REGISTER NUMBER: **Name :**

SAUVIK DAS

RA1811028010101

REGISTER NUMBER: **Name :**

Ex : No :	
Date:	

Q 1. Write a python code to create a Simple TCP Client and Server, for the client to get date time from server machine.

Sol.

Server:

```
import socket
import datetime
Date = datetime.datetime.now()
#print(Date)
Sock = socket.socket()
host = socket.gethostname()
port = 9999
Sock.bind((host,port))
print("Connecting")
Sock.listen(10)
while True:
conn,addr = Sock.accept()
print('Connected to ',addr)
conn.send(str(Date).encode())
Sock.close()
```

Client:

```
import socket
Sock = socket.socket()
host = socket.gethostname()
```

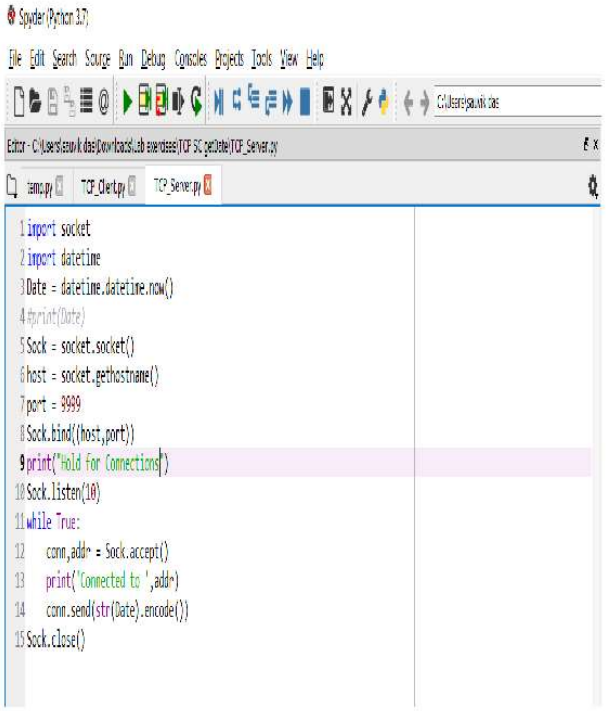
REGISTER NUMBER: **Name :**

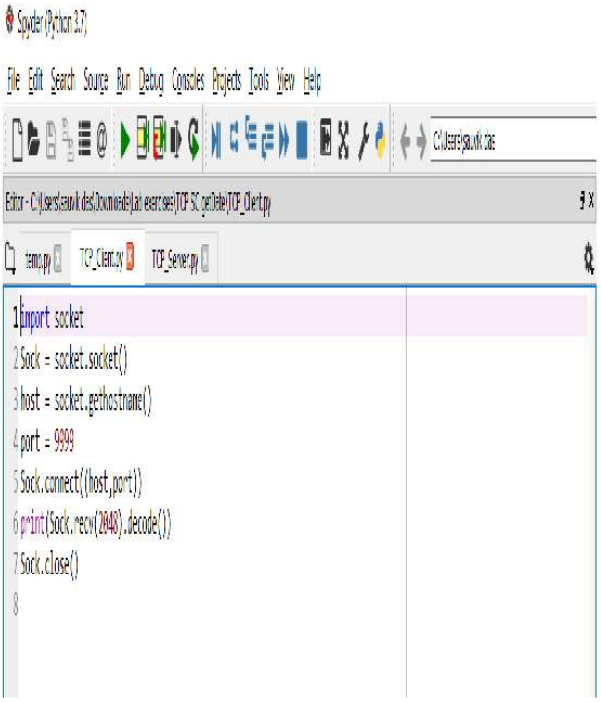
port = 9999

Sock.connect((host,port))

print(Sock.recv(2048).decode())

Sock.close()





REGISTER NUMBER: **Name :**

Q 2. Write a python code to create Echo client server application using UDP Sockets. The message sent by the client is echoed back to it by the server.

Sol.

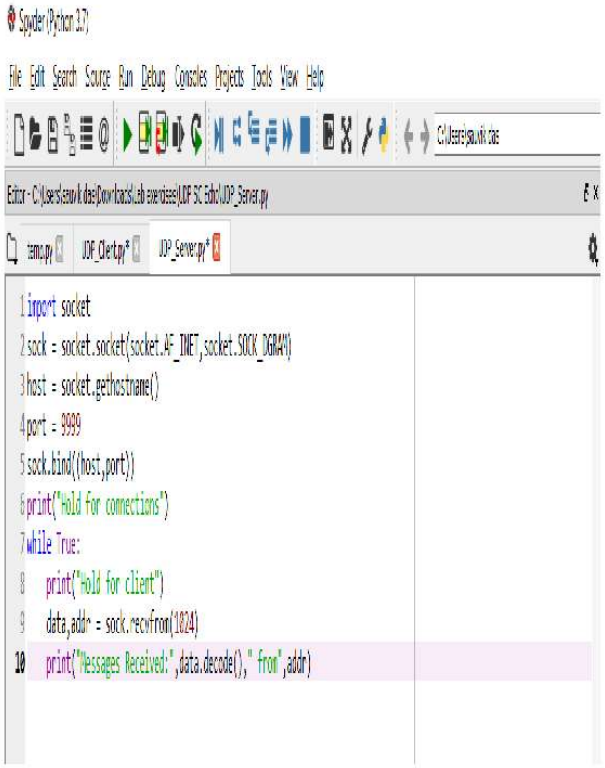
Server:

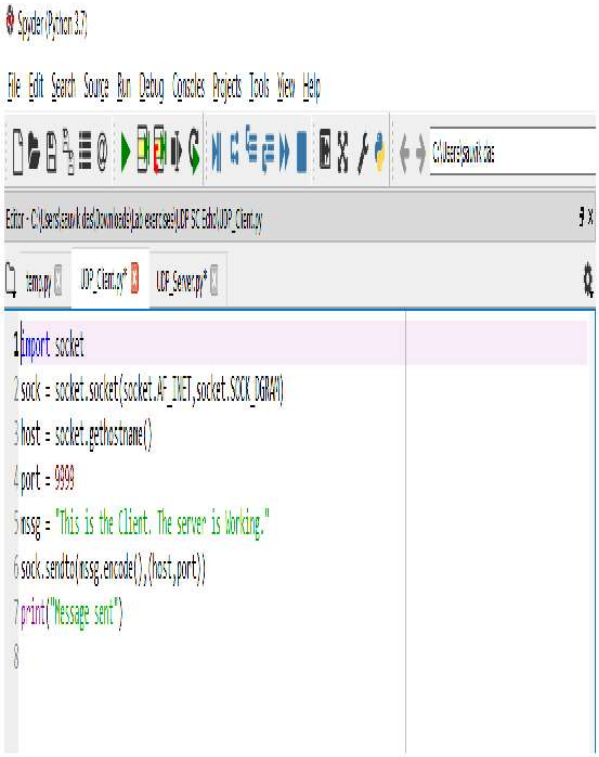
```
import socket

sock = socket.socket(socket.AF_INET,socket.SOCK_DGRAM)

host = socket.gethostname()

port = 9999

sock.bind((host,port))

print("Connecting")

while True:

print("Hold for client")

data,addr = sock.recvfrom(1024)

print("Message received !",data.decode()," from",addr)
```

Client:

```
import socket

sock = socket.socket(socket.AF_INET,socket.SOCK_DGRAM)

host = socket.gethostname()

port = 9999

mssg = "This is the Client. The server is Working."

sock.sendto(mssg.encode(),(host,port))

print("Message sent")
```

REGISTER NUMBER: **Name :**





REGISTER NUMBER: **Name :**

Q 3. Write a python code to simulate simple client server chat application using TCP/IP Sockets.

Sol.

Server:

```
import socket
sock = socket.socket()
host = socket.gethostname()
port = 9999
sock.bind((host,port))
sock.listen(5)
print("Hold for client")
con,addr = sock.accept()
print("Connected to ",addr)
while True:
data = con.recv(1024)
print("Client : "+ data.decode())
if data == 'quit' or not data:
print("Server Quiting")
break
data = input("Server: ")
con.sendall(data.encode())
con.close()
```

Client:

```
import socket
import datetime
name = input("Please input your name: ")
```

REGISTER NUMBER: **Name :**

```
sock = socket.socket()
host = socket.gethostname()
port = 9999
sock.connect((host,port))
mssg = "You are now connected to " + name + " at " + str(datetime.datetime.now())
print("Server connected")
sock.sendall(mssg.encode())
while True:
mssg = sock.recv(2048)
if mssg.decode() == 'quit' or not mssg:
print(name+" is now disconnecting at "+ str(datetime.datetime.now()))
break
else:
print("Server: "+mssg.decode())
mssg = input("Client: ")
sock.sendall(mssg.encode())
sock.close()
```

REGISTER NUMBER: **Name :**

Spyder (Python 3.7)

File Edit Search Source Run Debug Consoles Projects Tools View Help

Editor - C:\Users\sawvik das\Downloads\lab exercises\TCP Chat APP\Server.py

temp.py | Client.py | Server.py

```python
import socket
sock = socket.socket()
host = socket.gethostname()
port = 9999
sock.bind((host,port))
sock.listen(5)
print("Hold for client")
con,addr = sock.accept()
print("Connected to ",addr)
while True:
    data = con.recv(1024)
    print("Client: "+ data.decode())
    if data == 'quit' or not data:
        print("Server Quiting")
        break
    data = input("Server: ")
    con.sendall(data.encode())
con.close()
```

Spyder (Python 3.7)

File Edit Search Source Run Debug Consoles Projects Tools View Help

Editor - C:\Users\sawvik das\Downloads\Lab exercises\TCP Chat APP\Client.py

temp.py | Client.py | Server.py

```python
import socket
import datetime
name = input("Please input your name: ")
sock = socket.socket()
host = socket.gethostname()
port = 9999
sock.connect((host,port))
mssg = "You are now connected to " + name + " at " + str(datetime.datetime.now())
print("Server connected")
sock.sendall(mssg.encode())
while True:
    mssg = sock.recv(2048)
    if mssg.decode() == 'quit' or not mssg:
        print(name+" is now disconnecting at "+ str(datetime.datetime.now()))
        break
    else:
        print("Server: "+mssg.decode())
        mssg = input("Client: ")
        sock.sendall(mssg.encode())
sock.close()
```

REGISTER NUMBER: **Name :**

Q 4. Write a python code to get the IP address of the client machine using UDP sockets.

Sol.

Server:

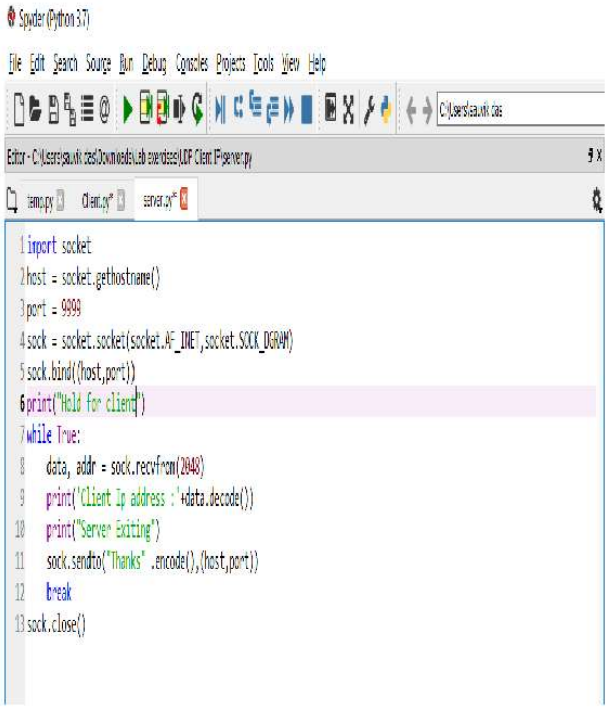
```
import socket
host = socket.gethostname()
port = 9999
sock = socket.socket(socket.AF_INET,socket.SOCK_DGRAM)
sock.bind((host,port))
print("Hold for client")
while True:
data, addr = sock.recvfrom(2048)
print('Client Ip address :'+data.decode())
print("Server Exiting")
sock.sendto("Thanks" .encode(),(host,port))
break
sock.close()
```

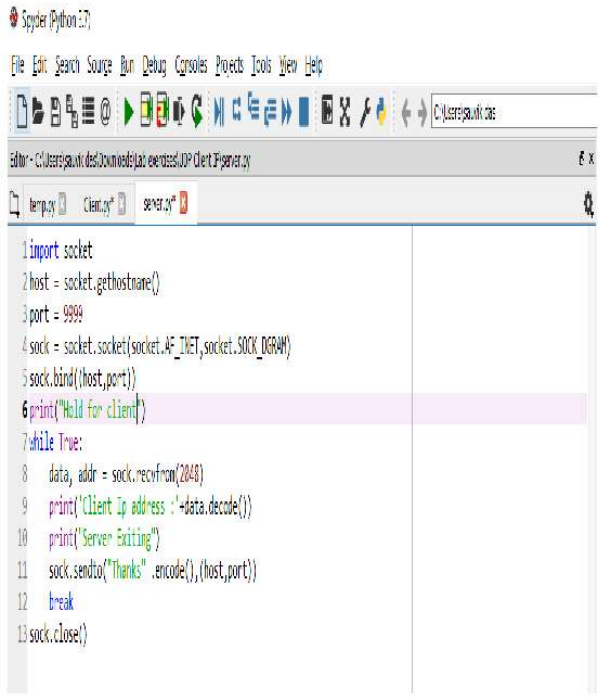
Client:

```
import socket
import socket
host = socket.gethostname()
port = 9999
sock = socket.socket(socket.AF_INET,socket.SOCK_DGRAM)
hostname = socket.gethostname()
IPAddr = str(socket.gethostbyname(hostname))
sock.sendto(IPAddr.encode(),(host,port))
```

REGISTER NUMBER: **Name :**

print("IP Sent to Server")





REGISTER NUMBER: **Name :**

Q 5. Create a Simple multi point client server chat using TCP Sockets.

Sol.

Server:

```
import socket
from threading import Thread
def threads():
while True:
data = conn.recv(1024)
print('Client Request :' + data.decode())
if data == 'quit' or not data:
print("Server Exiting")
break
data = input('Server Response:')
conn.sendall(data.encode())
host = socket.gethostname()
port = 9999
s = socket.socket()
s.bind((host,port))
s.listen(10)
print("Hold For Clients")
while True:
conn,addr = s.accept()
print("Connected to ", addr)
child = Thread(target=threads)
child.start()
conn.close()
```

REGISTER NUMBER: **Name :**

Client:

```
import socket
import datetime
name = input("Enter Client Name : ")
sock = socket.socket()
host = socket.gethostname()
port = 9999
sock.connect((host,port))
mssg = "You are now connected to " + name + " at " + str(datetime.datetime.now())
print("Server connected")
sock.sendall(mssg.encode())
while True:
mssg = sock.recv(2048)
if mssg.decode() == 'quit' or not mssg:
print(name+"is now Disconnecting at "+ str(datetime.datetime.now()))
break
else:
print("Server: "+mssg.decode())
mssg = input("Client: ")
sock.sendall(mssg.encode())
sock.close()
```

REGISTER NUMBER: **Name :**

Spyder (Python 3.7)

File Edit Search Source Run Debug Consoles Projects Tools View Help

C:\Users\saurik das

Editor - C:\Users\saurik das\Downloads\Lab exercises\TCP Multichat APP\server.py

temp.py | Client.py | server.py*

```python
import socket
from threading import Thread
def threads():
    while True:
        data = conn.recv(1024)
        print('Client Request :' + data.decode())
        if data == 'quit' or not data:
            print("Server Exiting")
            break
        data = input('Server Response: ')
        conn.sendall(data.encode())


host = socket.gethostname()
port = 9999
s = socket.socket()
s.bind((host,port))
s.listen(10)
print("Hold For Clients")
while True:
    conn,addr = s.accept()
    print("Connected by ", addr)
    child = Thread(target=threads)
    child.start()
conn.close()
```

Spyder (Python 3.7)

File Edit Search Source Run Debug Consoles Projects Tools View Help

C:\Users\saurik das

Editor - C:\Users\saurik das\Downloads\Lab exercises\TCP Multichat APP\Client.py

temp.py | Client.py | server.py*

```python
import socket
import datetime
name = input("Enter Client Name : ")
sock = socket.socket()
host = socket.gethostname()
port = 9999
sock.connect((host,port))
mssg = "You are now connected to " + name + " at " + str(datetime.datetime.now())
print("Server connected")
sock.sendall(mssg.encode())
while True:
    mssg = sock.recv(2048)
    if mssg.decode() == 'quit' or not mssg:
        print(name+"is now Disconnecting at "+ str(datetime.datetime.now()))
        break
    else:
        print("Server: "+mssg.decode())
        mssg = input("Client: ")
        sock.sendall(mssg.encode())
sock.close()
```

REGISTER NUMBER: **Name :**

Q 6. Write a Python code to Implement the File Transfer Protocol using TCP Sockets. (FTP Example in common classroom).

Sol.

Server:

```
import socket
host = socket.gethostname()
port = 9999
sock = socket.socket()
sock.bind((host,port))
sock.listen(10)
print("Hold for connections")
while True:
conn, addr = sock.accept()
print("Connected to ",addr)
data = conn.recv(2048)
print("Server recieved : " ,data.decode())
file = open("input.txt",'rb')
l = file.read(1024)
while(l):
conn.send(l)
print('sent',l)
l = file.read(1024)
file.close()
print("File Sent!")
conn.close()
```

REGISTER NUMBER: **Name :**

Client:

```
import socket
host = socket.gethostname()
port =9999
sock = socket.socket()
sock.connect((host,port))
sock.send("Cleint Connected.".encode())
with open('output.txt','wb') as file:
print('File Created.')
while True:
print('Recieving Data...')
data = sock.recv(1024)
print('Data Recieved : ', data.decode())
if not data:
break
file.write(data)
file.close()
print("Write Successful")
sock.close()
print("Client Disconnected")
```

REGISTER NUMBER: **Name :**

Spyder (Python 3.7)

File Edit Search Source Run Debug Consoles Projects Tools View Help

Editor - C:\Users\saurik das\Downloads\Lab exercises\FTP\server.py

temp.py client.py server.py*

```python
import socket
host = socket.gethostname()
port = 9999
sock = socket.socket()
sock.bind((host,port))
sock.listen(10)
print("Hold for connections")
while True:
    conn, addr = sock.accept()
    print("Connected to ",addr)
    data = conn.recv(2048)
    print("Server recieved : " ,data.decode())
    file = open("input.txt",'rb')
    l = file.read(1024)
    while(l):
        conn.send(l)
        print('sent',l)
        l = file.read(1024)
    file.close()
    print("File Sent!")
    conn.close()
```

Spyder (Python 3.7)

File Edit Search Source Run Debug Consoles Projects Tools View Help

Editor - C:\Users\saurik das\Downloads\Lab exercises\FTP\client.py

temp.py client.py server.py

```python
import socket
host = socket.gethostname()
port =9999
sock = socket.socket()
sock.connect((host,port))
sock.send("Cleint Connected.".encode())
with open('output.txt','wb') as file:
    print('File Created.')
    while True:
        print('Recieving Data...')
        data = sock.recv(1024)
        print('Data Recieved : ', data.decode())
        if not data:
            break
        file.write(data)
file.close()
print("Write Successful")
sock.close()
print("Client Disconnected")
```

REGISTER NUMBER: **Name :**

Q 7. Create a server application which send the Ip address to client for the given domain name using UDP and use dictionary to store the domain and IP address combination. (DNS Server Example in common classroom).

Sol.

Server:

```
import socket
dns = {
'google': '172.13.31.89',
'gmail': '154.32.176.11',
'w3school': '143.76.142.21',
'yahoo': '162.54.26.11',
'bing': '148.47.25.33',
'youtube': '163.38.31.11'
}
host = socket.gethostname()
sock_r = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
sock_s = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
sock_r.bind((host,9999))
print("Waiting for client...")
while True:
data, addr = sock_r.recvfrom(1024)
print('Domain :'+data.decode(), addr)
ip = ""
if data.decode() in dns.keys():
ip = dns.get(data.decode())
elif( data.decode() == "bye" ) or not data:
break
```

REGISTER NUMBER: **Name :**

```
else:

ip = "Not Found"
sock_s.sendto(ip.encode(),(host,9998))
sock_r.close()
sock_s.close()
```

Client:

```
import socket
host = socket.gethostname()
sock_r = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
sock_s = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
sock_r.bind((host, 9998))
print("Connecting...")
while True:
data = input("Domain : ")
if data == 'bye' or not data:
break
else:
sock_s.sendto(data.encode(), (host, 9999))
data = sock_r.recv(1024)
print("IP: "+data.decode())
sock_s.close()
sock_r.close()
```

REGISTER NUMBER: **Name :**

Spyder (Python 3.7)

File Edit Search Source Run Debug Consoles Projects Tools View Help

Editor - C:\Users\sauvik das\Downloads\Lab exercises\UDP IP Domain\server.py

temp.py | Client.py* | server.py*

```python
import socket
dns = {
    'google': '172.13.31.89',
    'gmail': '154.32.176.11',
    'w3school': '143.76.142.21',
    'yahoo': '162.54.26.11',
    'bing': '148.47.25.33',
    'youtube': '163.38.31.11'
    }
host = socket.gethostname()
sock_r = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
sock_s = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
sock_r.bind((host,9999))
print("Waiting for client...")
while True:
    data, addr = sock_r.recvfrom(1024)
    print('Domain :'+data.decode(), addr)
    ip = ""
    if data.decode() in dns.keys():
        ip = dns.get(data.decode())
    elif( data.decode() == "bye" ) or not data:
        break
    else:
        ip = "Not Found"
    sock_s.sendto(ip.encode(),(host,9998))
sock_r.close()
sock_s.close()
```

Spyder (Python 3.7)

File Edit Search Source Run Debug Consoles Projects Tools View Help

Editor - C:\Users\sauvik das\Downloads\Lab exercises\UDP IP Domain\Client.py

temp.py | Client.py* | server.py*

```python
import socket
host = socket.gethostname()
sock_r = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
sock_s = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
sock_r.bind((host, 9998))
print("Connecting...")
while True:
    data = input("Domain : ")
    if data == "bye" or not data:
        break
    else:
        sock_s.sendto(data.encode(), (host, 9999))
        data = sock_r.recv(1024)
        print("IP: "+data.decode())
sock_s.close()
sock_r.close()
```