Assignment 3

Implementation of TCP Socket Programming

NAME: Shirish Manoj Bobde

Reg. No.: 812

Roll No.: ECE/21152

Problem Statements 1

Write a TCP socket program (in C/C++/Java/Python) to establish connection between client and server. The client program will send a message to the server and the server program will display the message.

Code

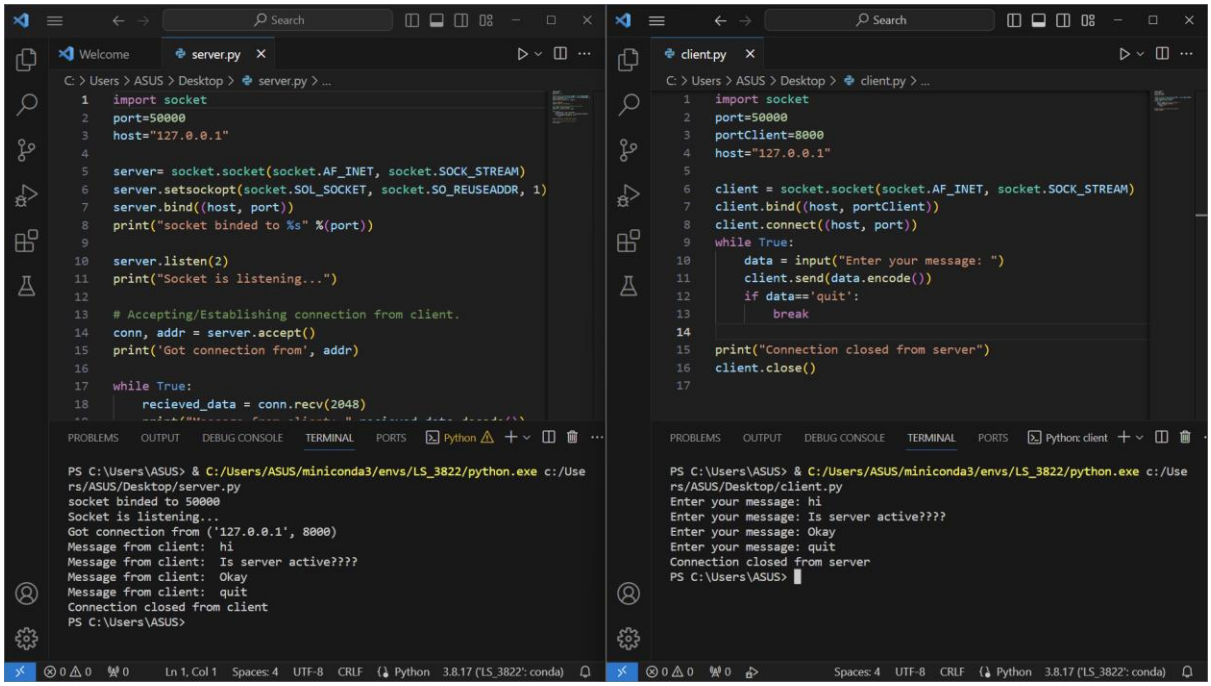
SERVER END

```
import socket
port=50000
host="127.0.0.1"

server= socket.socket(socket.AF_INET, socket.SOCK_STREAM)
server.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
server.bind((host, port))
print("socket binded to %s" %(port))

server.listen(2)
print("Socket is listening...")

# Accepting/Establishing connection from client.
conn, addr = server.accept()
print('Got connection from', addr)

while True:
   recieved_data = conn.recv(2048)
   print("Message from client: ",recieved_data.decode())
   if recieved_data.decode()=='quit':
      break

print("Connection closed from client")

#Close the connection with the client
conn.close()
```

CLIENT END

```
import socket
port=50000
portClient=8000
host="127.0.0.1"

client = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
client.bind((host, portClient))
client.connect((host, port))
while True:
    data = input("Enter your message: ")
    client.send(data.encode())
    if data=='quit':
        break

print("Connection closed from server")
client.close()
```

Result



Everything is working as desired. Server is receiving message from client and displaying. Connection is closed when client inputs quit.

Problem Statements 2

Write a TCP socket program where client sends a message (string) to server; server echo back the characters at even position if length of the string is even otherwise, echo back the characters at odd position. This process continues until the client sends ‘bye’.

Code

SERVER END

```
import socket


port=30000
host="127.0.0.1"


server= socket.socket(socket.AF_INET, socket.SOCK_STREAM)
server.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
server.bind((host, port))
print("socket binded to %s" %(port))


server.listen(2)
print("Socket is listening...")


# Accepting/Establishing connection from client.
conn, addr = server.accept()
print('Got connection from', addr)


while True:
        recieved_data = conn.recv(2048)
        print("Message from client: ", recieved_data.decode())
        decoded_data = recieved_data.decode()


        if recieved_data.decode()=='bye':
                break


        characters = ""  # For storing Even or Odd characters
```

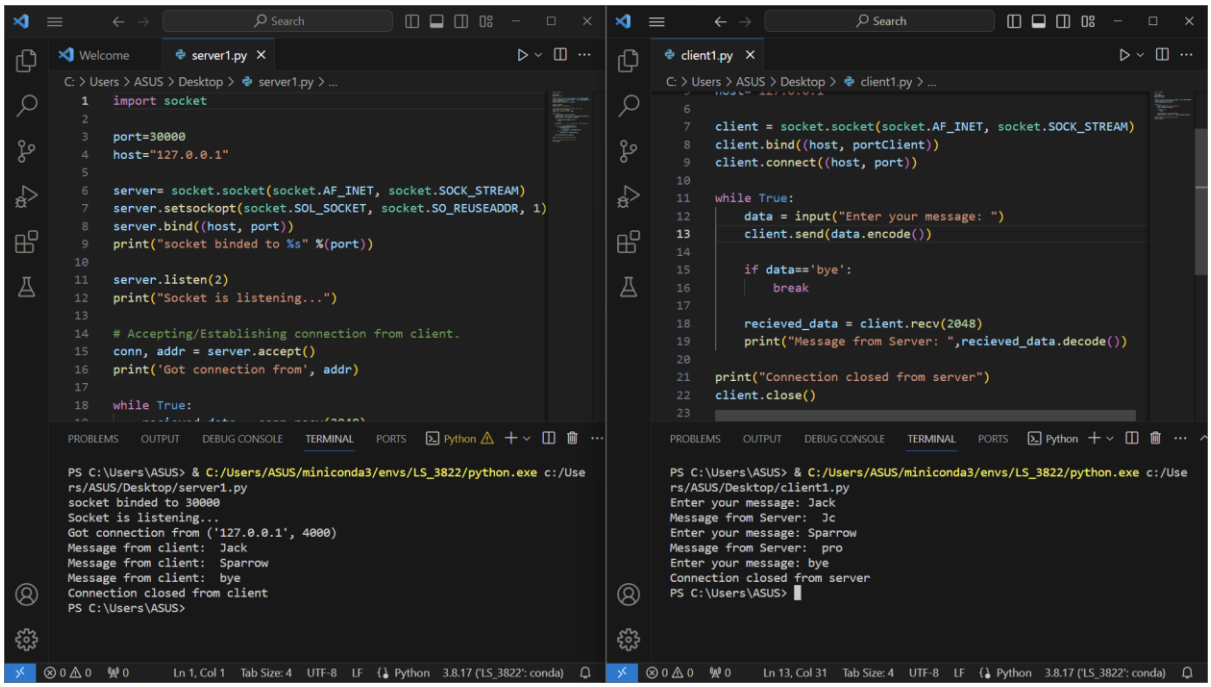
```
        for i in range(len(decoded_data)):
                if len(decoded_data) % 2 == 0:
                        if i % 2 == 0:
                                characters += decoded_data[i]
                elif i % 2 != 0 :
                        characters += decoded_data[i]

        conn.send(characters.encode())

print("Connection closed from client")

#Close the connection with the client
conn.close()
```

CLIENT END

```
import socket

port=30000
portClient=4000
host="127.0.0.1"

client = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
client.bind((host, portClient))
client.connect((host, port))

while True:
        data = input("Enter your message: ")
        client.send(data.encode())

        if data=='bye':
```

```
            break

        recieved_data = client.recv(2048)
        print("Message from Server: ",recieved_data.decode())

print("Connection closed from server")
client.close()
```

Result



The echo output printed on client end after getting echoed and even/odd filtering from server end is correct with respect to indexing as it, starts from 0. So, for “Jack” even index letters are J (0^{th} index) and c (2^{nd} index). Similarly for Sparrow odd index letters are p, r and o. When client inputs bye message connection is closed and no return message to client is sent.