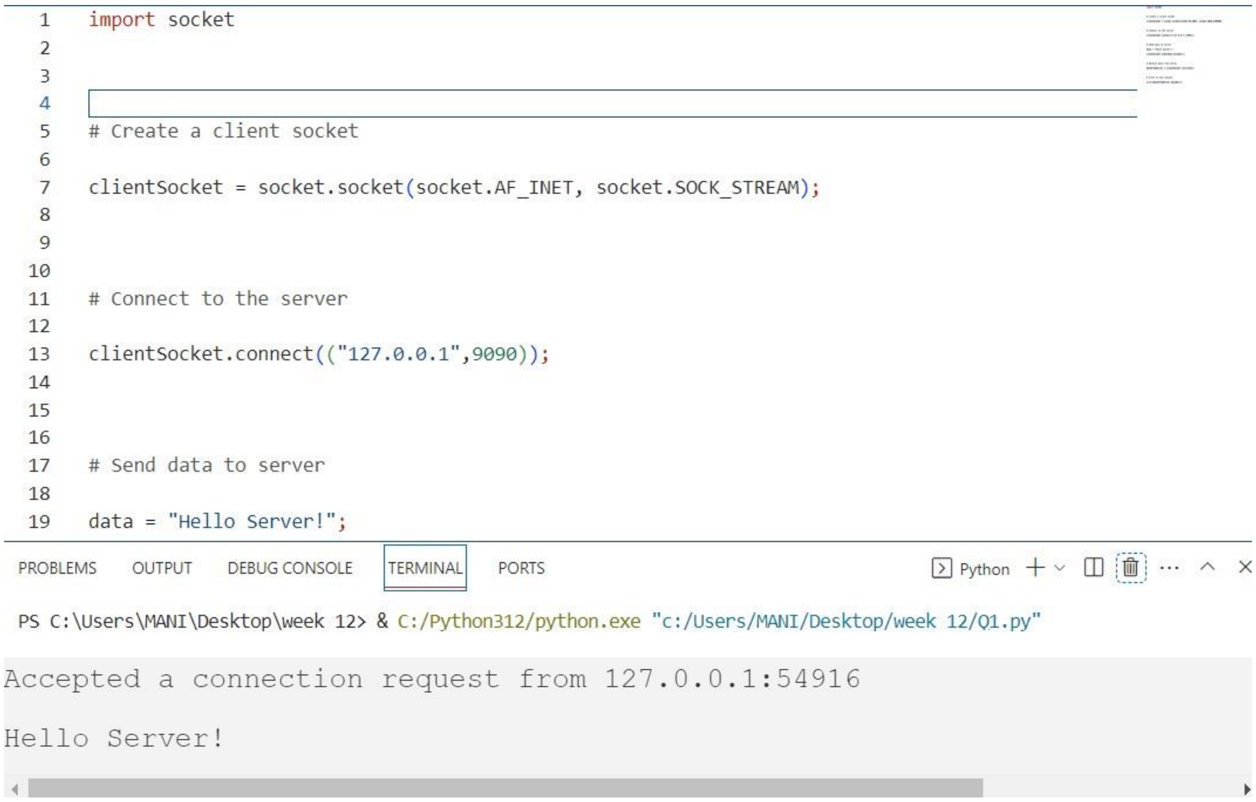
WEEK 12

Implement Python program - TCP/UDP program using Sockets

Q1) Develop a simple Python program of TCP, client that can connect to the server and client can send a "Hello, Server!" message to the server



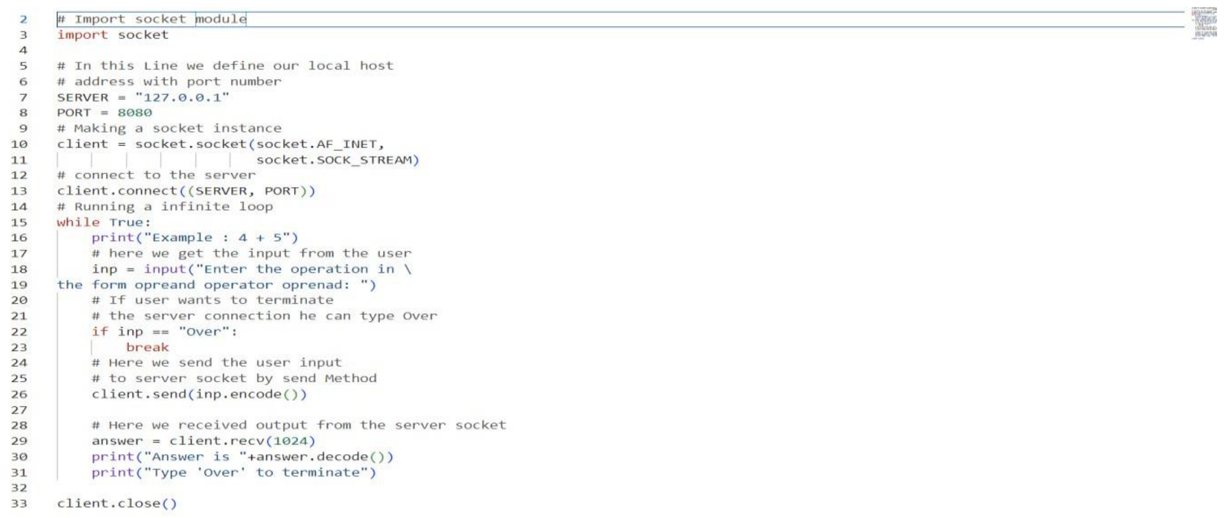
```python
import socket



# Create a client socket

clientSocket = socket.socket(socket.AF_INET, socket.SOCK_STREAM);



# Connect to the server

clientSocket.connect(("127.0.0.1",9090));



# Send data to server

data = "Hello Server!";
```

```
PS C:\Users\MANI\Desktop\week 12> & C:/Python312/python.exe "c:/Users/MANI/Desktop/week 12/Q1.py"
```

```
Accepted a connection request from 127.0.0.1:54916

Hello Server!
```

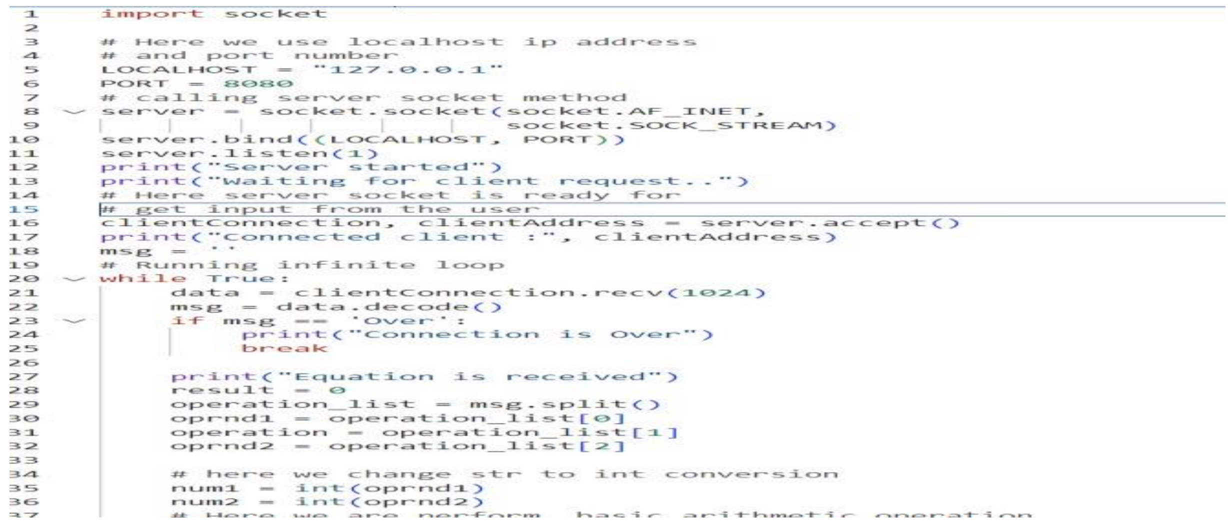
Q2) Develop a Python program that allows the TCP client to send a list of numbers to the server. The server should calculate and return the sum of the numbers to the client.

CLIENT SIDE

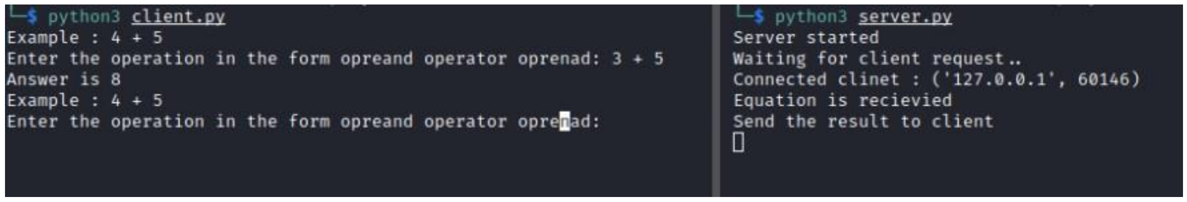


```
# Import socket module
import socket

# In this Line we define our local host
# address with port number
SERVER = "127.0.0.1"
PORT = 8080
# Making a socket instance
client = socket.socket(socket.AF_INET,
                       socket.SOCK_STREAM)
# connect to the server
client.connect((SERVER, PORT))
# Running a infinite loop
while True:
    print("Example : 4 + 5")
    # here we get the input from the user
    inp = input("Enter the operation in \
the form opreand operator oprenad: ")
    # If user wants to terminate
    # the server connection he can type Over
    if inp == "Over":
        break
    # Here we send the user input
    # to server socket by send Method
    client.send(inp.encode())

    # Here we received output from the server socket
    answer = client.recv(1024)
    print("Answer is "+answer.decode())
    print("Type 'Over' to terminate")

client.close()
```

SERVER SIDE



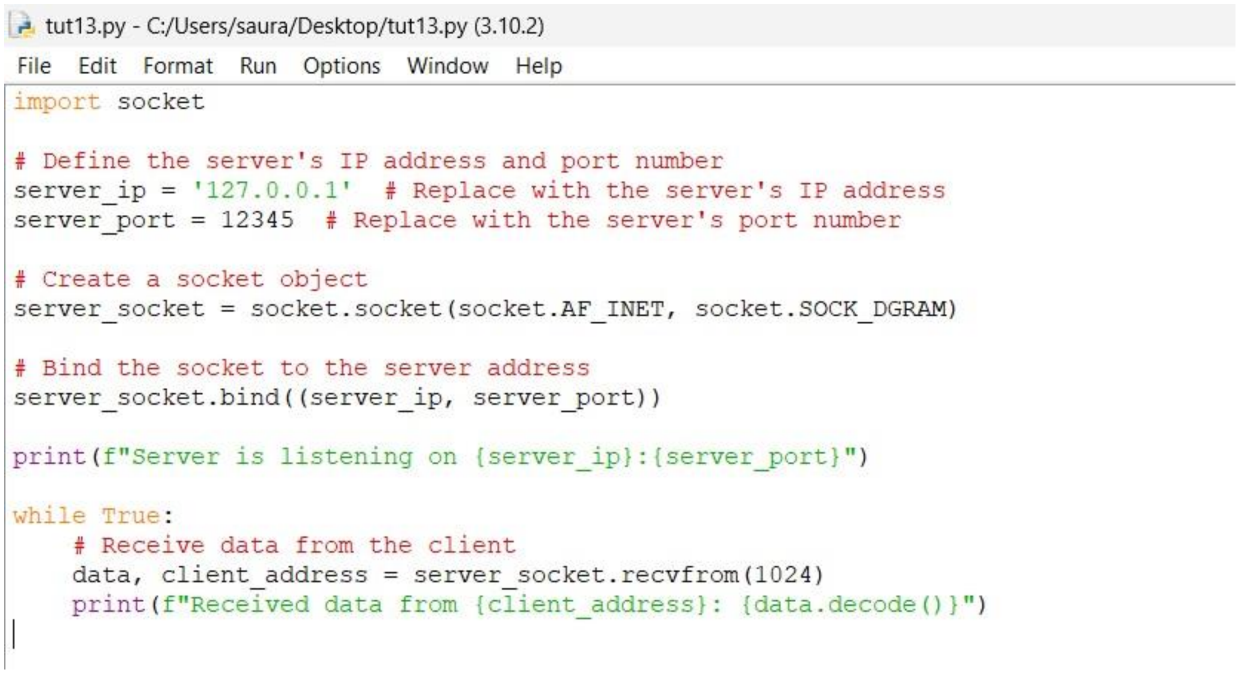
```
import socket

# Here we use localhost ip address
# and port number
LOCALHOST = "127.0.0.1"
PORT = 8080
# calling server socket method
server = socket.socket(socket.AF_INET,
                       socket.SOCK_STREAM)
server.bind((LOCALHOST, PORT))
server.listen(1)
print("Server started")
print("Waiting for client request..")
# Here server socket is ready for
# get input from the user
clientConnection, clientAddress = server.accept()
print("Connected client :", clientAddress)
msg = ''
# Running infinite loop
while True:
    data = clientConnection.recv(1024)
    msg = data.decode()
    if msg == 'Over':
        print("Connection is Over")
        break

    print("Equation is received")
    result = 0
    operation_list = msg.split()
    oprnd1 = operation_list[0]
    operation = operation_list[1]
    oprnd2 = operation_list[2]

    # here we change str to int conversion
    num1 = int(oprnd1)
    num2 = int(oprnd2)
    # Here we are perform  basic arithmetic operation
```



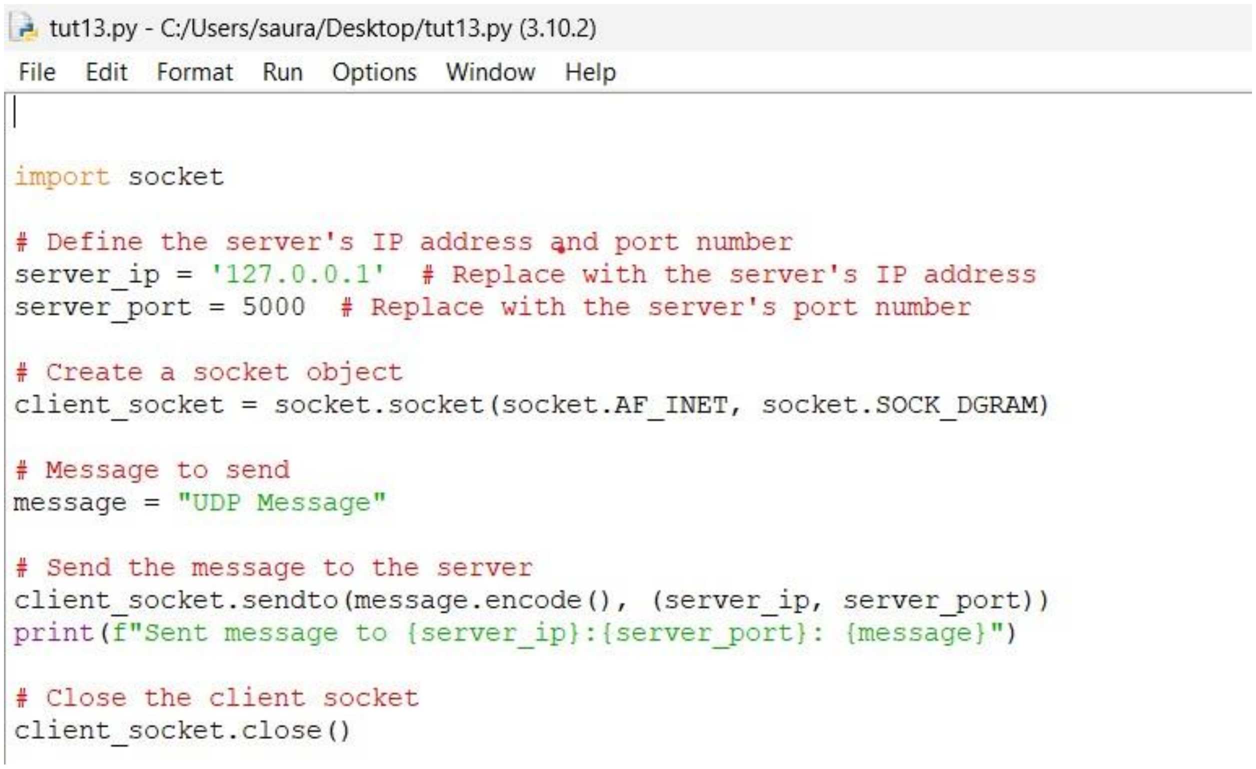
```
└─$ python3 client.py
Example : 4 + 5
Enter the operation in the form opreand operator oprenad: 3 + 5
Answer is 8
Example : 4 + 5
Enter the operation in the form opreand operator oprenad:
```

```
└─$ python3 server.py
Server started
Waiting for client request..
Connected clinet : ('127.0.0.1', 60146)
Equation is recievied
Send the result to client
```

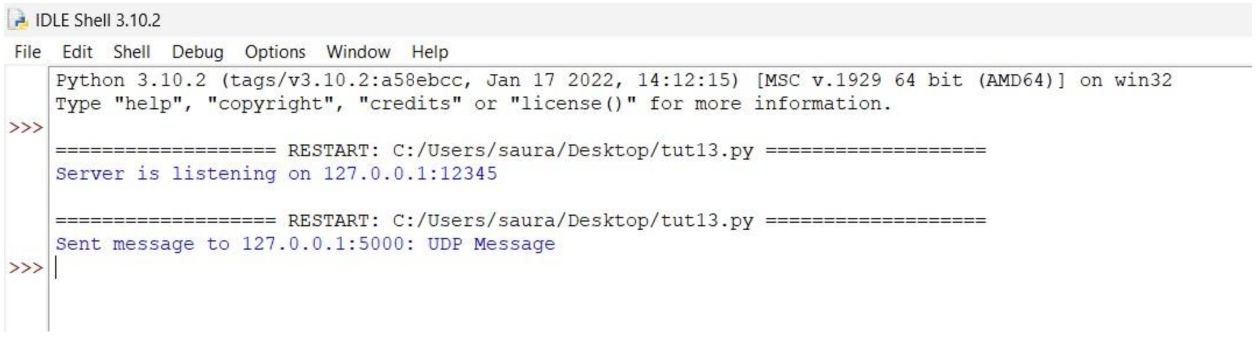
Q3) Create a Python UDP client that sends a "UDP Message" packet to a UDP server. Demonstrate the sending and receiving of the packet.


tut13.py - C:/Users/saura/Desktop/tut13.py (3.10.2)
File Edit Format Run Options Window Help

```
import socket

# Define the server's IP address and port number
server_ip = '127.0.0.1'  # Replace with the server's IP address
server_port = 12345  # Replace with the server's port number

# Create a socket object
server_socket = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)

# Bind the socket to the server address
server_socket.bind((server_ip, server_port))

print(f"Server is listening on {server_ip}:{server_port}")

while True:
    # Receive data from the client
    data, client_address = server_socket.recvfrom(1024)
    print(f"Received data from {client_address}: {data.decode()}")
```

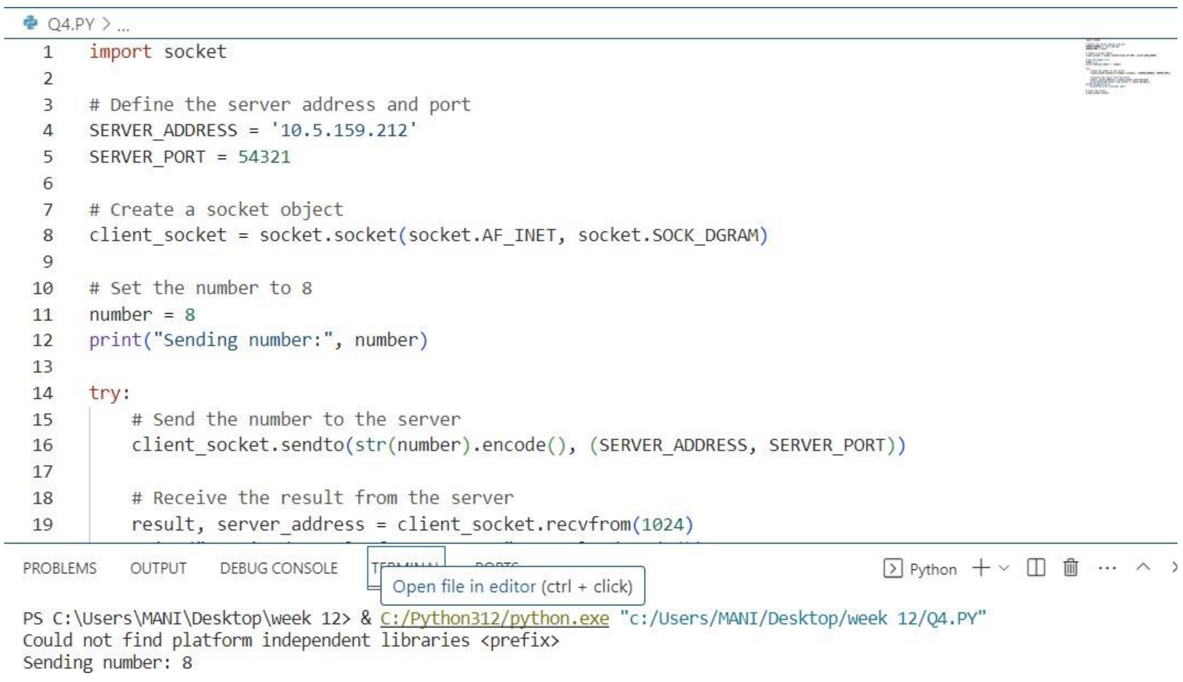

tut13.py - C:/Users/saura/Desktop/tut13.py (3.10.2)
File Edit Format Run Options Window Help

```
import socket

# Define the server's IP address and port number
server_ip = '127.0.0.1'  # Replace with the server's IP address
server_port = 5000  # Replace with the server's port number

# Create a socket object
client_socket = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)

# Message to send
message = "UDP Message"

# Send the message to the server
client_socket.sendto(message.encode(), (server_ip, server_port))
print(f"Sent message to {server_ip}:{server_port}: {message}")

# Close the client socket
client_socket.close()
```



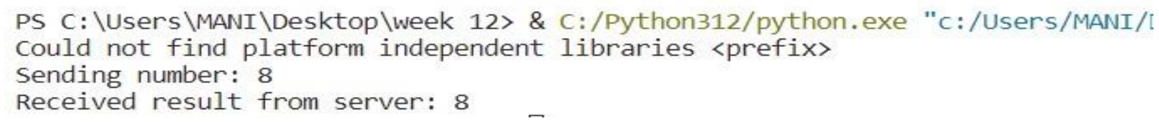
```
    Python 3.10.2 (tags/v3.10.2:a58ebcc, Jan 17 2022, 14:12:15) [MSC v.1929 64 bit (AMD64)] on win32
    Type "help", "copyright", "credits" or "license()" for more information.
>>>
    =================== RESTART: C:/Users/saura/Desktop/tut13.py ===================
    Server is listening on 127.0.0.1:12345

    =================== RESTART: C:/Users/saura/Desktop/tut13.py ===================
    Sent message to 127.0.0.1:5000: UDP Message
>>>
```

Q4) Create a Python UDP client that sends a random number to the UDP server. The server should check if the number is even or odd and send the result back to the client.



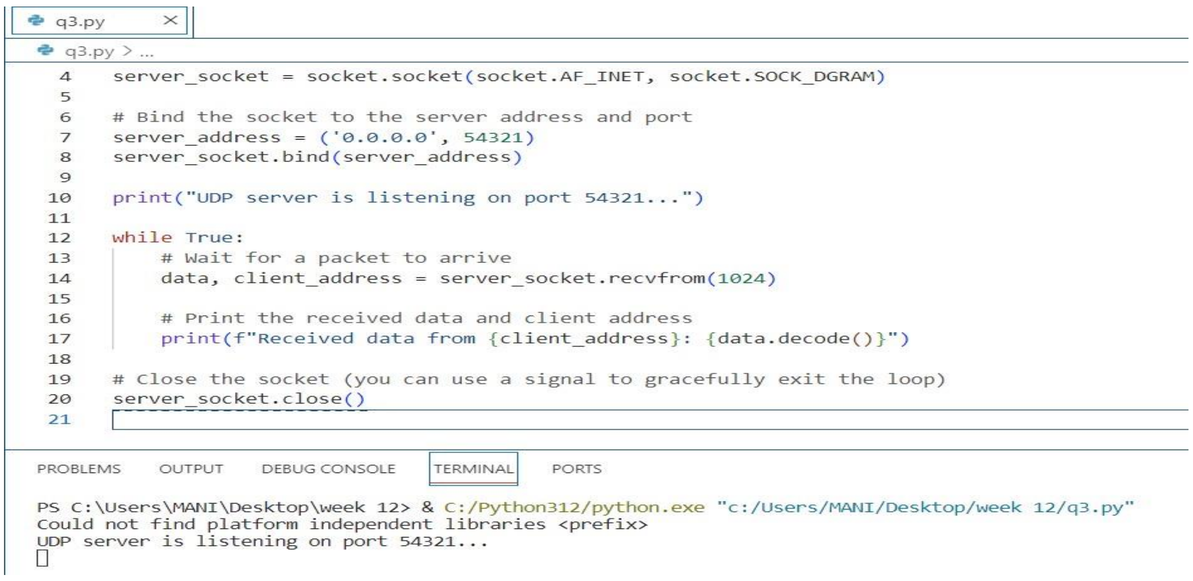
```python
import socket

# Define the server address and port
SERVER_ADDRESS = '10.5.159.212'
SERVER_PORT = 54321

# Create a socket object
client_socket = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)

# Set the number to 8
number = 8
print("Sending number:", number)

try:
    # Send the number to the server
    client_socket.sendto(str(number).encode(), (SERVER_ADDRESS, SERVER_PORT))

    # Receive the result from the server
    result, server_address = client_socket.recvfrom(1024)
```

```
PS C:\Users\MANI\Desktop\week 12> & C:/Python312/python.exe "c:/Users/MANI/Desktop/week 12/Q4.PY"
Could not find platform independent libraries <prefix>
Sending number: 8
```



```
PS C:\Users\MANI\Desktop\week 12> & C:/Python312/python.exe "c:/Users/MANI/I
Could not find platform independent libraries <prefix>
Sending number: 8
Received result from server: 8
```

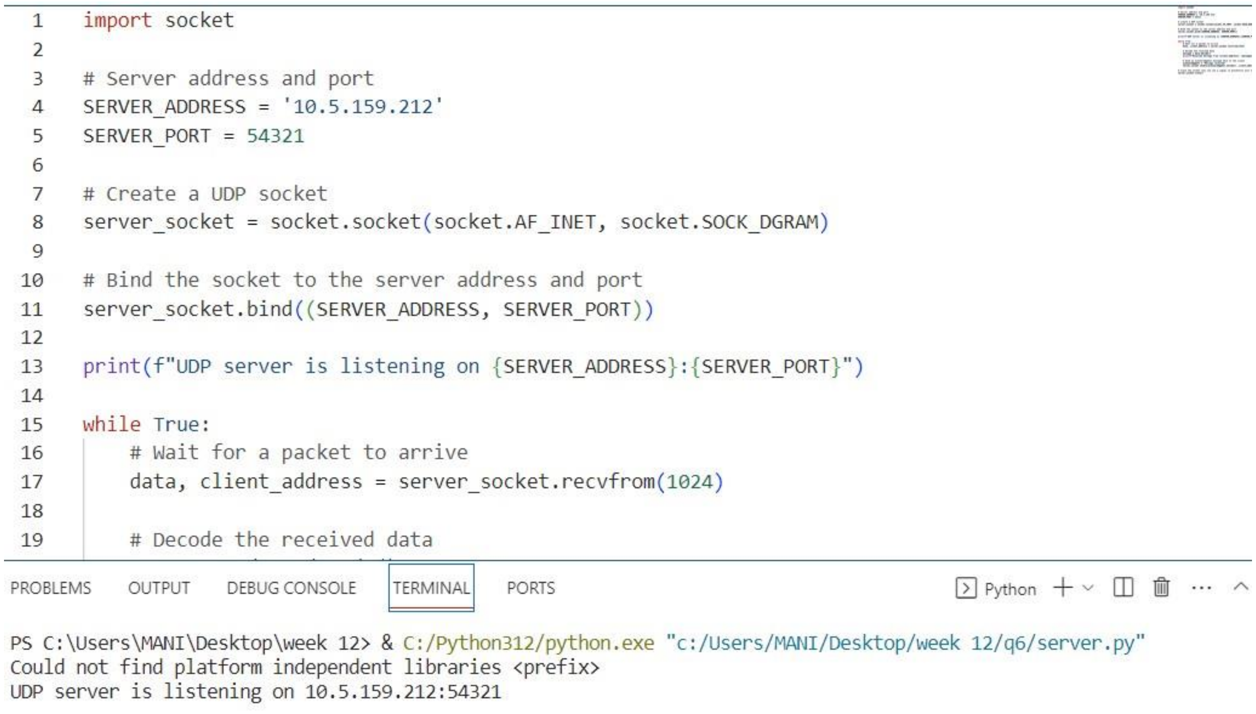
Q5) Write a Python program to create a UDP server that listens on port 54321. Ensure the server can receive UDP packets from clients



```python
server_socket = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)

# Bind the socket to the server address and port
server_address = ('0.0.0.0', 54321)
server_socket.bind(server_address)

print("UDP server is listening on port 54321...")

while True:
    # Wait for a packet to arrive
    data, client_address = server_socket.recvfrom(1024)

    # Print the received data and client address
    print(f"Received data from {client_address}: {data.decode()}")

# Close the socket (you can use a signal to gracefully exit the loop)
server_socket.close()
```

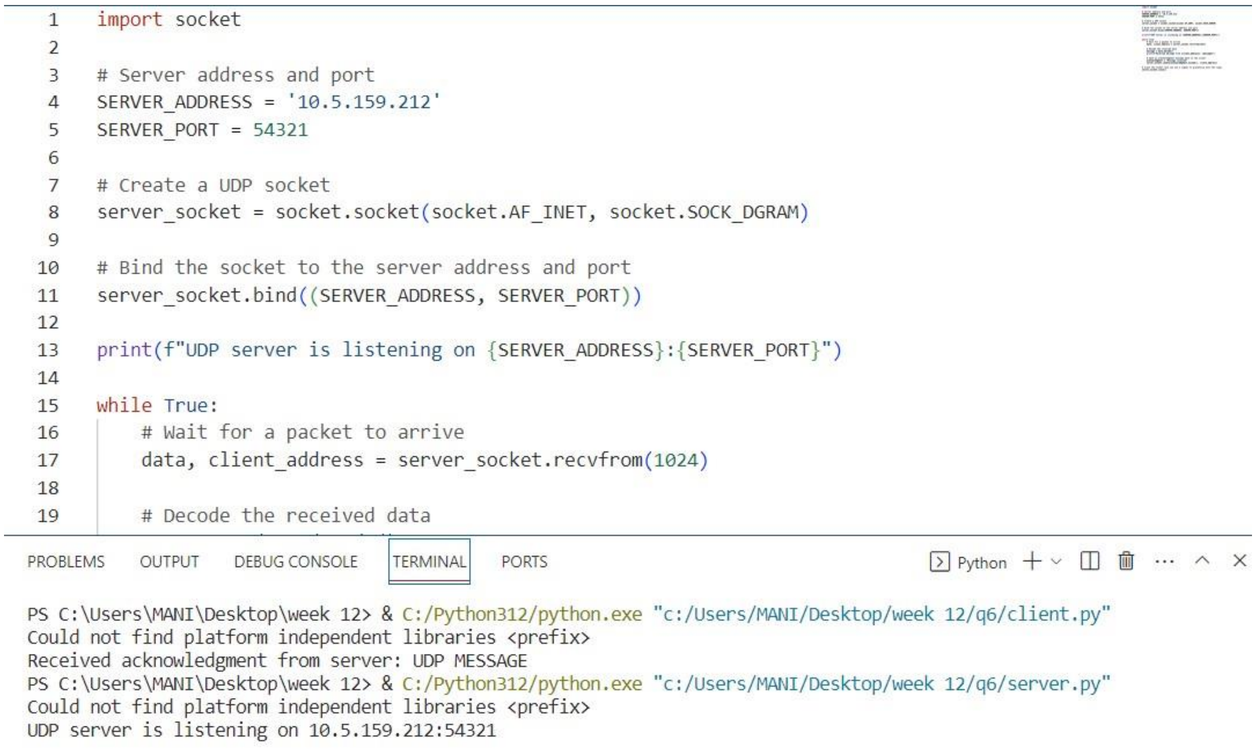
```
PS C:\Users\MANI\Desktop\week 12> & C:/Python312/python.exe "c:/Users/MANI/Desktop/week 12/q3.py"
Could not find platform independent libraries <prefix>
UDP server is listening on port 54321...
```

Q6) Extend the UDP server to respond to the client's "UDP Message" packet with an acknowledgment message. Provide the code for the server-client interaction.



```python
import socket

# Server address and port
SERVER_ADDRESS = '10.5.159.212'
SERVER_PORT = 54321

# Create a UDP socket
server_socket = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)

# Bind the socket to the server address and port
server_socket.bind((SERVER_ADDRESS, SERVER_PORT))

print(f"UDP server is listening on {SERVER_ADDRESS}:{SERVER_PORT}")

while True:
    # Wait for a packet to arrive
    data, client_address = server_socket.recvfrom(1024)

    # Decode the received data
```

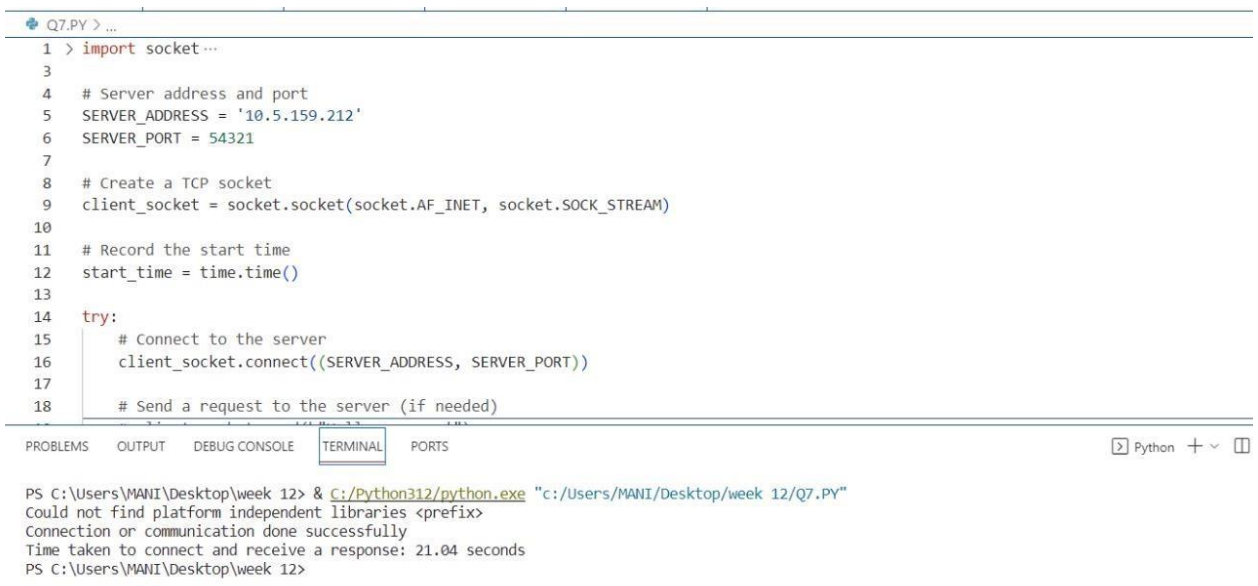
```
PS C:\Users\MANI\Desktop\week 12> & C:/Python312/python.exe "c:/Users/MANI/Desktop/week 12/q6/server.py"
Could not find platform independent libraries <prefix>
UDP server is listening on 10.5.159.212:54321
```



```python
import socket

# Server address and port
SERVER_ADDRESS = '10.5.159.212'
SERVER_PORT = 54321

# Create a UDP socket
server_socket = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)

# Bind the socket to the server address and port
server_socket.bind((SERVER_ADDRESS, SERVER_PORT))

print(f"UDP server is listening on {SERVER_ADDRESS}:{SERVER_PORT}")

while True:
    # Wait for a packet to arrive
    data, client_address = server_socket.recvfrom(1024)

    # Decode the received data
```

```
PS C:\Users\MANI\Desktop\week 12> & C:/Python312/python.exe "c:/Users/MANI/Desktop/week 12/q6/client.py"
Could not find platform independent libraries <prefix>
Received acknowledgment from server: UDP MESSAGE
PS C:\Users\MANI\Desktop\week 12> & C:/Python312/python.exe "c:/Users/MANI/Desktop/week 12/q6/server.py"
Could not find platform independent libraries <prefix>
UDP server is listening on 10.5.159.212:54321
```

Q7) Implement a Python program that calculates and displays the time taken for a TCP client to connect to the server and receive a response. Measure the time elapsed in seconds



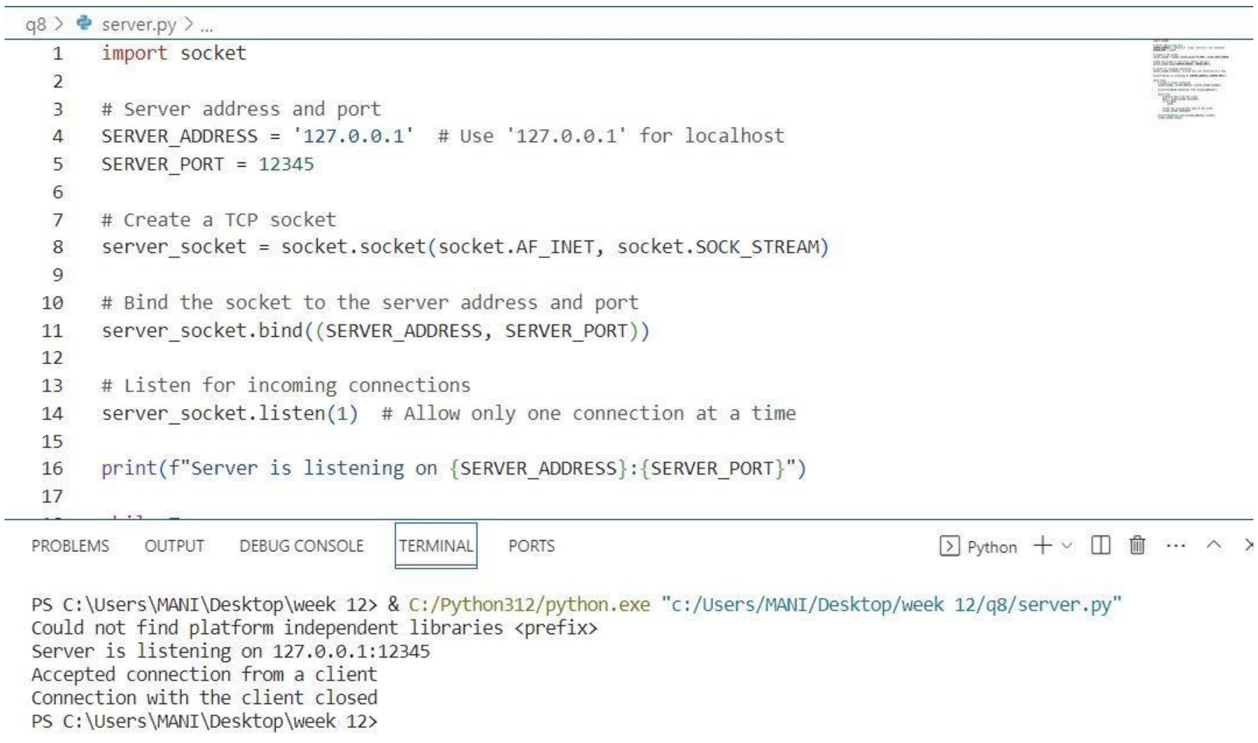
Q7.PY > ...

```python
import socket ···

# Server address and port
SERVER_ADDRESS = '10.5.159.212'
SERVER_PORT = 54321

# Create a TCP socket
client_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

# Record the start time
start_time = time.time()

try:
    # Connect to the server
    client_socket.connect((SERVER_ADDRESS, SERVER_PORT))

    # Send a request to the server (if needed)
```

```
PS C:\Users\MANI\Desktop\week 12> & C:/Python312/python.exe "c:/Users/MANI/Desktop/week 12/Q7.PY"
Could not find platform independent libraries <prefix>
Connection or communication done successfully
Time taken to connect and receive a response: 21.04 seconds
PS C:\Users\MANI\Desktop\week 12>
```

Q8) Create a TCP server that echoes back any message it receives from a client. Develop a Python client to send messages to the server and display the echoed response.

Server.py

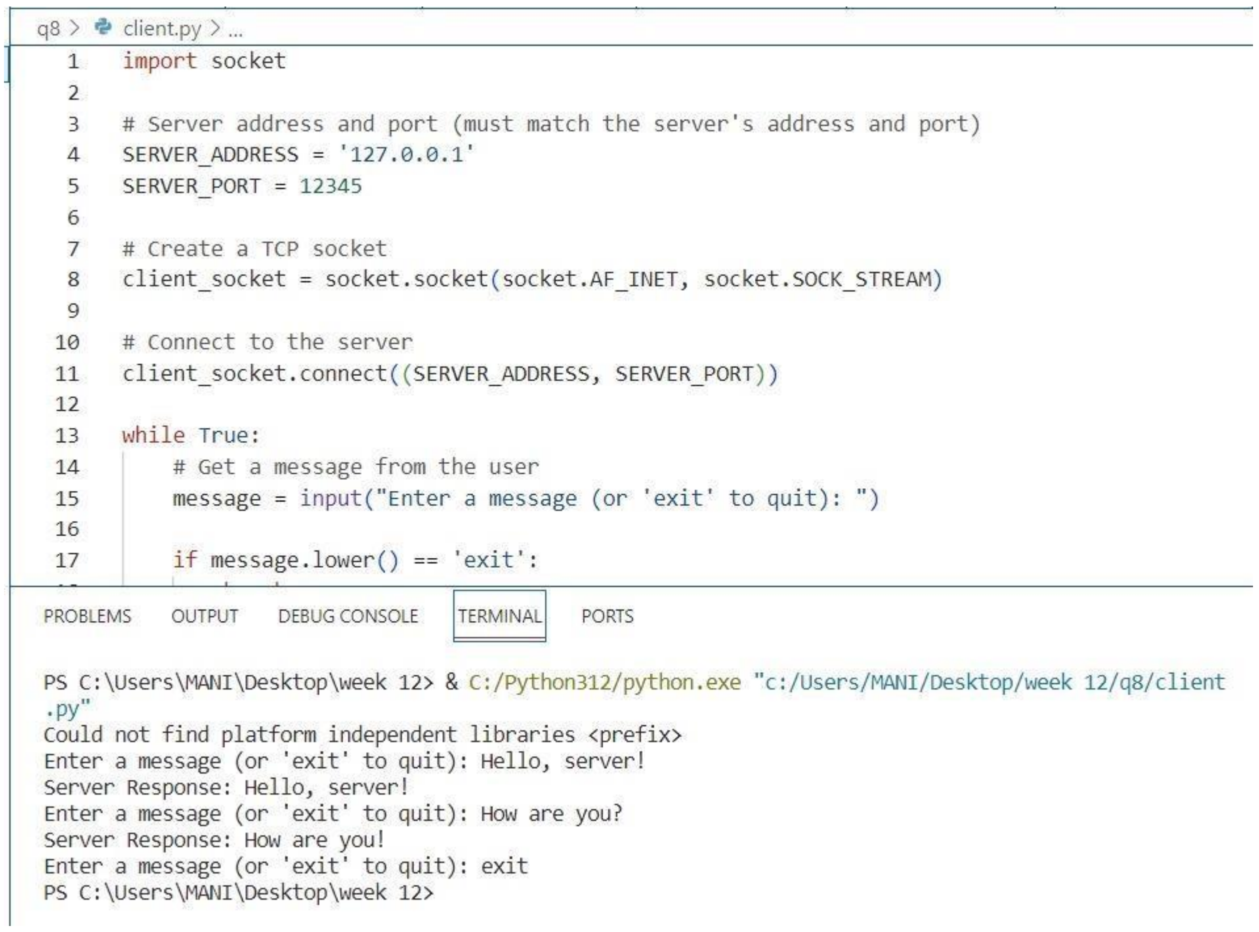


q8 > server.py > ...

```python
import socket

# Server address and port
SERVER_ADDRESS = '127.0.0.1'  # Use '127.0.0.1' for localhost
SERVER_PORT = 12345

# Create a TCP socket
server_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

# Bind the socket to the server address and port
server_socket.bind((SERVER_ADDRESS, SERVER_PORT))

# Listen for incoming connections
server_socket.listen(1)  # Allow only one connection at a time

print(f"Server is listening on {SERVER_ADDRESS}:{SERVER_PORT}")

```

```
PS C:\Users\MANI\Desktop\week 12> & C:/Python312/python.exe "c:/Users/MANI/Desktop/week 12/q8/server.py"
Could not find platform independent libraries <prefix>
Server is listening on 127.0.0.1:12345
Accepted connection from a client
Connection with the client closed
PS C:\Users\MANI\Desktop\week 12>
```

Client.py



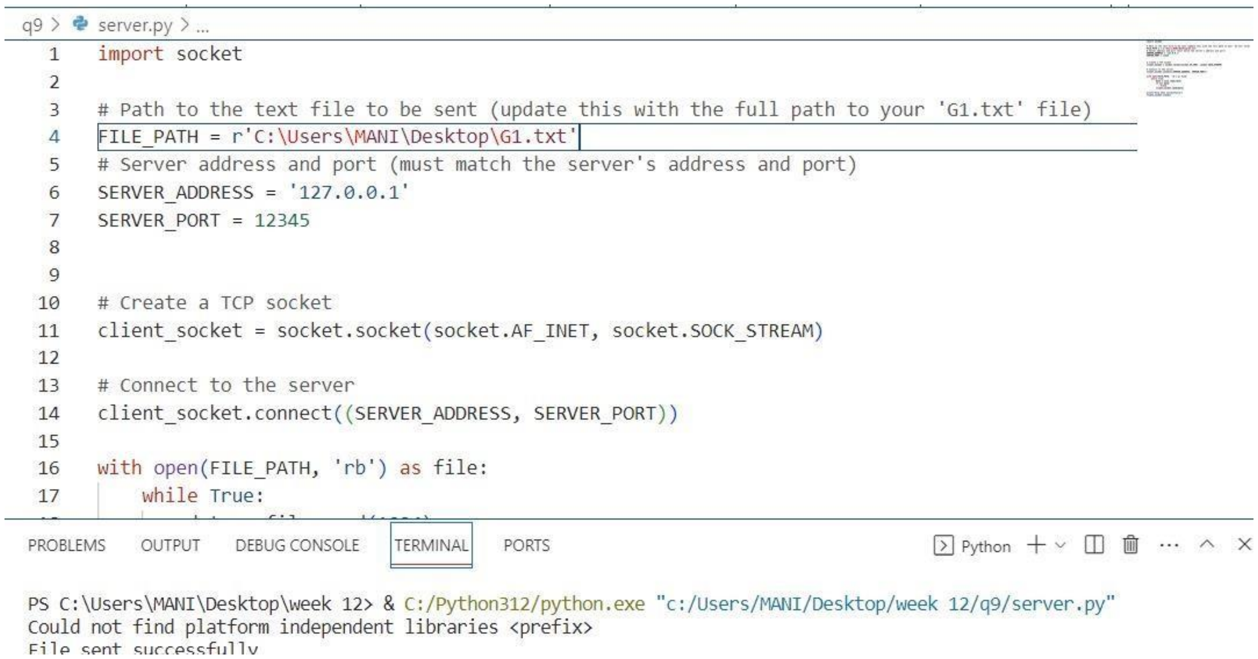
q8 > client.py > ...

```python
import socket

# Server address and port (must match the server's address and port)
SERVER_ADDRESS = '127.0.0.1'
SERVER_PORT = 12345

# Create a TCP socket
client_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

# Connect to the server
client_socket.connect((SERVER_ADDRESS, SERVER_PORT))

while True:
    # Get a message from the user
    message = input("Enter a message (or 'exit' to quit): ")

    if message.lower() == 'exit':
```

```
PS C:\Users\MANI\Desktop\week 12> & C:/Python312/python.exe "c:/Users/MANI/Desktop/week 12/q8/client
.py"
Could not find platform independent libraries <prefix>
Enter a message (or 'exit' to quit): Hello, server!
Server Response: Hello, server!
Enter a message (or 'exit' to quit): How are you?
Server Response: How are you!
Enter a message (or 'exit' to quit): exit
PS C:\Users\MANI\Desktop\week 12>
```

Q9) Develop a simple Python program that sends a small text file from a TCP client to a TCP server. Confirm that the file is received and saved correctly.

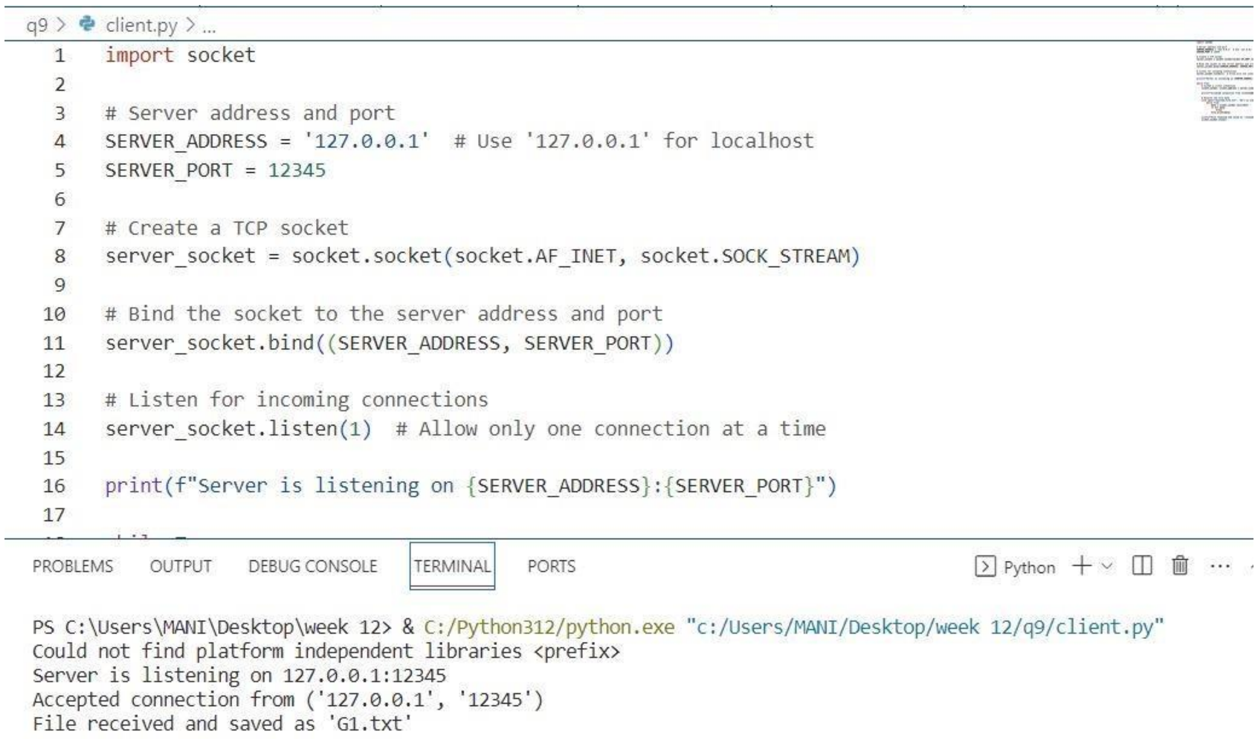
SERVER.PY



```python
import socket

# Path to the text file to be sent (update this with the full path to your 'G1.txt' file)
FILE_PATH = r'C:\Users\MANI\Desktop\G1.txt'
# Server address and port (must match the server's address and port)
SERVER_ADDRESS = '127.0.0.1'
SERVER_PORT = 12345


# Create a TCP socket
client_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

# Connect to the server
client_socket.connect((SERVER_ADDRESS, SERVER_PORT))

with open(FILE_PATH, 'rb') as file:
    while True:
```

```
PS C:\Users\MANI\Desktop\week 12> & C:/Python312/python.exe "c:/Users/MANI/Desktop/week 12/q9/server.py"
Could not find platform independent libraries <prefix>
File sent successfully
```

CLIENT.PY

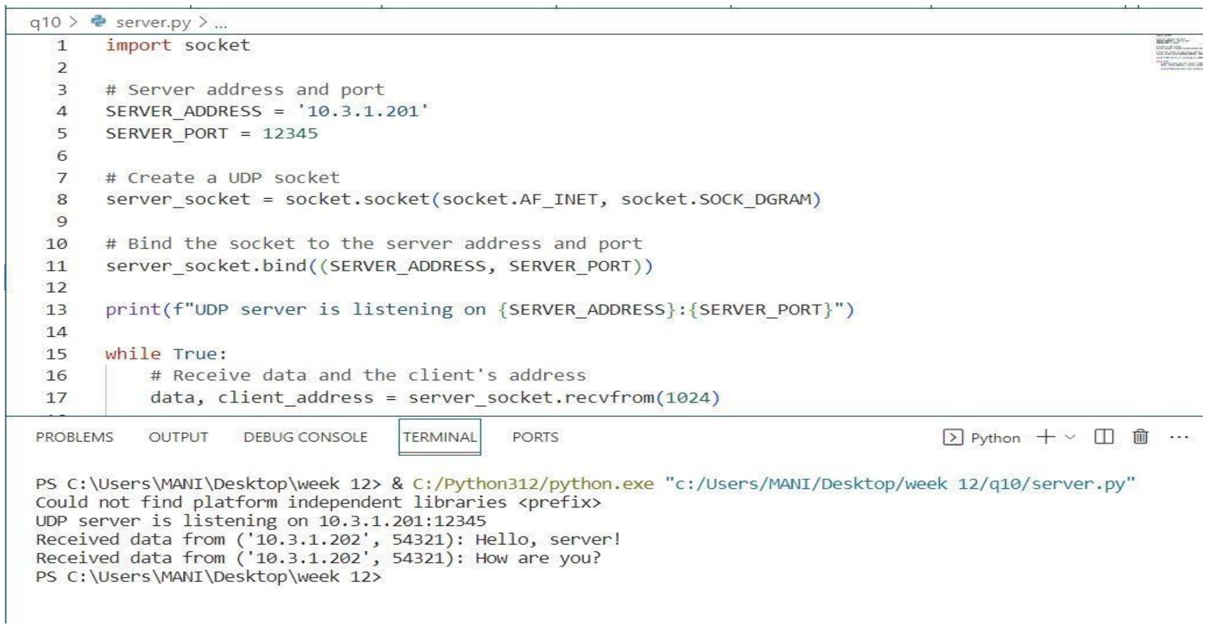


```python
import socket

# Server address and port
SERVER_ADDRESS = '127.0.0.1'  # Use '127.0.0.1' for localhost
SERVER_PORT = 12345

# Create a TCP socket
server_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

# Bind the socket to the server address and port
server_socket.bind((SERVER_ADDRESS, SERVER_PORT))

# Listen for incoming connections
server_socket.listen(1)  # Allow only one connection at a time

print(f"Server is listening on {SERVER_ADDRESS}:{SERVER_PORT}")

```

```
PS C:\Users\MANI\Desktop\week 12> & C:/Python312/python.exe "c:/Users/MANI/Desktop/week 12/q9/client.py"
Could not find platform independent libraries <prefix>
Server is listening on 127.0.0.1:12345
Accepted connection from ('127.0.0.1', '12345')
File received and saved as 'G1.txt'
```

Q10) Write a Python program to receive UDP packets and display their content. Simulate sending UDP packets from a separate client program

SERVER.PY

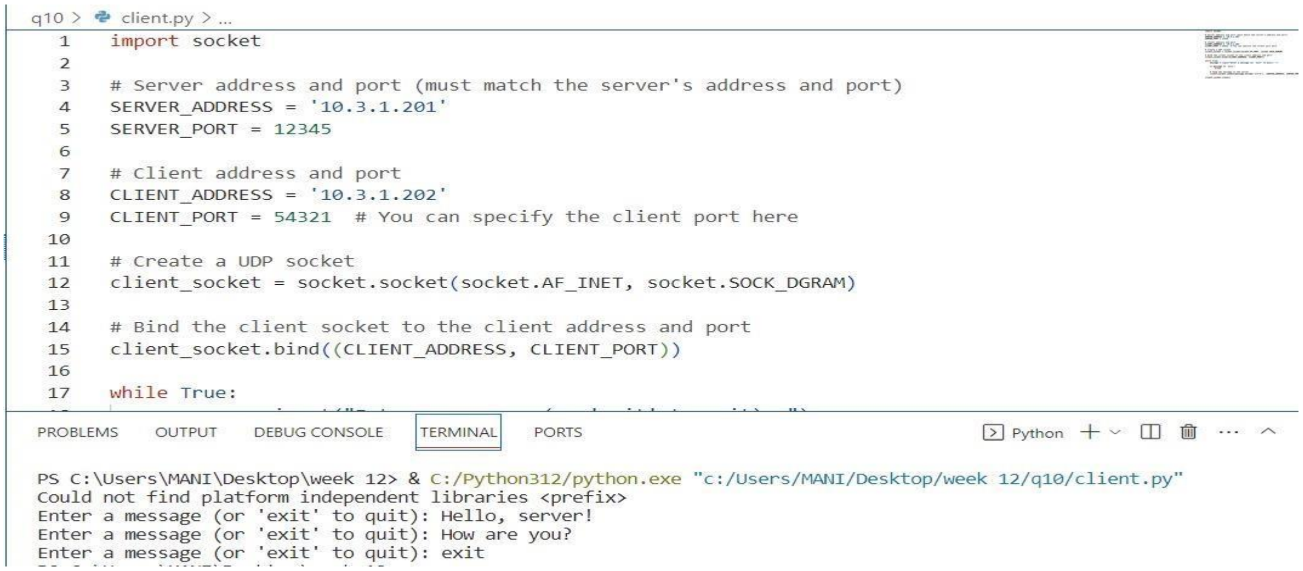


q10 > server.py > ...

```python
import socket

# Server address and port
SERVER_ADDRESS = '10.3.1.201'
SERVER_PORT = 12345

# Create a UDP socket
server_socket = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)

# Bind the socket to the server address and port
server_socket.bind((SERVER_ADDRESS, SERVER_PORT))

print(f"UDP server is listening on {SERVER_ADDRESS}:{SERVER_PORT}")

while True:
    # Receive data and the client's address
    data, client_address = server_socket.recvfrom(1024)
```

```
PS C:\Users\MANI\Desktop\week 12> & C:/Python312/python.exe "c:/Users/MANI/Desktop/week 12/q10/server.py"
Could not find platform independent libraries <prefix>
UDP server is listening on 10.3.1.201:12345
Received data from ('10.3.1.202', 54321): Hello, server!
Received data from ('10.3.1.202', 54321): How are you?
PS C:\Users\MANI\Desktop\week 12>
```

CLIENT.PY



q10 > client.py > ...

```python
import socket

# Server address and port (must match the server's address and port)
SERVER_ADDRESS = '10.3.1.201'
SERVER_PORT = 12345

# Client address and port
CLIENT_ADDRESS = '10.3.1.202'
CLIENT_PORT = 54321  # You can specify the client port here

# Create a UDP socket
client_socket = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)

# Bind the client socket to the client address and port
client_socket.bind((CLIENT_ADDRESS, CLIENT_PORT))

while True:
```

```
PS C:\Users\MANI\Desktop\week 12> & C:/Python312/python.exe "c:/Users/MANI/Desktop/week 12/q10/client.py"
Could not find platform independent libraries <prefix>
Enter a message (or 'exit' to quit): Hello, server!
Enter a message (or 'exit' to quit): How are you?
Enter a message (or 'exit' to quit): exit
```