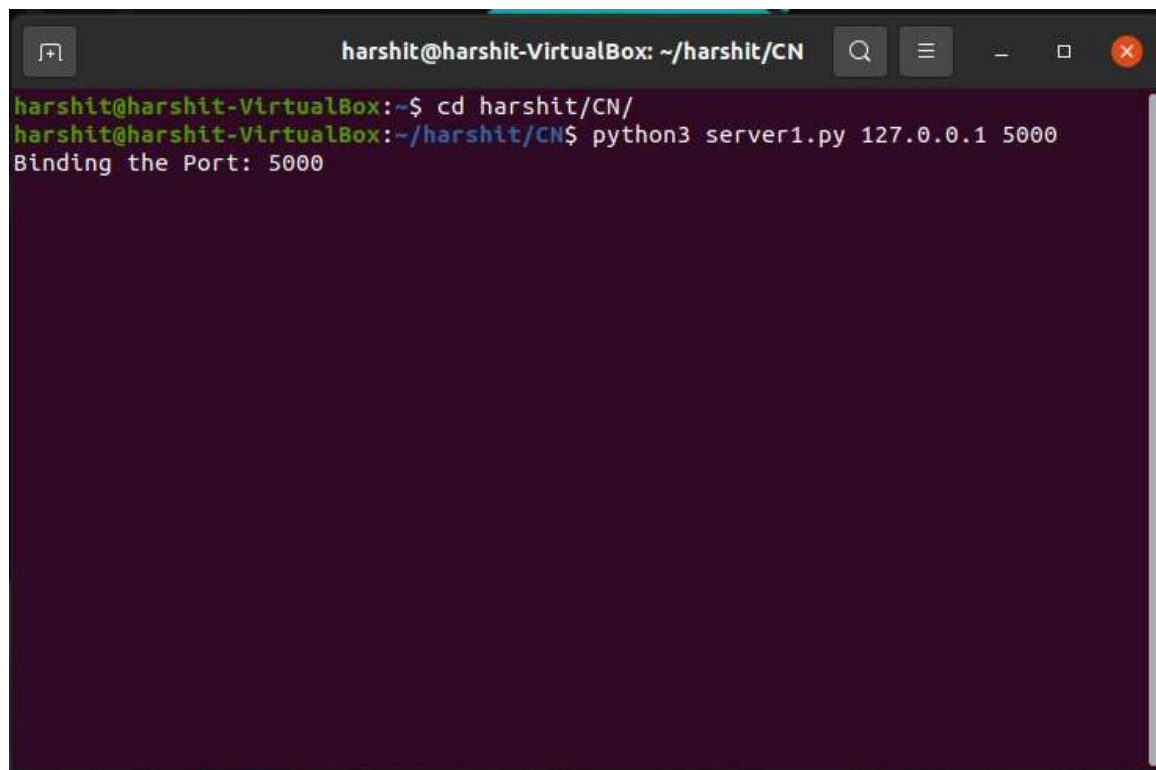


REPORT

Instructions to run the code locally:

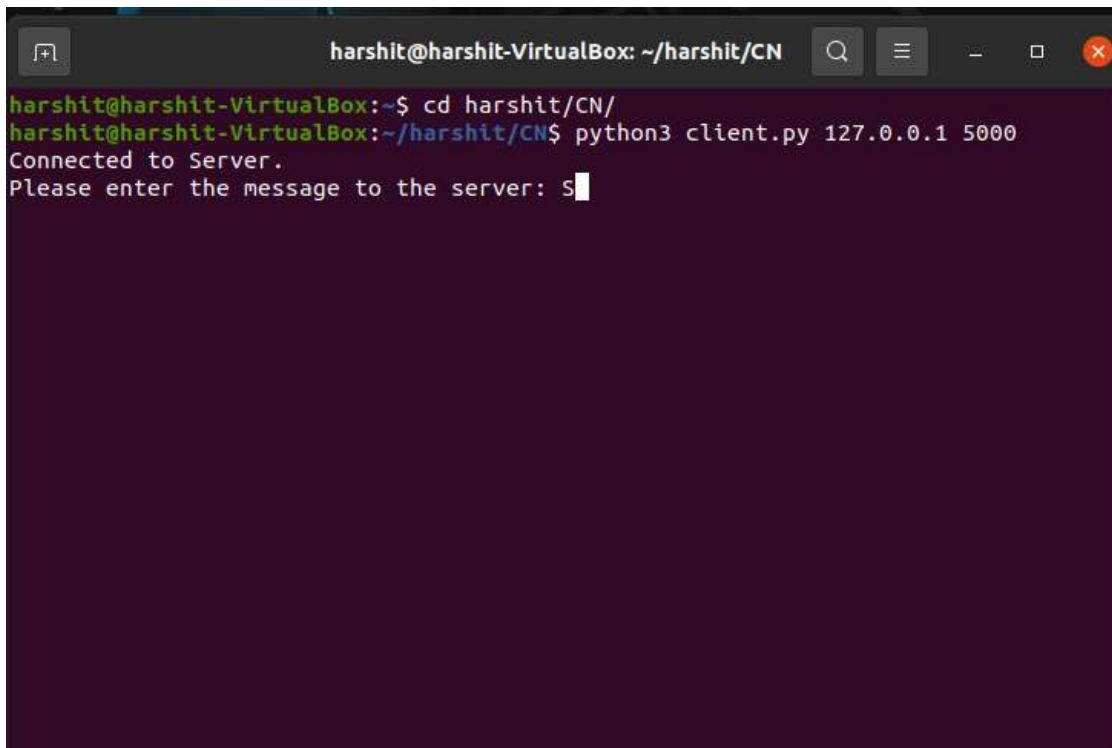
1. First, we will run the terminal and go to the directory where the server code is present.
2. Type in the following command in the terminal – `python3 server1.py IP PORT`.
3. We will use localhost i.e. 127.0.0.1 for IP and 5000 for PORT.

A screenshot of a terminal window titled 'harshit@harshit-VirtualBox: ~/harshit/CN'. The terminal shows the following commands and output:

```
harshit@harshit-VirtualBox:~$ cd harshit/CN/  
harshit@harshit-VirtualBox:~/harshit/CN$ python3 server1.py 127.0.0.1 5000  
Binding the Port: 5000
```

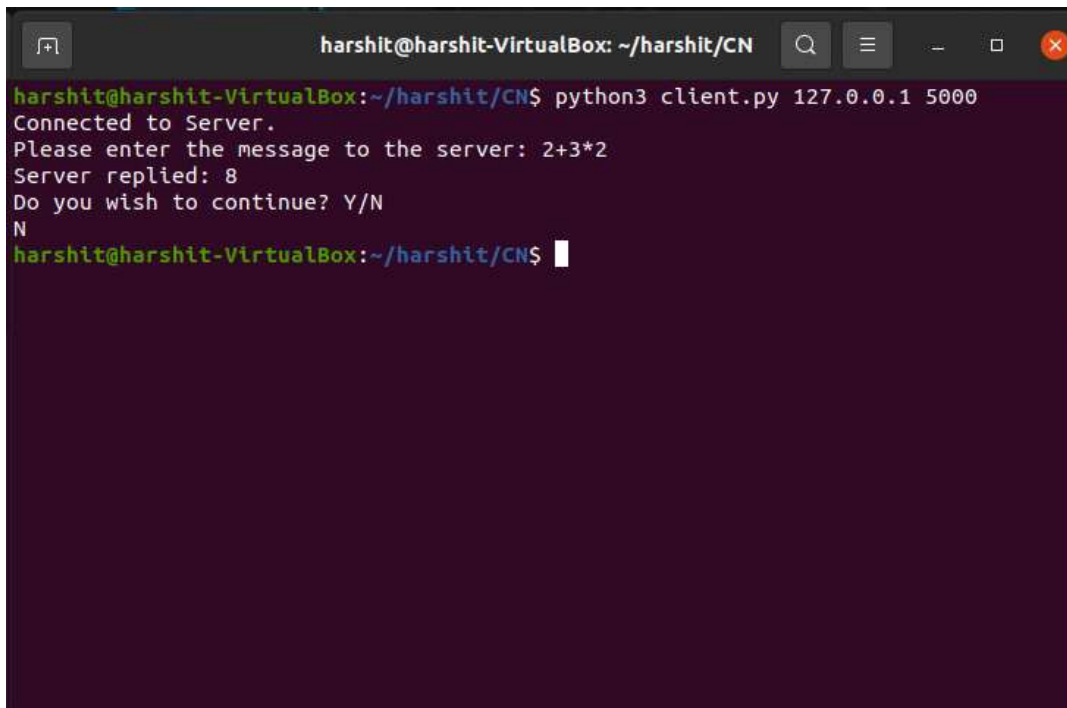
The terminal has a dark purple background and a light gray border. The window title bar includes standard Linux window controls (minimize, maximize, close) and a search icon.

4. Open another terminal and open the directory where the client.py code is present.
5. Enter the following command – `python3 client.py 127.0.0.1 5000` (the IP and port used by the server).



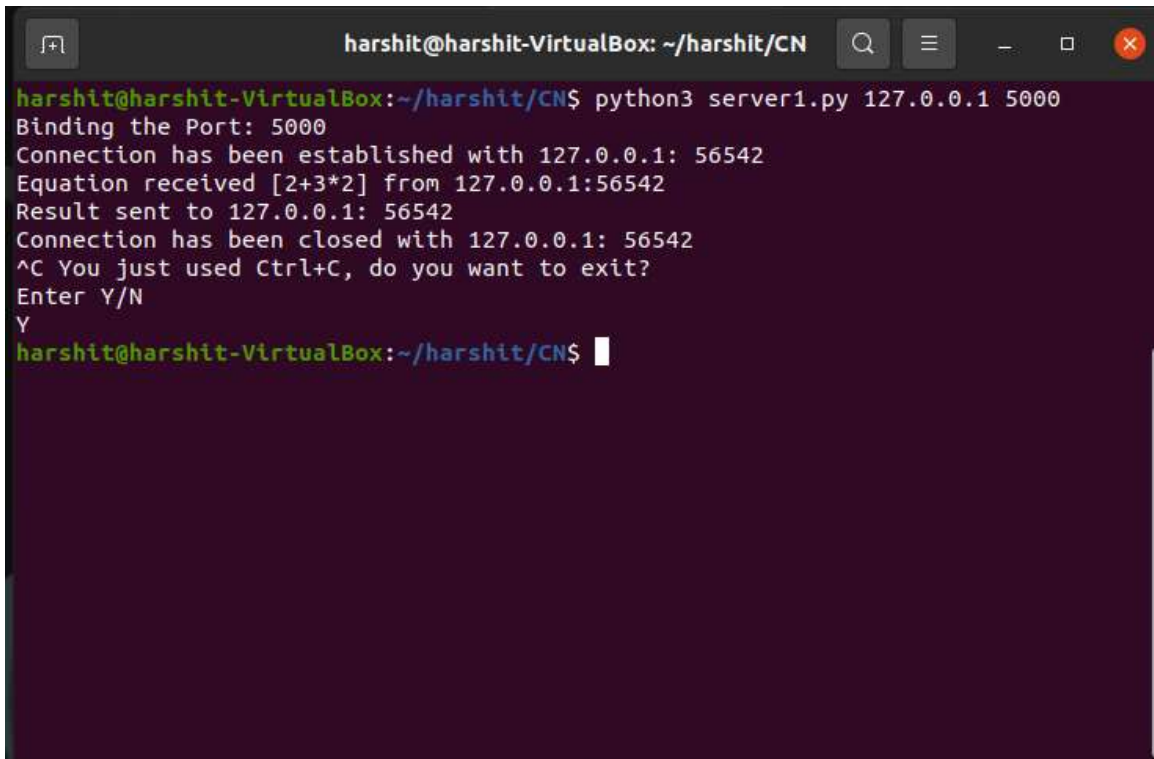
```
harshit@harshit-VirtualBox: ~/harshit/CN
harshit@harshit-VirtualBox:~$ cd harshit/CN/
harshit@harshit-VirtualBox:~/harshit/CN$ python3 client.py 127.0.0.1 5000
Connected to Server.
Please enter the message to the server: s
```

6. After this, the server-side terminal should prompt that "Connection has been established with 127.0.0.1: PORT"
7. Now the client can send requests to the server. Depending on the type of server single or multiple clients can be connected
8. In order to exit the client.py, after entering the query the program asks you whether you want to continue or not. Enter N to exit.



```
harshit@harshit-VirtualBox: ~/harshit/CN
harshit@harshit-VirtualBox:~/harshit/CN$ python3 client.py 127.0.0.1 5000
Connected to Server.
Please enter the message to the server: 2+3*2
Server replied: 8
Do you wish to continue? Y/N
N
harshit@harshit-VirtualBox:~/harshit/CN$
```

9. To exit server.py press ctrl+C and y get a prompt whether you want to exit or not. Press Y.

A terminal window titled 'harshit@harshit-VirtualBox: ~/harshit/CN' with standard window controls. The terminal shows the execution of 'python3 server1.py 127.0.0.1 5000'. The output includes: 'Binding the Port: 5000', 'Connection has been established with 127.0.0.1: 56542', 'Equation received [2+3*2] from 127.0.0.1:56542', 'Result sent to 127.0.0.1: 56542', 'Connection has been closed with 127.0.0.1: 56542', and a prompt '^C You just used Ctrl+C, do you want to exit? Enter Y/N'. The user has entered 'Y', and the prompt has returned to 'harshit@harshit-VirtualBox: ~/harshit/CN\$'.

```
harshit@harshit-VirtualBox: ~/harshit/CN$ python3 server1.py 127.0.0.1 5000
Binding the Port: 5000
Connection has been established with 127.0.0.1: 56542
Equation received [2+3*2] from 127.0.0.1:56542
Result sent to 127.0.0.1: 56542
Connection has been closed with 127.0.0.1: 56542
^C You just used Ctrl+C, do you want to exit?
Enter Y/N
Y
harshit@harshit-VirtualBox: ~/harshit/CN$
```

Instructions to run the code on different systems:

1. In step 3 of locally use ip of your own device and any free port and in step 5 use the ip of server device.

Screenshot of output on testcase 1:

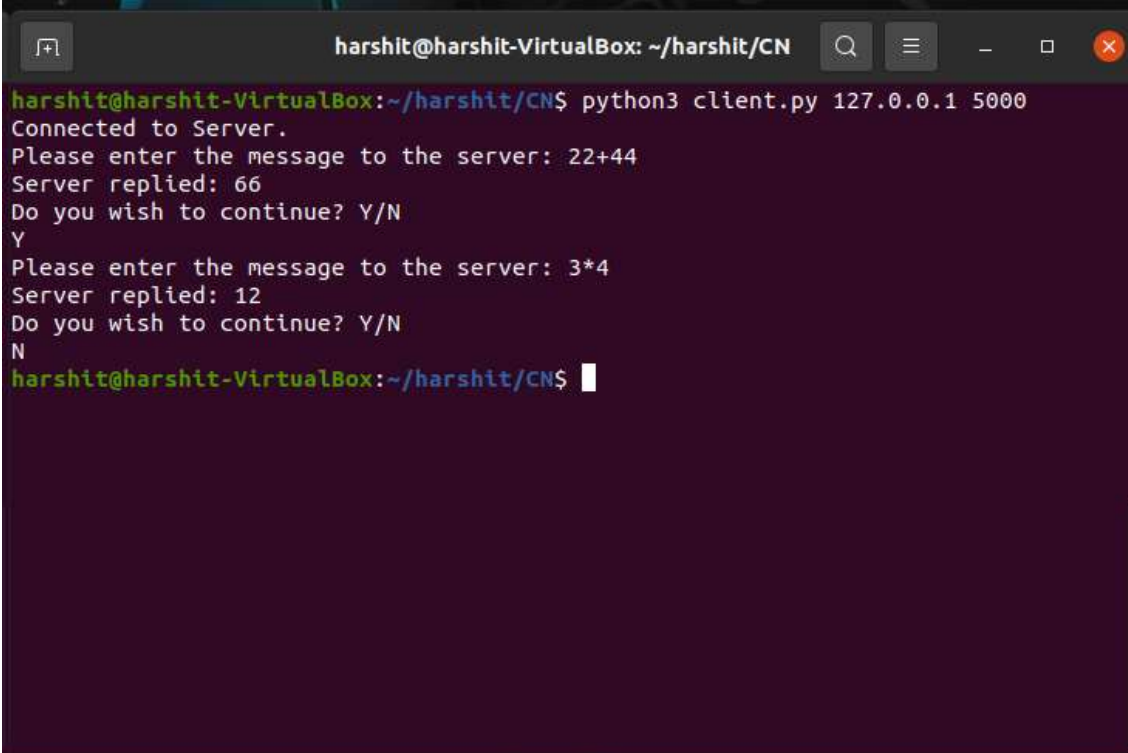
Testcase 1:

```
$ python client.py 127.0.0.1 5000
Connected to server
Please enter the message to the server: 22 + 44
Server replied: 66
Please enter the message to the server: 3 * 4
Server replied: 12
```

In this testcase we assume that only 1 client is allowed to be connected to the server .

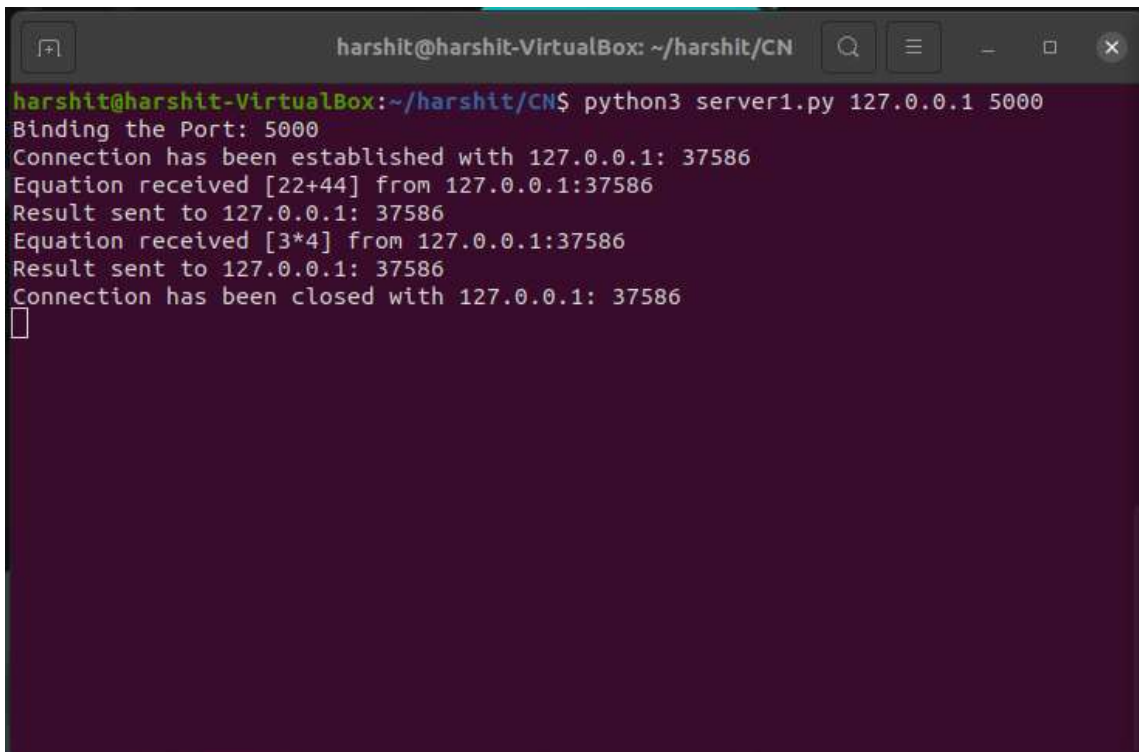
Output:

Client:

A screenshot of a terminal window titled "harshit@harshit-VirtualBox: ~/harshit/CN". The terminal shows the execution of a Python script "client.py" with arguments "127.0.0.1 5000". The output of the script is as follows: "Connected to Server.", "Please enter the message to the server: 22+44", "Server replied: 66", "Do you wish to continue? Y/N", "Y", "Please enter the message to the server: 3*4", "Server replied: 12", "Do you wish to continue? Y/N", "N". The prompt "harshit@harshit-VirtualBox: ~/harshit/CN\$" is shown at the bottom of the terminal.

```
harshit@harshit-VirtualBox: ~/harshit/CN$ python3 client.py 127.0.0.1 5000
Connected to Server.
Please enter the message to the server: 22+44
Server replied: 66
Do you wish to continue? Y/N
Y
Please enter the message to the server: 3*4
Server replied: 12
Do you wish to continue? Y/N
N
harshit@harshit-VirtualBox: ~/harshit/CN$
```

Server:

A screenshot of a terminal window titled 'harshit@harshit-VirtualBox: ~/harshit/CN'. The terminal shows the execution of a Python script 'server1.py' with arguments '127.0.0.1' and '5000'. The output indicates that the server is binding to port 5000 and successfully establishes a connection with a client at 127.0.0.1:37586. It then receives two equations: '[22+44]' and '[3*4]', sends the results '66' and '12' respectively, and finally closes the connection.

```
harshit@harshit-VirtualBox:~/harshit/CN$ python3 server1.py 127.0.0.1 5000
Binding the Port: 5000
Connection has been established with 127.0.0.1: 37586
Equation received [22+44] from 127.0.0.1:37586
Result sent to 127.0.0.1: 37586
Equation received [3*4] from 127.0.0.1:37586
Result sent to 127.0.0.1: 37586
Connection has been closed with 127.0.0.1: 37586
█
```

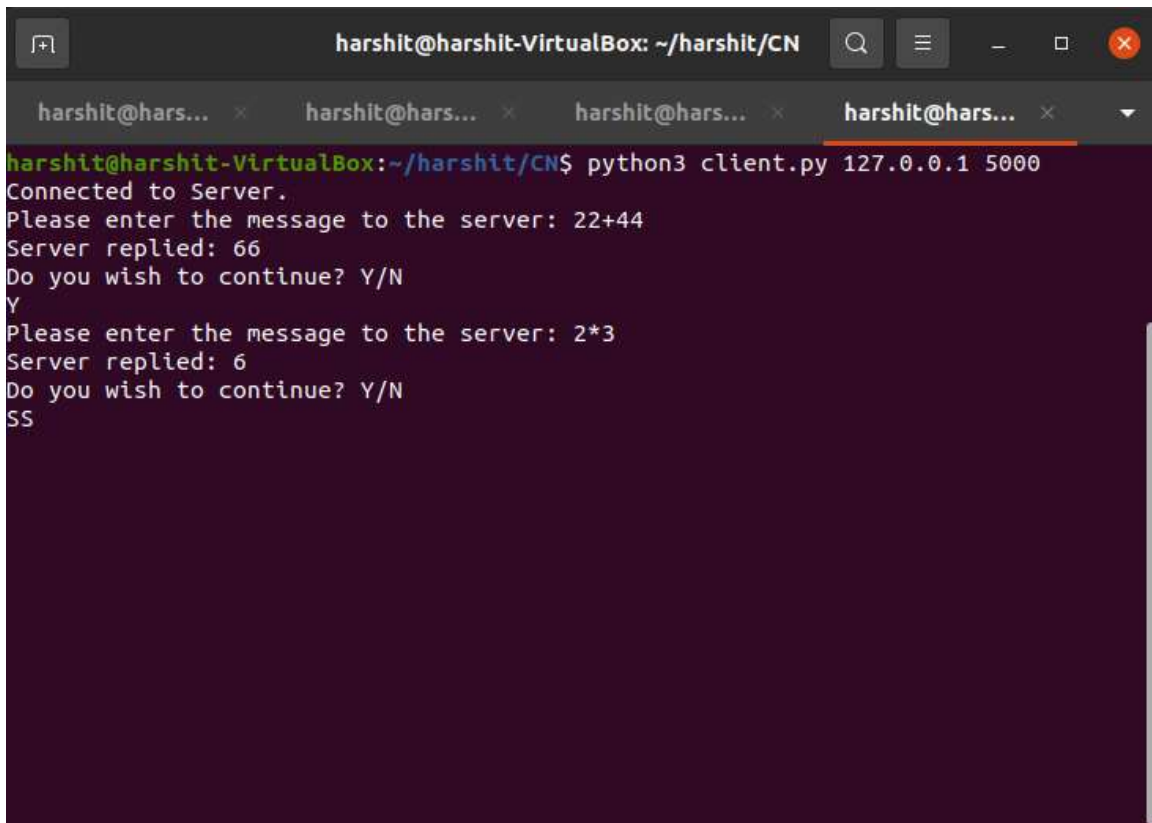
Testcase2:

```
$ python server1.py 127.0.0.1 5000
Connected with client socket number 4
Client socket 4 sent message: 22 + 44
Sending reply: 66
Client socket 4 sent message: 3 * 4
Sending reply: 12
```

In this testcase we assume that 4 clients are connected to the server and that these queries are from the 4th client, hence “socket 4”.

Output:

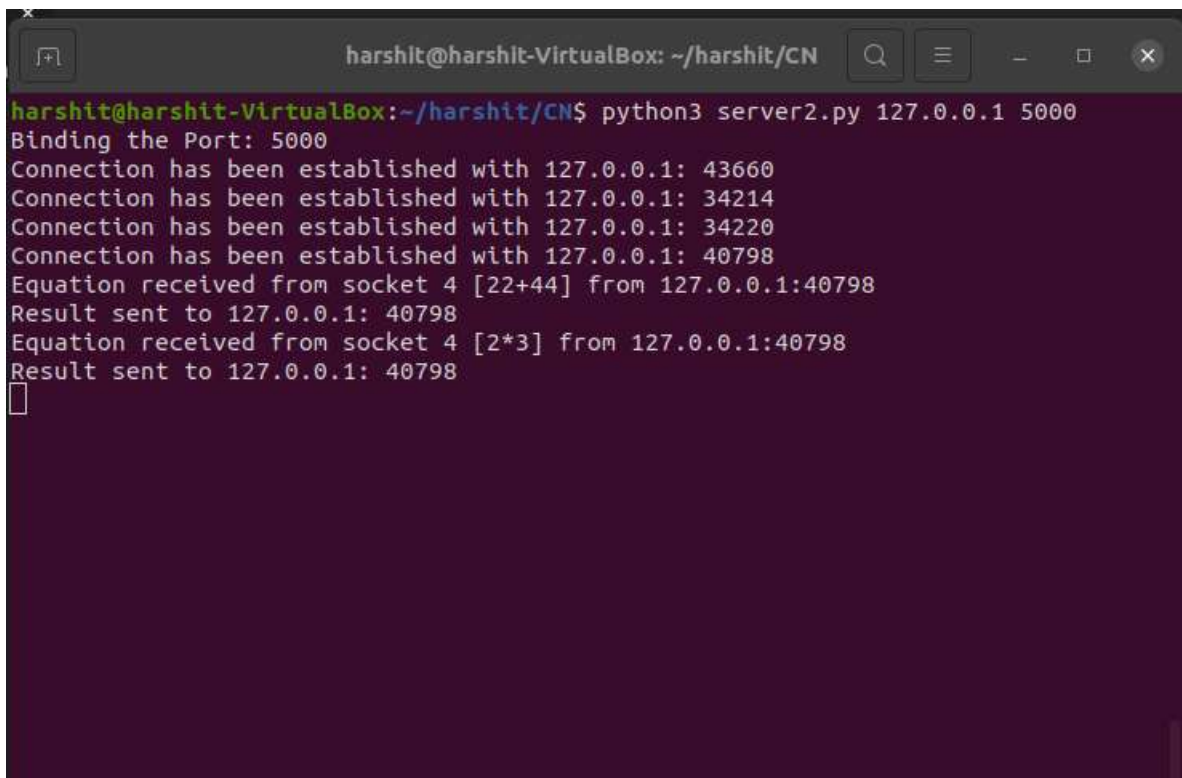
Client:

A terminal window titled 'harshit@harshit-VirtualBox: ~/harshit/CN' with four tabs. The active tab shows the execution of 'python3 client.py 127.0.0.1 5000'. The output shows a successful connection to the server, followed by two interactions: first, the user enters '22+44' and the server replies '66'; second, the user enters '2*3' and the server replies '6'. The user then enters 'Y' to continue and 'SS' to stop. The terminal has a dark purple background and green text.

```
harshit@harshit-VirtualBox: ~/harshit/CN$ python3 client.py 127.0.0.1 5000
Connected to Server.
Please enter the message to the server: 22+44
Server replied: 66
Do you wish to continue? Y/N
Y
Please enter the message to the server: 2*3
Server replied: 6
Do you wish to continue? Y/N
SS
```

Here we can see that there 4 terminals open, each terminal has one client open and the visible terminal is the client no. 4.

Server:

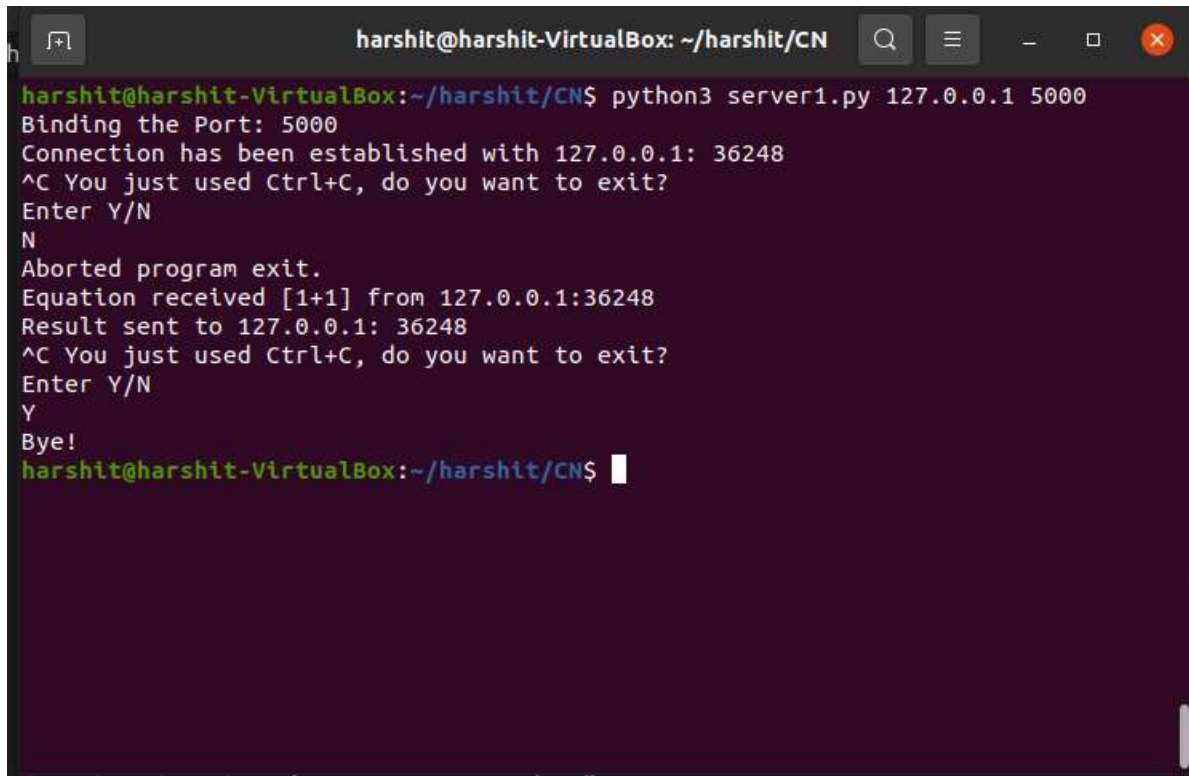
A terminal window titled 'harshit@harshit-VirtualBox: ~/harshit/CN' showing the execution of 'python3 server2.py 127.0.0.1 5000'. The output shows the server binding to port 5000 and establishing four connections with clients at 127.0.0.1 on ports 43660, 34214, 34220, and 40798. It then shows the server receiving and responding to two equations from the client at 40798: '[22+44]' and '[2*3]', both with the result '40798'. The terminal has a dark purple background and green text.

```
harshit@harshit-VirtualBox: ~/harshit/CN$ python3 server2.py 127.0.0.1 5000
Binding the Port: 5000
Connection has been established with 127.0.0.1: 43660
Connection has been established with 127.0.0.1: 34214
Connection has been established with 127.0.0.1: 34220
Connection has been established with 127.0.0.1: 40798
Equation received from socket 4 [22+44] from 127.0.0.1:40798
Result sent to 127.0.0.1: 40798
Equation received from socket 4 [2*3] from 127.0.0.1:40798
Result sent to 127.0.0.1: 40798
█
```

We can see that 4 connections are established and the queries obtained are from the 4th client and the server is mentioning this in its output.

Additional Features:

1. Once the user enters Ctrl+C to end the server program, instead of giving errors and terminating, we have done error handling and reconfirm with the user whether they want to exit or not and if yes, exit cleanly and if not then continue with the program execution.

A terminal window titled 'harshit@harshit-VirtualBox: ~/harshit/CN' showing the execution of a Python script. The user runs 'python3 server1.py 127.0.0.1 5000'. The output shows the port binding, a connection from 127.0.0.1:36248, and a received query '[1+1]'. When the user presses Ctrl+C, the program asks 'You just used Ctrl+C, do you want to exit? Enter Y/N'. If the user enters 'N', the program says 'Aborted program exit.' and continues. If the user enters 'Y', the program says 'Bye!' and exits.

```
harshit@harshit-VirtualBox: ~/harshit/CN$ python3 server1.py 127.0.0.1 5000
Binding the Port: 5000
Connection has been established with 127.0.0.1: 36248
^C You just used Ctrl+C, do you want to exit?
Enter Y/N
N
Aborted program exit.
Equation received [1+1] from 127.0.0.1:36248
Result sent to 127.0.0.1: 36248
^C You just used Ctrl+C, do you want to exit?
Enter Y/N
Y
Bye!
harshit@harshit-VirtualBox:~/harshit/CN$
```

Here, we can see that once we enter ctrl+c, the program asks the user whether they want to exit, if the user enters N, then exit is aborted and normal execution is continued.

When the user enters Y, the program exits with a Bye!