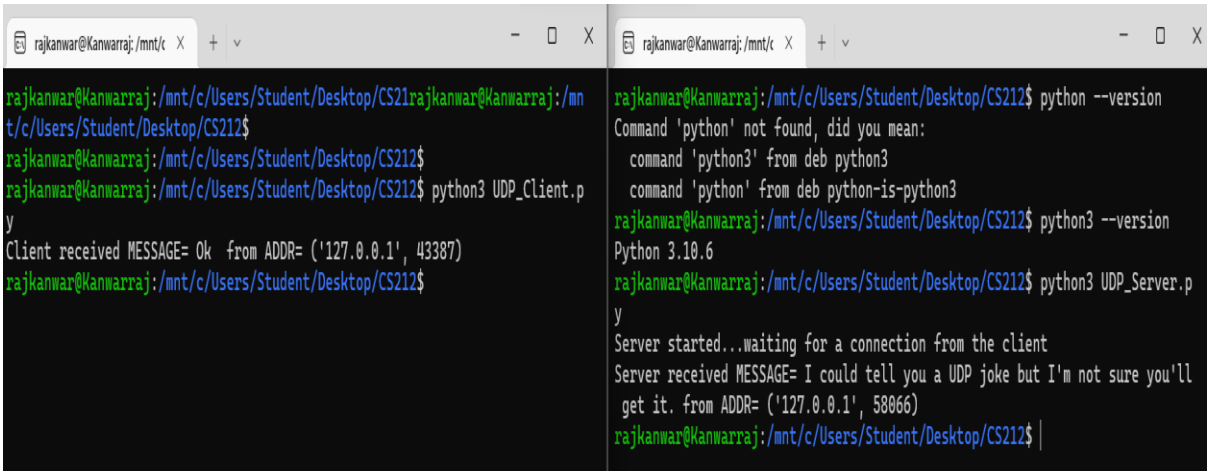
LAB2 – SOCKETS Computer Network (CS212)

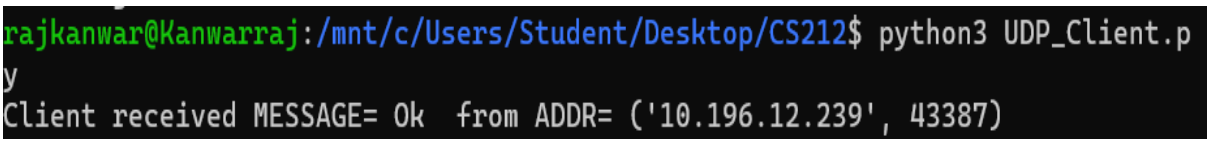
Date: - 01/02/2023 Kanwar raj Singh (1903122) &

PART A

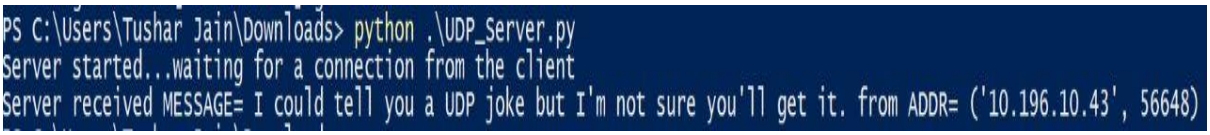
1. We got following Output after running given python files



2. I run UDP_Client.py in my computer and got following output:



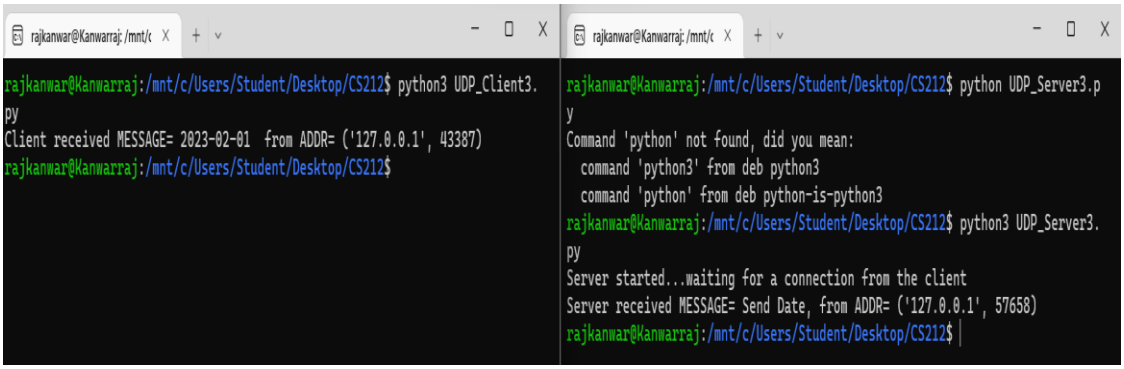
Output of another computer:



3. I used following code for sending date to client in Server file.

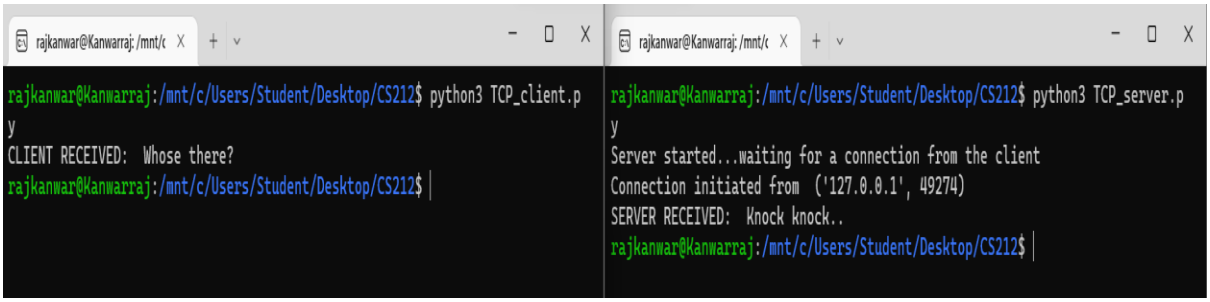
```
from datetime import date
dateAsString = str(date.today())

client.send(dateAsString.encode())
```



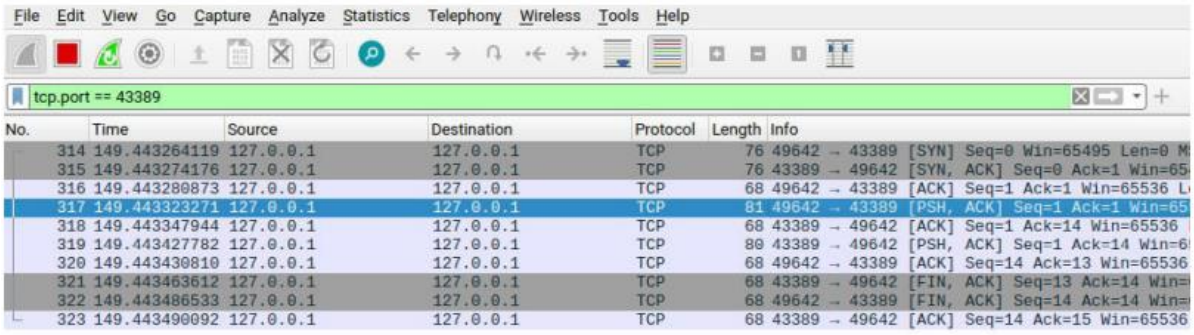
PART B

4. Following output, I got in terminal

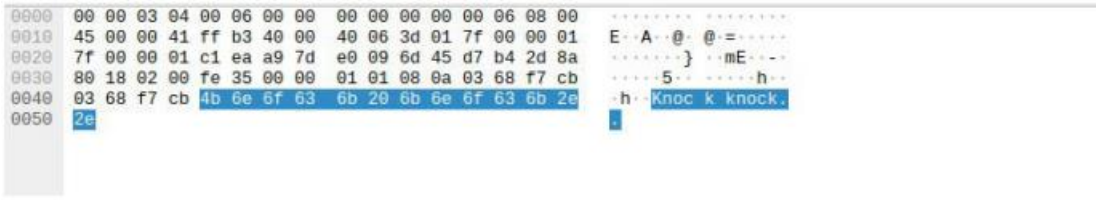


5. In the Wireshark, we started capturing the packets and applied the filter **tcp.port==43389**. Then we ran the TCP server and TCP client files in different windows of terminal. After the execution of the program, we investigated the contents of the packets captured. As we have not used HTTPS protocol, the messages were visible to us in the packet descriptions (the images of which are attached below).

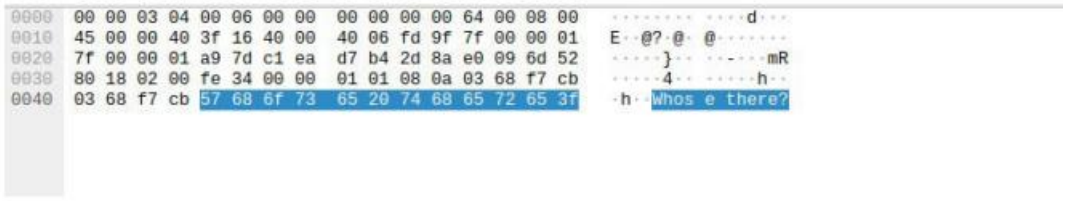
Packet captured: -



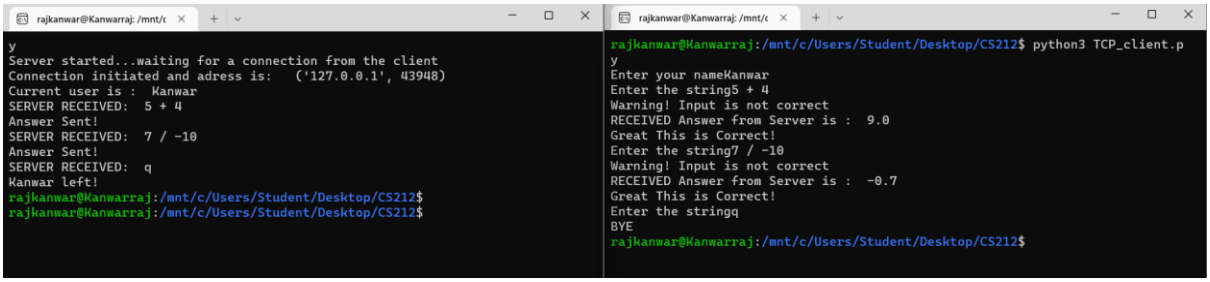
Message received by server: -



Message received by client: -



6. Output:



7. A **stateless protocol** is a communication protocol in which the receiver must not retain session state from previous requests. A protocol is stateful if any information about the connection is remembered. Whereas in a stateless communication, no information related to the connection is remembered.
   • Yes, the application-layer protocol designed by us for question 3 is Stateless as no information is remembered about any client. Whenever a client sends a request, server sends the message without any evaluation of prior information.

• No, the application-layer protocol designed by us for question 6 is not Stateless as the Server remembers the name sent by the client for the entire session.
• No, TCP is a Stateful protocol as a particular socket is assigned to every client and this socket is remembered throughout the session with the client.
• Yes, UDP is a stateless protocol as messages can be sent between any 2 sockets just by using the opposite one's Ip and port no. and as no remembering is happening.

8. Output which I got,