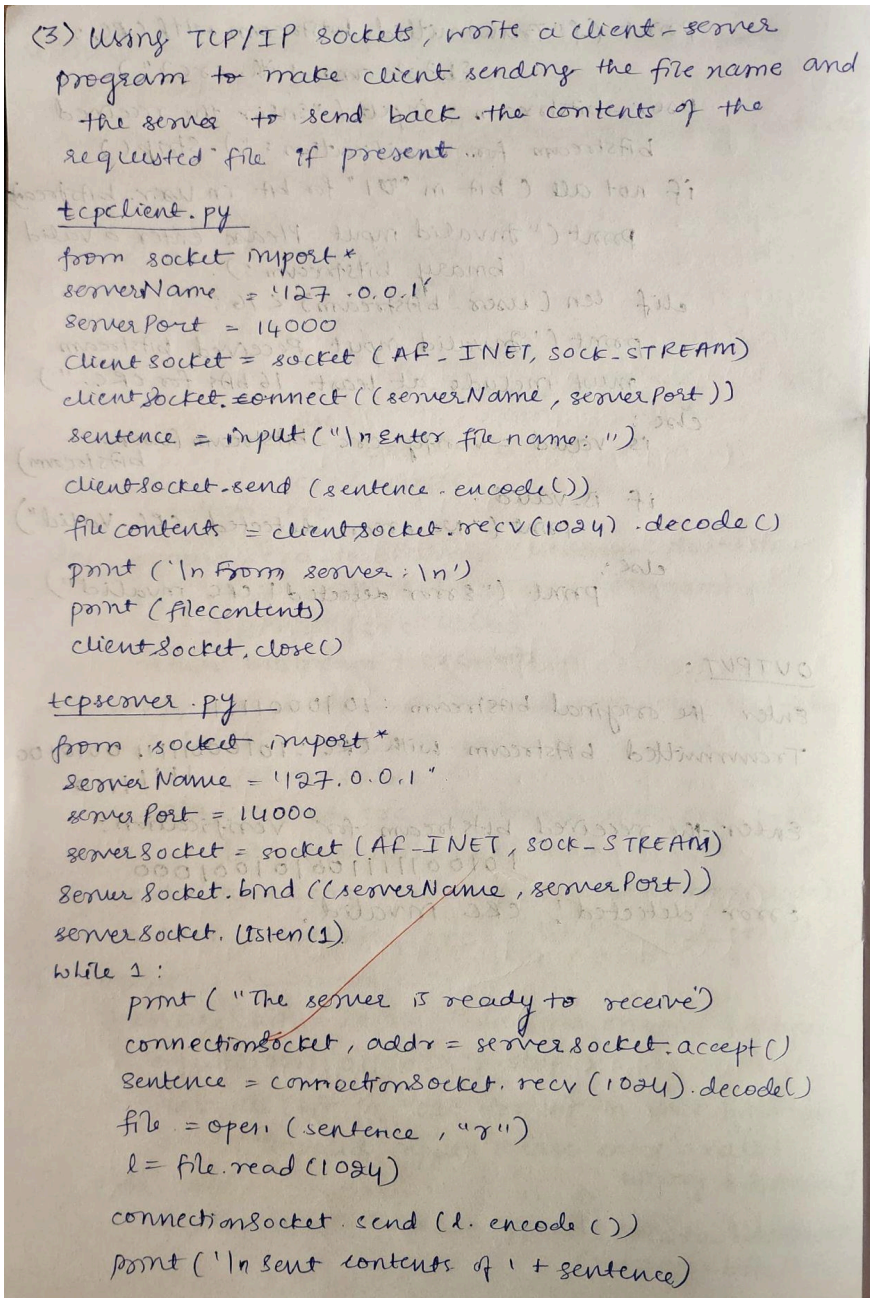
1/1/25 Experiment - 15

Program 3:

Question:

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.



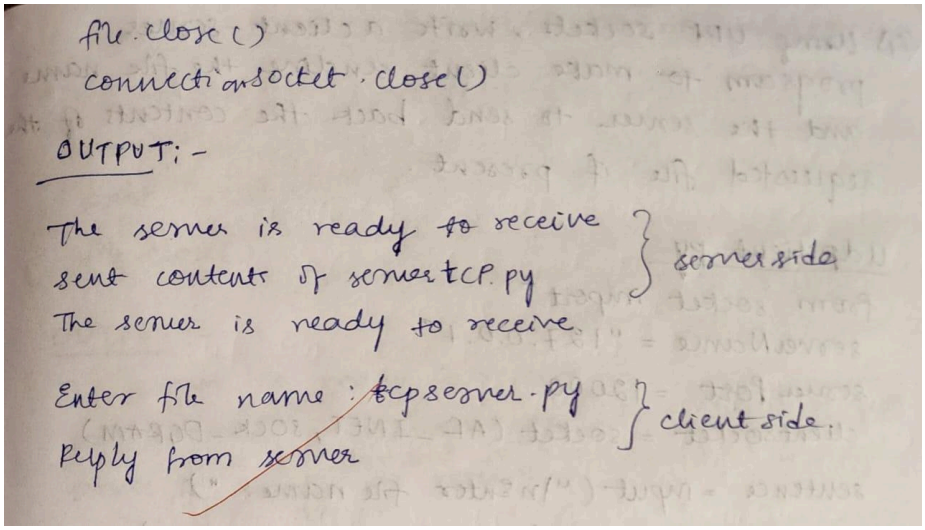
(3) Using TCP/IP sockets, write a client-server program to make client sending the file name and the server to send back the contents of the requested file if present..

tcpclient.py

```
from socket import *
serverName = '127.0.0.1'
serverPort = 14000
clientSocket = socket(AF_INET, SOCK_STREAM)
clientSocket.connect((serverName, serverPort))
sentence = input("\nEnter file name: ")
clientSocket.send(sentence.encode())
fileContents = clientSocket.recv(1024).decode()
print('\nFrom server:\n')
print(fileContents)
clientSocket.close()
```

tcpserver.py

```
from socket import *
serverName = '127.0.0.1'
serverPort = 14000
serverSocket = socket(AF_INET, SOCK_STREAM)
serverSocket.bind((serverName, serverPort))
serverSocket.listen(1)
while 1:
    print("The server is ready to receive")
    connectionSocket, addr = serverSocket.accept()
    sentence = connectionSocket.recv(1024).decode()
    file = open(sentence, "r")
    l = file.read(1024)
    connectionSocket.send(l.encode())
    print('\nSent contents of ' + sentence)
```



Code:

tcpserver.py file:

```
from socket import *
serverName="127.0.0.1"
serverPort = 14000
serverSocket = socket(AF_INET,SOCK_STREAM)
serverSocket.bind((serverName,serverPort))
serverSocket.listen(1)
while 1:
    print ("The server is ready to receive")
    connectionSocket, addr = serverSocket.accept()
    sentence = connectionSocket.recv(1024).decode()

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

    connectionSocket.send(l.encode())
    print ('\nSent contents of ' + sentence)
    file.close()
    connectionSocket.close()
```

tcpclient.py file:

```
from socket import *
serverName = '127.0.0.1'
serverPort = 14000
clientSocket = socket(AF_INET, SOCK_STREAM)
clientSocket.connect((serverName,serverPort))
sentence = input("\nEnter file name: ")

clientSocket.send(sentence.encode())
filecontents = clientSocket.recv(1024).decode()
print ('\nFrom Server:\n')
print(filecontents)
clientSocket.close()
```

Output:

```
PS C:\Users\Dell\OneDrive\Desktop\code> python tcpserver.py
The server is ready to receive

Sent contents of example.txt
The server is ready to receive
```

```
PS C:\Users\Dell\OneDrive\Desktop\code> python tcpclient.py

Enter file name: example.txt

From Server:

Hello, this is a sample file.
It is used for testing the TCP server.
```