1. **Using TCP/IP sockets, write a client-server program to make client sending the file name and the server to send back the contents of the requested file if present.**

2. **Client.py**

```
from socket import *
serverName = “127.0.0.1”
serverPort = 12000
clientSocket = socket(AF_INET, SOCK_STREAM)
clientSocket.connect((serverName,serverPort))
sentence = input("Enter file name")

clientSocket.send(sentence.encode())
filecontents = clientSocket.recv(1024).decode()
print ('From Server:', filecontents)
clientSocket.close()
```

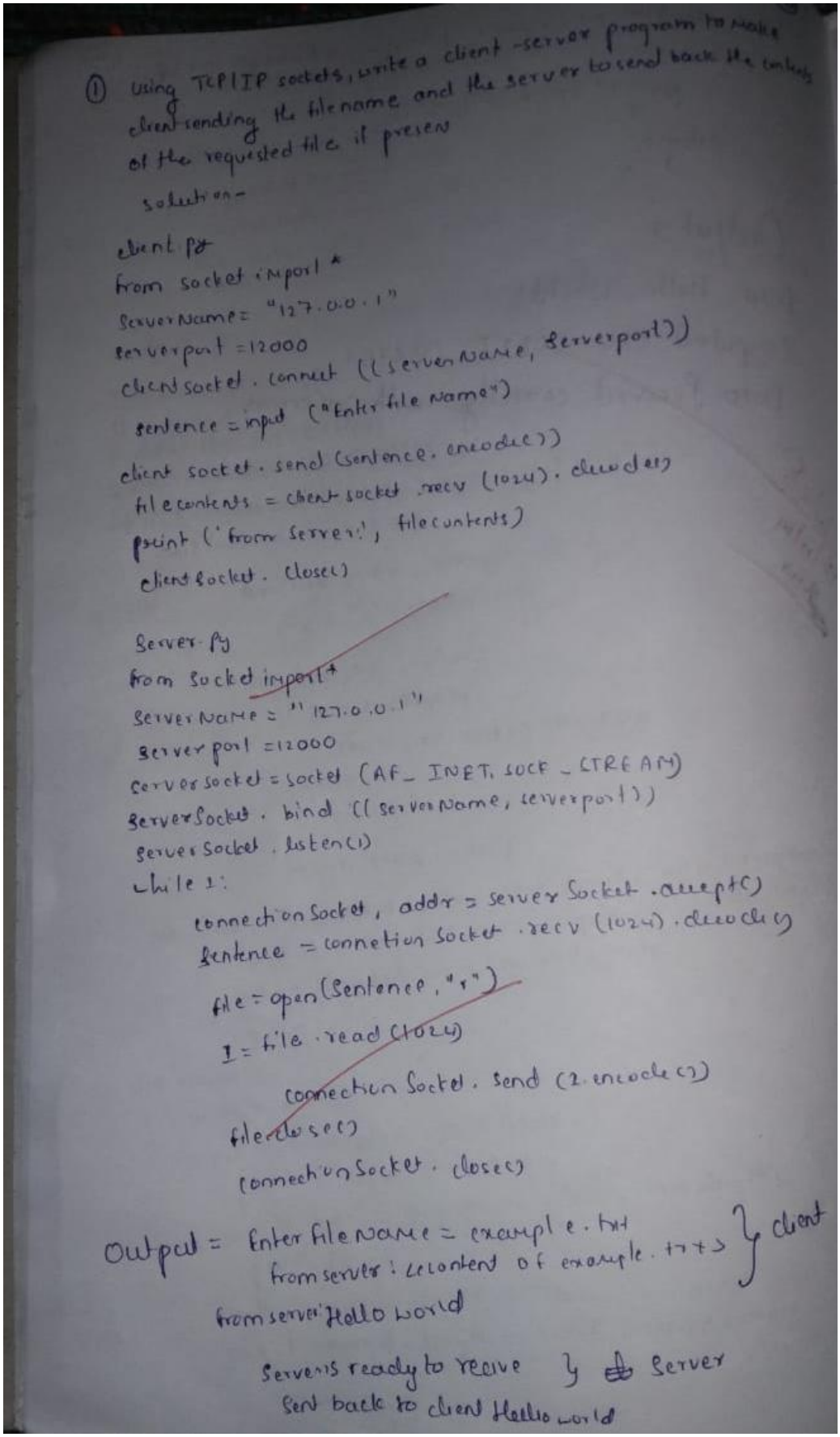
Server.py

```
from socket import *
serverName=”127.0.0.1”
serverPort = 12000
serverSocket = socket(AF_INET,SOCK_STREAM)
serverSocket.bind((serverName,serverPort))
serverSocket.listen(1)
print ("The server is ready to receive")
while 1:
connectionSocket, addr = serverSocket.accept()
sentence = connectionSocket.recv(1024).decode()

file=open(sentence,"r")
l=file.read(1024)

connectionSocket.send(l.encode())
file.close()
connectionSocket.close()
```

OUTPUT:=

```
The server is ready to receive
sent back to client hello world!!

Enter file nameexample.txt
From Server: b'hello world!!'
```

Observation:=



① Using TCP/IP sockets, write a client-server program to make client sending the filename and the server to send back the contents of the requested file if present

solution -

client.py

```
from socket import *
serverName = "127.0.0.1"
serverport = 12000
clientSocket.connect ((serverName, serverport))
sentence = input ("Enter file name")
clientsocket.send (sentence.encode())
filecontents = clientsocket.recv (1024).decode()
print ('from Server:', filecontents)
clientSocket.close()
```

Server.py

```
from Socket import*
ServerName = "127.0.0.1"
serverport = 12000
serversocket = socket (AF_INET, SOCK_STREAM)
serverSocket.bind ((serverName, serverport))
serverSocket.listen(1)
while 1:
    connectionSocket, addr = serverSocket.accept()
    sentence = connectionSocket.recv (1024).decode()
    file = open(sentence, "r")
    l = file.read (1024)
        connectionSocket.send (l.encode())
    file.close()
    connectionSocket.close()
```

Output = Enter file name = example.txt
from server: Content of example.txt } client
from server Hello World

Server is ready to receive } Server
Sent back to client Hello world

2. **Using UDP sockets, write a client-server program to make client sending the file name and the server to send back the contents of the requested file if present.**

program:=

ClientUDP.py

```
from socket import *
serverName = "127.0.0.1"
serverPort = 12000
clientSocket = socket(AF_INET, SOCK_DGRAM)

sentence = input("Enter file name")
clientSocket.sendto(bytes(sentence,"utf-8"),(serverName, serverPort))
filecontents,serverAddress = clientSocket.recvfrom(2048)
print ('From Server:', filecontents)

clientSocket.close()
```

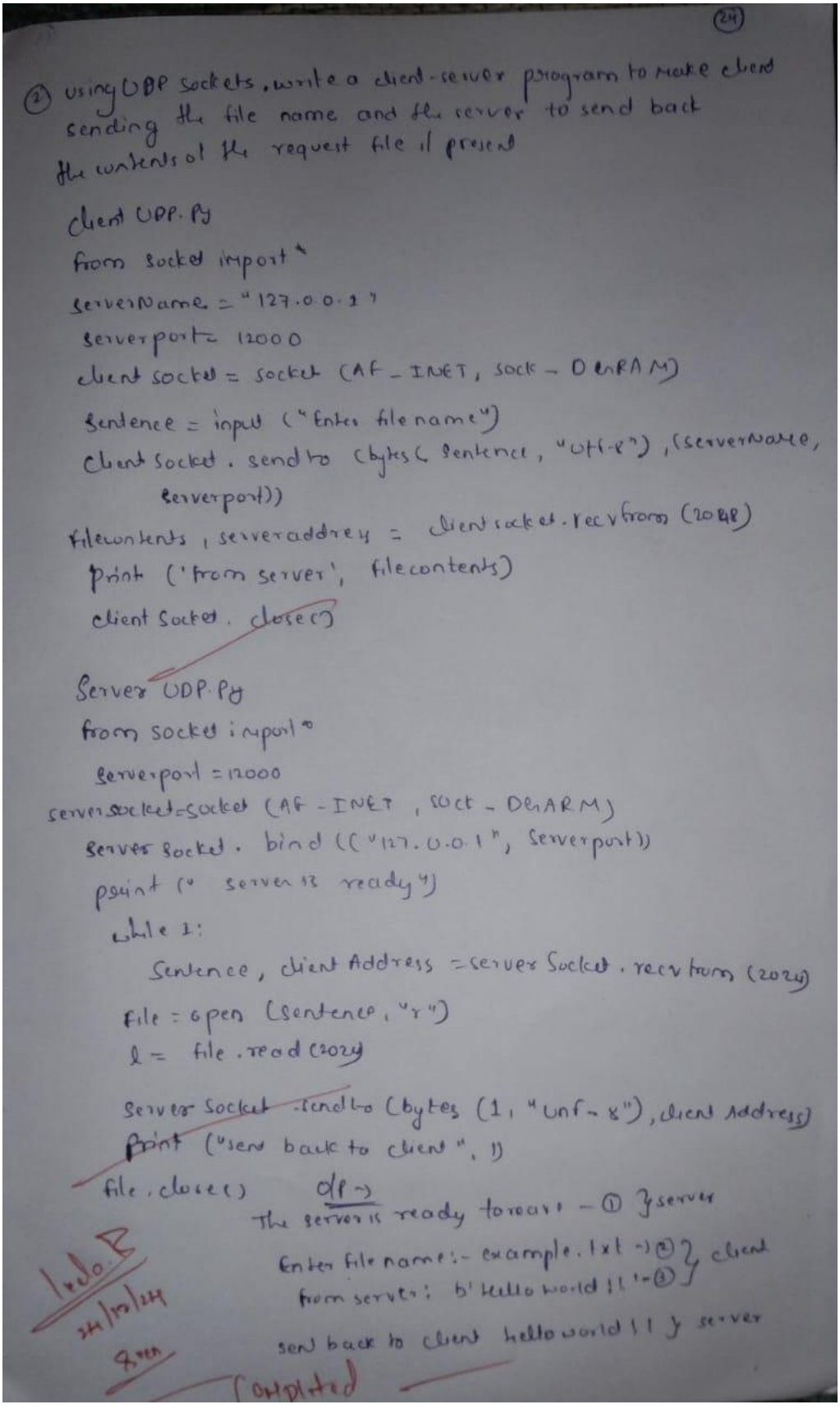
ServerUDP.py

```
from socket import *
serverPort = 12000
serverSocket = socket(AF_INET, SOCK_DGRAM)
serverSocket.bind(("127.0.0.1", serverPort))
print ("The server is ready to receive")
while 1:
    sentence,clientAddress = serverSocket.recvfrom(2048)

    file=open(sentence,"r")
    l=file.read(2048)

    serverSocket.sendto(bytes(l,"utf-8"),clientAddress)
    print("sent back to client",l)
file.close()
```

Output:=

```
The server is ready to receive
sent back to client hello world!!


Enter file nameexample.txt
From Server: b'hello world!!'
```

Observation:=



(24)

② using UDP sockets, write a client-server program to make client sending the file name and the server to send back the contents of the request file if present

Client UDP.Py

```
from socket import *
serverName = "127.0.0.1"
serverport = 12000
clientsocket = socket (AF_INET, sock_DGRAM)
Sentence = input ("Enter filename")
ClientSocket.sendto (bytes( Sentence, "utf-8"), (serverName,
    serverport))
filecontents, serveraddress = clientsocket.recvfrom (2048)
print ('from server', filecontents)
clientSocket.close()
```

Server UDP.Py

```
from socket import*
serverport = 12000
serversocket=socket (AF_INET, SOCK_DGRAM)
serverSocket.bind (("127.0.0.1", serverport))
print ("server is ready")
while 1:
    Sentence, clientAddress = serverSocket.recvfrom (2024)
    File = open (sentence, "r")
    l = file.read (2024)
    serverSocket.sendto (bytes (1, "unf-8"), clientAddress)
    print ("send back to client", l)
file.close()
```

o/p ->

The server is ready to recv - ① } server

Enter file name:- example.txt -② } client

from server: b'Hello world!!'-③

send back to client Hello world!! } server

Lab-8 24/10/24

Completed