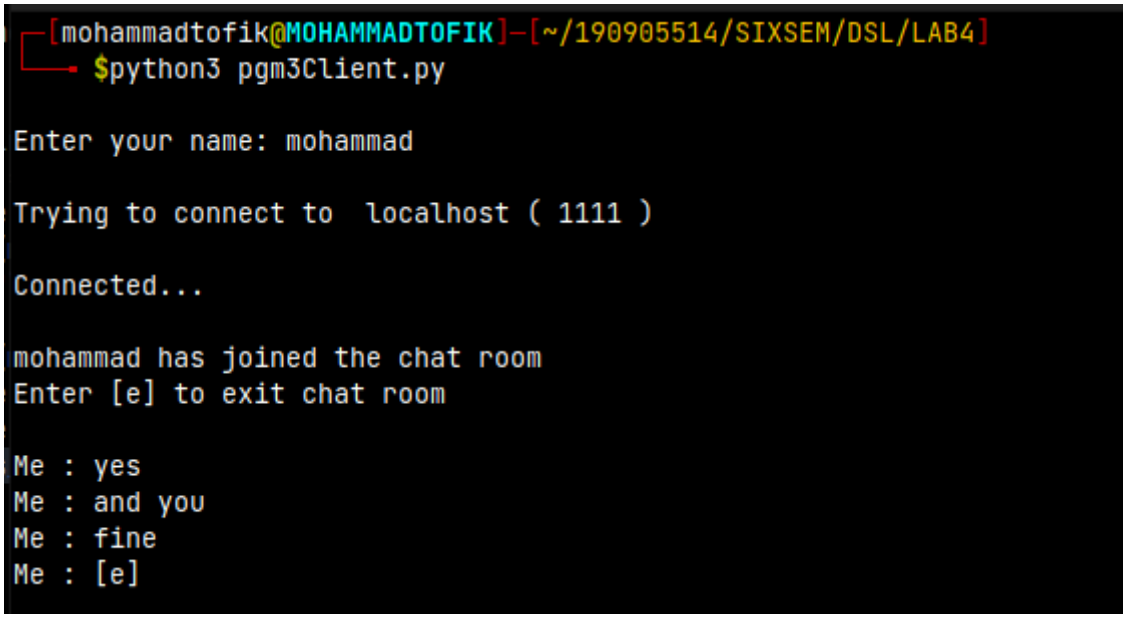
```
import socket
HOST = '172.16.57.223'
PORT = 31620
s = socket.socket()
s.bind((HOST, PORT))
s.listen()
```

```
print("\nWaiting for incoming connections ... \n")
conn, addr = s.accept()
print("Received connection from ", addr[0], "(", addr[1], ")\n")
s_name = conn.recv(1024)
s_name = s_name.decode()
print(s_name, "has connected to the chat room\nEnter [e] to exit chat
room\n")
name = input(str("Enter your name: "))
conn.send(name.encode())
while True:
message = conn.recv(1024)
message = message.decode()
print(s_name, ":", message)
message = input(str("Me: "))
if message == "[e]":
message = "Left chat room!"
conn.send(message.encode())
print("\n")
break
conn.send(message.encode())
```

**pgm3Client.py**

```
import socket
HOST = 'localhost'
PORT = 1111
s = socket.socket()
name = input(str("\nEnter your name: "))
print("\nTrying to connect to ", HOST, "(", PORT, ")\n")
s.connect((HOST, PORT))
print("Connected ... \n")
s.send(name.encode())
s_name = s.recv(1024)
s_name = s_name.decode()
print(s_name, "has joined the chat room\nEnter [e] to exit chat room\n")
while True:
message = input(str("Me : "))
if message == "[e]":
message = "Left chat room!"
s.send(message.encode())
print("\n")
break
s.send(message.encode())
message = s.recv(1024)
message = message.decode()
print(s_name, ":", message)
```




**4.Try to debug the error**

**pgm4Server.py**

```
import socket
serverIP = 'localhost'
serverPORT = 1515
serverSocket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
serverSocket.bind((serverIP, serverPORT))
serverSocket.listen(1)
print("TCP server has started and ready to recieve : ")
connection, addr = serverSocket.accept()
print("connected ... \n")
while 1:
data = connection.recv(1024)
if not data:
break
```

```
data = data.decode()
print(data)
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.encode(msg))
```

**pgm4Client.py**

```
import socket
serverIP = 'localhost'
serverPORT = 1515
clientSocket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
clientSocket.connect((serverIP, serverPORT))
message1 = str(input("Input integer with space between : "))
message2 = str(input("Enter the length of the set : "))
clientSocket.send(str.encode(message1))
data = clientSocket.recv(1024)
data = data.decode()
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]))
clientSocket.close()
```

```
┌─[mohammadtofik@MOHAMMADTOFIK]─[~/190905514/SIXSEM/DSL/LAB4]
└──╼ $python3 pgm4Client.py
Input integer with space between : 1 2 3 4
Enter the length of the set : 4
The total of all numbers is = 10.0
The lowest number is = 1.0
The highest number  is = 4.0
The mean is = 2.5
┌─[mohammadtofik@MOHAMMADTOFIK]─[~/190905514/SIXSEM/DSL/LAB4]
└──╼ $
```

```
┌─[mohammadtofik@MOHAMMADTOFIK]─[~/190905514/SIXSEM/DSL/LAB4]
└──╼ $python3 pgm4Server.py
TCP server has started and ready to recieve :
connected...

1 2 3 4
Received data :  [1.0, 2.0, 3.0, 4.0]
┌─[mohammadtofik@MOHAMMADTOFIK]─[~/190905514/SIXSEM/DSL/LAB4]
└──╼ $
```