CS358 Assignment-03

Name: C M Devanand

RollNO: 2001CS19

Client.py:

This is a Python script that creates a client socket to connect to a server using TCP/IP, allowing the user to send messages to the server and receive responses. The script includes error handling for connection issues and user input validation.

Server1.py:

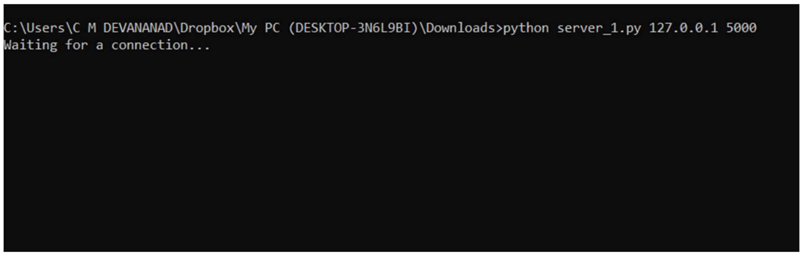
The server1.py is a simple single-process server that listens for incoming TCP connections on a specified port and IP address. Once a client connects to the server, it reads the data sent by the client and parses it as an arithmetic expression. If the expression is valid, the server evaluates it and sends the result back to the client. If the expression is not valid, the server sends an "Invalid input" message back to the client. The server closes the connection when the client disconnects.

How to run:

Step1:

Run server1.py in terminal using Command: python server1.py 127.0.0.1 5000

It will wait for the clients

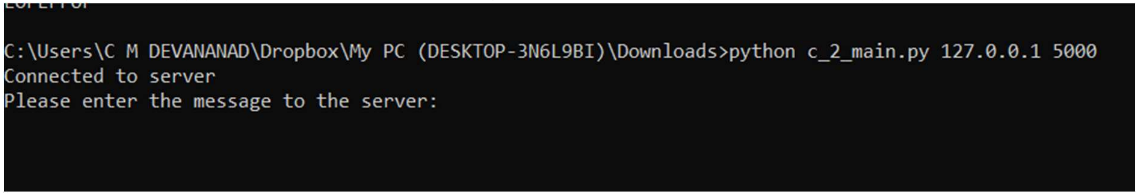


```
C:\Users\C M DEVANANAD\Dropbox\My PC (DESKTOP-3N6L9BI)\Downloads>python server_1.py 127.0.0.1 5000
Waiting for a connection...
```

Step2:

z

Run Client.py file in terminal using command: python client.py 127.0.0.1 5000



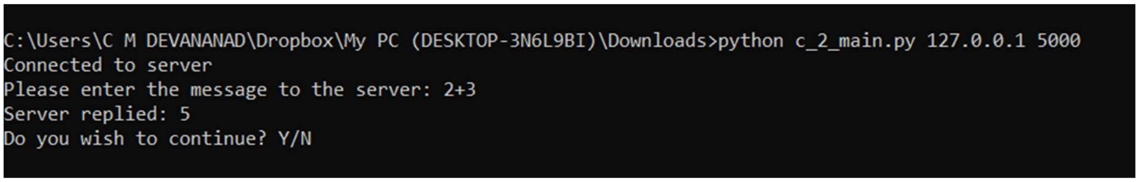
```
EOFError

C:\Users\C M DEVANANAD\Dropbox\My PC (DESKTOP-3N6L9BI)\Downloads>python c_2_main.py 127.0.0.1 5000
Connected to server
Please enter the message to the server:
```

Step3:

Enter the arithmetic expression in client if the input expression is wrong it shows error message.

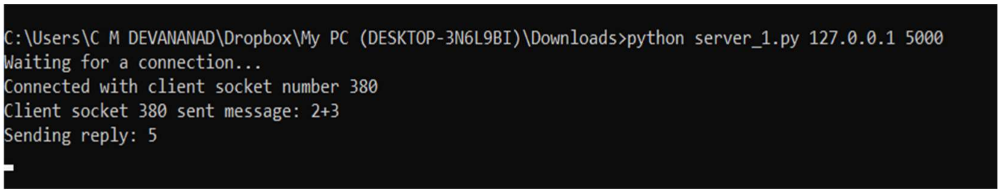
It will show the reply from the server and asks whether to continue or exist if yes type: y or n.

client1 response:



```
C:\Users\C M DEVANANAD\Dropbox\My PC (DESKTOP-3N6L9BI)\Downloads>python c_2_main.py 127.0.0.1 5000
Connected to server
Please enter the message to the server: 2+3
Server replied: 5
Do you wish to continue? Y/N
```

server response:

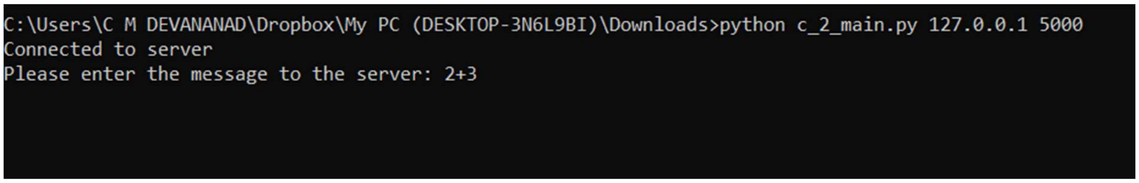


```
C:\Users\C M DEVANANAD\Dropbox\My PC (DESKTOP-3N6L9BI)\Downloads>python server_1.py 127.0.0.1 5000
Waiting for a connection...
Connected with client socket number 380
Client socket 380 sent message: 2+3
Sending reply: 5
```

These are steps to run the code.

If another client try to connect then server doesn’t connect to it until existed client ends.

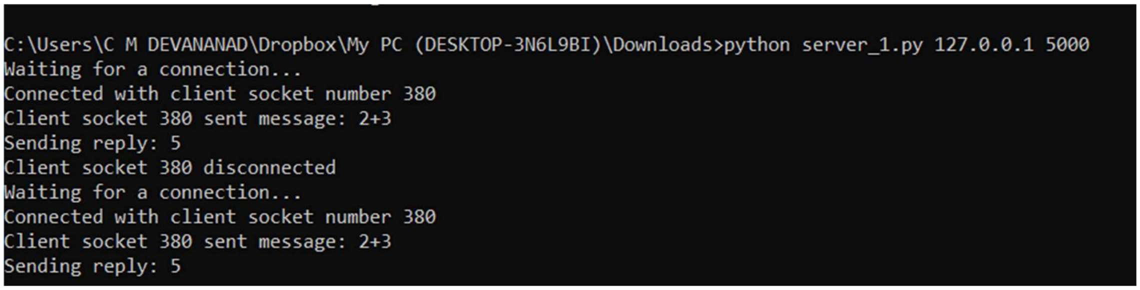
Client2:



```
C:\Users\C M DEVANANAD\Dropbox\My PC (DESKTOP-3N6L9BI)\Downloads>python c_2_main.py 127.0.0.1 5000
Connected to server
Please enter the message to the server: 2+3
```

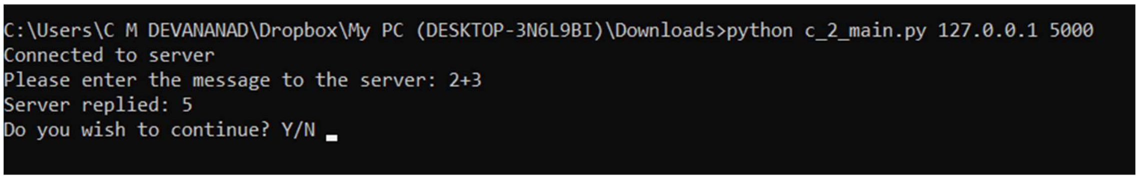
Server doesn’t give any answer until existed server closes.

server response:



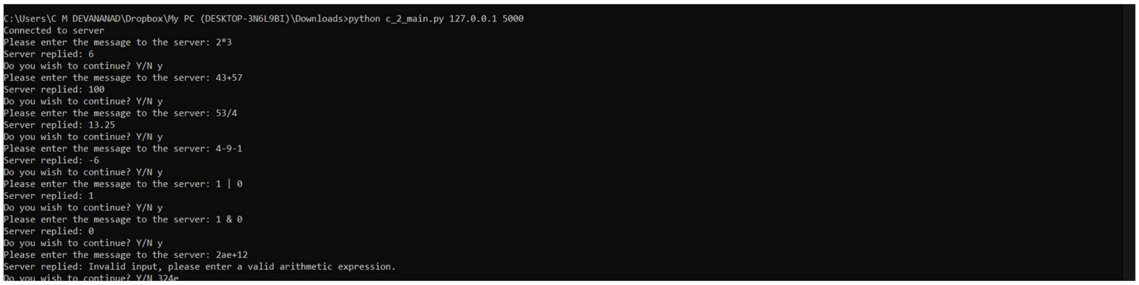
```
C:\Users\C M DEVANANAD\Dropbox\My PC (DESKTOP-3N6L9BI)\Downloads>python server_1.py 127.0.0.1 5000
Waiting for a connection...
Connected with client socket number 380
Client socket 380 sent message: 2+3
Sending reply: 5
Client socket 380 disconnected
Waiting for a connection...
Connected with client socket number 380
Client socket 380 sent message: 2+3
Sending reply: 5
```

client2:



```
C:\Users\C M DEVANANAD\Dropbox\My PC (DESKTOP-3N6L9BI)\Downloads>python c_2_main.py 127.0.0.1 5000
Connected to server
Please enter the message to the server: 2+3
Server replied: 5
Do you wish to continue? Y/N _
```

Results of the different expression:



```
C:\Users\C M DEVANANAD\Dropbox\My PC (DESKTOP-3N6L9BI)\Downloads>python c_2_main.py 127.0.0.1 5000
Connected to server
Please enter the message to the server: 2*3
Server replied: 6
Do you wish to continue? Y/N y
Please enter the message to the server: 43+57
Server replied: 100
Do you wish to continue? Y/N y
Please enter the message to the server: 53/4
Server replied: 13.25
Do you wish to continue? Y/N y
Please enter the message to the server: 4-9-1
Server replied: -6
Do you wish to continue? Y/N y
Please enter the message to the server: 1 | 0
Server replied: 1
Do you wish to continue? Y/N y
Please enter the message to the server: 1 & 0
Server replied: 0
Do you wish to continue? Y/N y
Please enter the message to the server: 2ae+12
Server replied: Invalid input, please enter a valid arithmetic expression.
Do you wish to continue? Y/N 324e
```

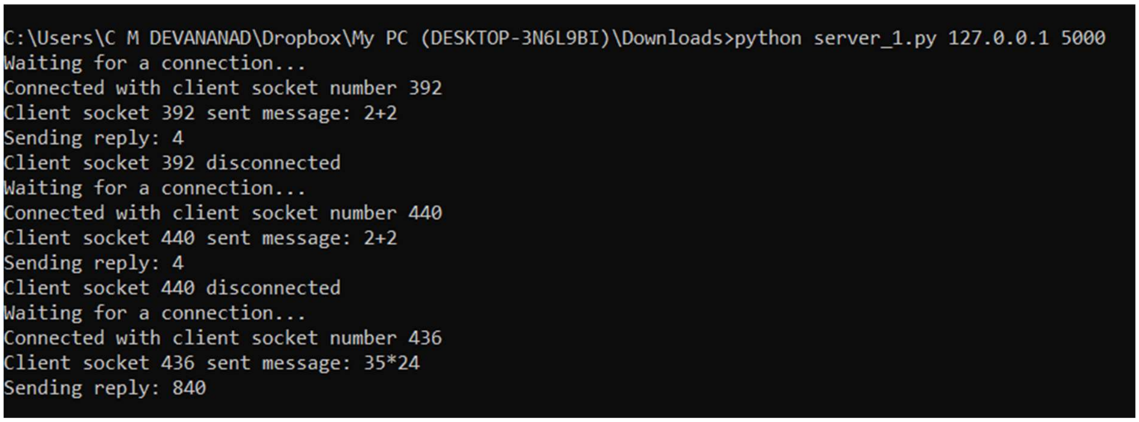
<u>Server2:</u>

Step 1:
Run the server file with command:
python server2.py 127.0.0.1 5000

Step2 :
Run the client file with command in different terminal
python client.py 127.0.0.1 5000

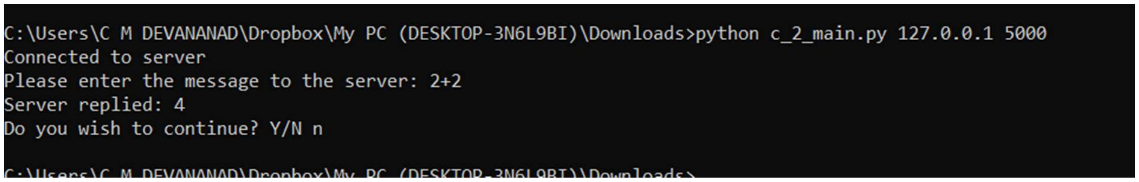
Since server2 can handle multiple clients, we can run 2 or more clients and all clients are connected to the server2.py without any condition and sends the result of the arithmetic expression.

Server response :



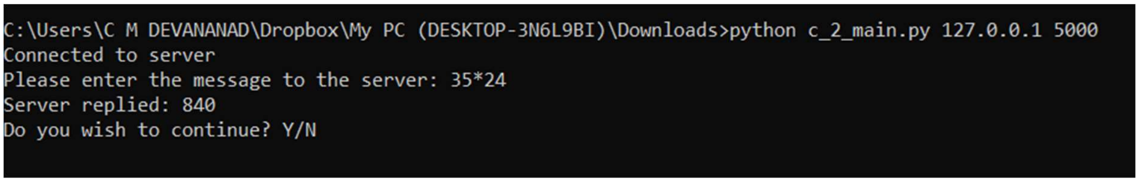
```
C:\Users\C M DEVANANAD\Dropbox\My PC (DESKTOP-3N6L9BI)\Downloads>python server_1.py 127.0.0.1 5000
Waiting for a connection...
Connected with client socket number 392
Client socket 392 sent message: 2+2
Sending reply: 4
Client socket 392 disconnected
Waiting for a connection...
Connected with client socket number 440
Client socket 440 sent message: 2+2
Sending reply: 4
Client socket 440 disconnected
Waiting for a connection...
Connected with client socket number 436
Client socket 436 sent message: 35*24
Sending reply: 840
```

Client1:



```
C:\Users\C M DEVANANAD\Dropbox\My PC (DESKTOP-3N6L9BI)\Downloads>python c_2_main.py 127.0.0.1 5000
Connected to server
Please enter the message to the server: 2+2
Server replied: 4
Do you wish to continue? Y/N n

C:\Users\C M DEVANANAD\Dropbox\My PC (DESKTOP-3N6L9BI)\Downloads>
```

Client2:

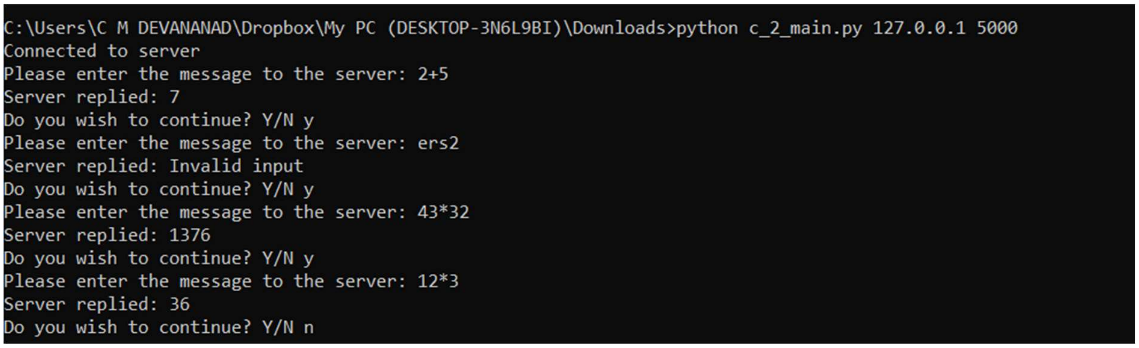


```
C:\Users\C M DEVANANAD\Dropbox\My PC (DESKTOP-3N6L9BI)\Downloads>python c_2_main.py 127.0.0.1 5000
Connected to server
Please enter the message to the server: 35*24
Server replied: 840
Do you wish to continue? Y/N
```

We can check in the server that it connected to both client at a time and server sent the corresponding results.

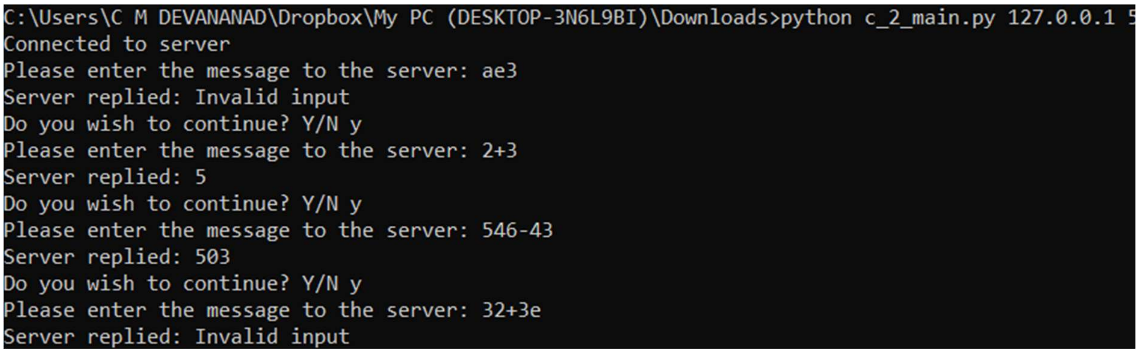
Results of different inputs:

Client1:



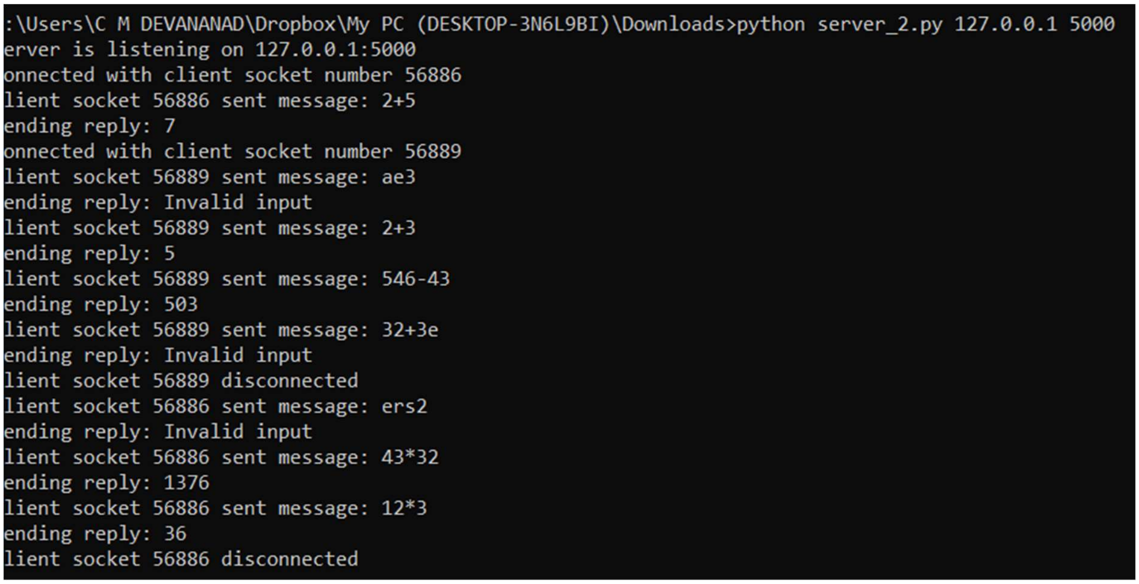
```
C:\Users\C M DEVANANAD\Dropbox\My PC (DESKTOP-3N6L9BI)\Downloads>python c_2_main.py 127.0.0.1 5000
Connected to server
Please enter the message to the server: 2+5
Server replied: 7
Do you wish to continue? Y/N y
Please enter the message to the server: ers2
Server replied: Invalid input
Do you wish to continue? Y/N y
Please enter the message to the server: 43*32
Server replied: 1376
Do you wish to continue? Y/N y
Please enter the message to the server: 12*3
Server replied: 36
Do you wish to continue? Y/N n
```

client2:



```
C:\Users\C M DEVANANAD\Dropbox\My PC (DESKTOP-3N6L9BI)\Downloads>python c_2_main.py 127.0.0.1 5
Connected to server
Please enter the message to the server: ae3
Server replied: Invalid input
Do you wish to continue? Y/N y
Please enter the message to the server: 2+3
Server replied: 5
Do you wish to continue? Y/N y
Please enter the message to the server: 546-43
Server replied: 503
Do you wish to continue? Y/N y
Please enter the message to the server: 32+3e
Server replied: Invalid input
```

Server2 response:



```
:\Users\C M DEVANANAD\Dropbox\My PC (DESKTOP-3N6L9BI)\Downloads>python server_2.py 127.0.0.1 5000
erver is listening on 127.0.0.1:5000
onnected with client socket number 56886
lient socket 56886 sent message: 2+5
ending reply: 7
onnected with client socket number 56889
lient socket 56889 sent message: ae3
ending reply: Invalid input
lient socket 56889 sent message: 2+3
ending reply: 5
lient socket 56889 sent message: 546-43
ending reply: 503
lient socket 56889 sent message: 32+3e
ending reply: Invalid input
lient socket 56889 disconnected
lient socket 56886 sent message: ers2
ending reply: Invalid input
lient socket 56886 sent message: 43*32
ending reply: 1376
lient socket 56886 sent message: 12*3
ending reply: 36
lient socket 56886 disconnected
```

z

Server3:

The "server3.py" program is a single process server that uses the "select" method to handle multiple clients concurrently. It creates a TCP/IP socket and binds it to a specific address and port. Then, it listens for incoming connections and uses select to multiplex between sockets. When a new client connects, it accepts the connection and adds the client socket to the list of sockets to select from. When data is received from a client socket, it evaluates an arithmetic expression sent by the client and sends the result back to the client. If a client disconnects, the server removes the client socket from the list of sockets and closes the socket. Finally, if there is an exceptional condition on a socket, it removes the socket from the list of sockets and closes the socket.

Step 1:
Run the server file with command:
python server3.py 127.0.0.1 5000

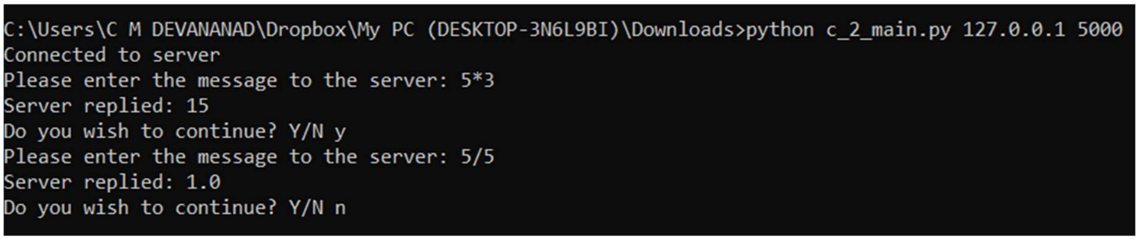
Step2 :
Run the client file with command in different terminal
python client.py 127.0.0.1 5000

Even though server3 is a single process server but it will select the client at which should send the output.

Which is same as server2 but it is a select based server.

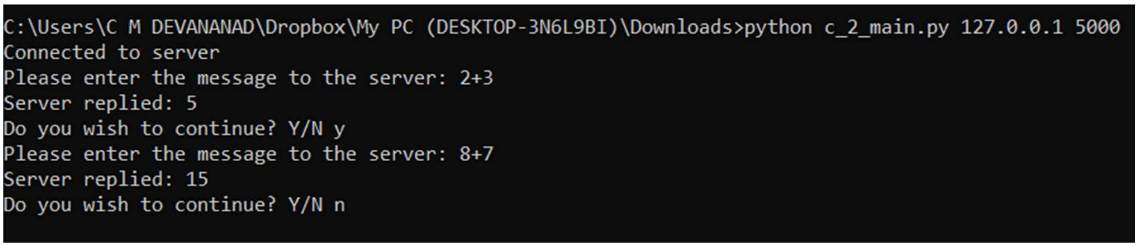
Outputs:

Client1:



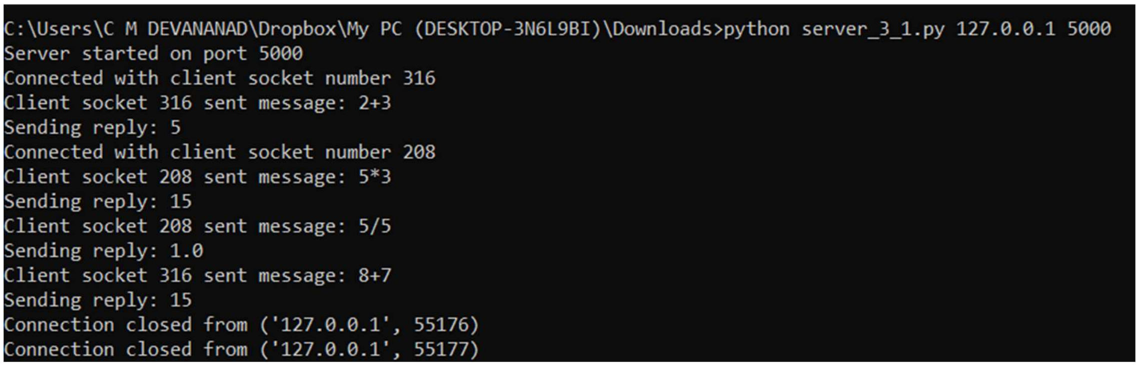
```
C:\Users\C M DEVANANAD\Dropbox\My PC (DESKTOP-3N6L9BI)\Downloads>python c_2_main.py 127.0.0.1 5000
Connected to server
Please enter the message to the server: 5*3
Server replied: 15
Do you wish to continue? Y/N y
Please enter the message to the server: 5/5
Server replied: 1.0
Do you wish to continue? Y/N n
```

Client2:



```
C:\Users\C M DEVANANAD\Dropbox\My PC (DESKTOP-3N6L9BI)\Downloads>python c_2_main.py 127.0.0.1 5000
Connected to server
Please enter the message to the server: 2+3
Server replied: 5
Do you wish to continue? Y/N y
Please enter the message to the server: 8+7
Server replied: 15
Do you wish to continue? Y/N n
```

Server3:



```
C:\Users\C M DEVANANAD\Dropbox\My PC (DESKTOP-3N6L9BI)\Downloads>python server_3_1.py 127.0.0.1 5000
Server started on port 5000
Connected with client socket number 316
Client socket 316 sent message: 2+3
Sending reply: 5
Connected with client socket number 208
Client socket 208 sent message: 5*3
Sending reply: 15
Client socket 208 sent message: 5/5
Sending reply: 1.0
Client socket 316 sent message: 8+7
Sending reply: 15
Connection closed from ('127.0.0.1', 55176)
Connection closed from ('127.0.0.1', 55177)
```

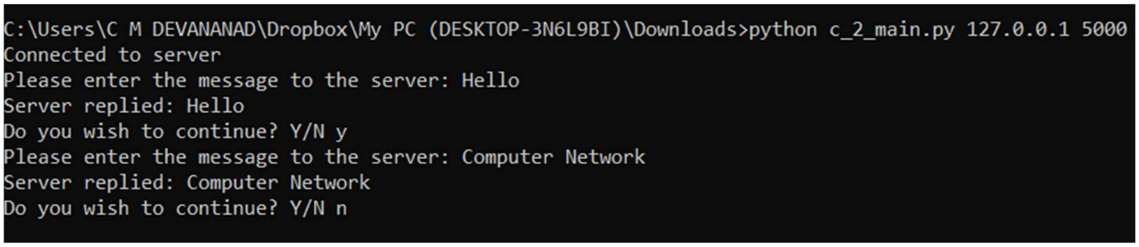
<u>Server4:</u>

Step 1:
Run the server file with command:
python server4.py 127.0.0.1 5000

Step2 :
Run the client file with command in different terminal
python client.py 127.0.0.1 5000

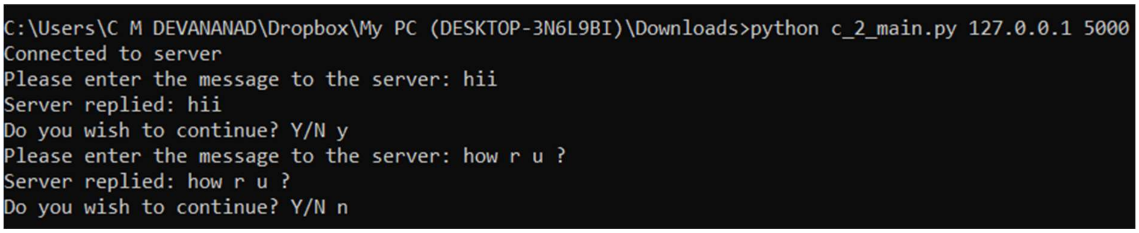
Server 4 is same as server3 but here server should send the output as a input data.

Client1:



```
C:\Users\C M DEVANANAD\Dropbox\My PC (DESKTOP-3N6L9BI)\Downloads>python c_2_main.py 127.0.0.1 5000
Connected to server
Please enter the message to the server: Hello
Server replied: Hello
Do you wish to continue? Y/N y
Please enter the message to the server: Computer Network
Server replied: Computer Network
Do you wish to continue? Y/N n
```

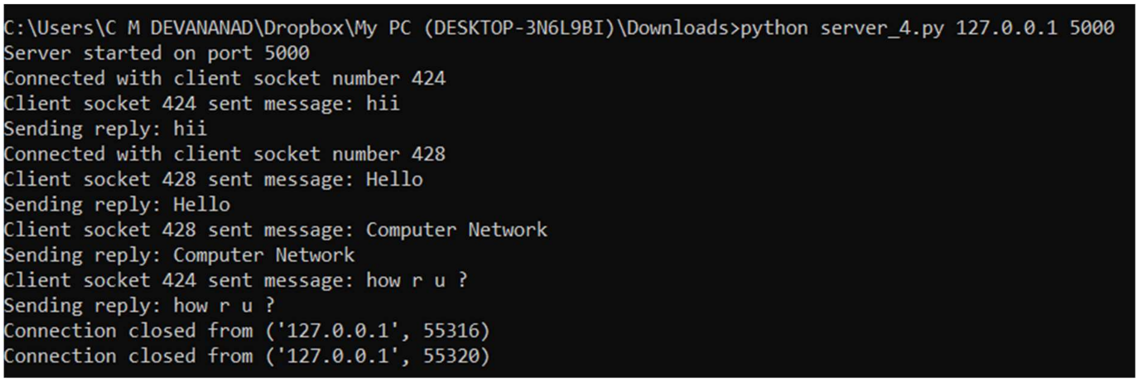
Client2:



```
C:\Users\C M DEVANANAD\Dropbox\My PC (DESKTOP-3N6L9BI)\Downloads>python c_2_main.py 127.0.0.1 5000
Connected to server
Please enter the message to the server: hii
Server replied: hii
Do you wish to continue? Y/N y
Please enter the message to the server: how r u ?
Server replied: how r u ?
Do you wish to continue? Y/N n
```

z

Server response:



```
C:\Users\C M DEVANANAD\Dropbox\My PC (DESKTOP-3N6L9BI)\Downloads>python server_4.py 127.0.0.1 5000
Server started on port 5000
Connected with client socket number 424
Client socket 424 sent message: hii
Sending reply: hii
Connected with client socket number 428
Client socket 428 sent message: Hello
Sending reply: Hello
Client socket 428 sent message: Computer Network
Sending reply: Computer Network
Client socket 424 sent message: how r u ?
Sending reply: how r u ?
Connection closed from ('127.0.0.1', 55316)
Connection closed from ('127.0.0.1', 55320)
```

The server program "server4.py" is an echo server that receives messages from clients and echoes them back to the same client. It is a single process server that uses the "select" method to handle multiple clients concurrently. The server creates a TCP/IP socket, binds it to a specific address and port, and listens for incoming connections. It uses the select() function to multiplex between sockets, and handles sockets with incoming data, new connections, and exceptional conditions. It parses the received message, sends the result back to the client, and closes the connection when the client disconnects. The server displays the client's message and the sent reply on the console.

z