WEEK 15

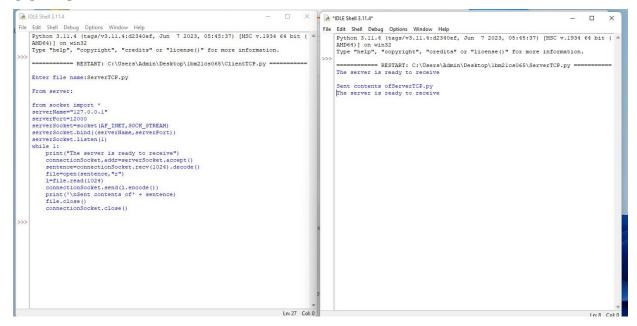
Using TCP/IP sockets, write a client-server program to make client sending the file name and the server to send back the contents of the requested file if present.

CODE:

```
ClientTCP.py
from socket import *
serverName = "127.0.0.1"
serverPort = 12000
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()
ServerTCP.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:
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()

OUTPUT:



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

CODE:

```
ClientUDP.py
from socket import *
serverName = "127.0.0.1"
serverPort = 12000
clientSocket = socket(AF_INET, SOCK_DGRAM)
sentence = input("\nEnter file name: ")
clientSocket.sendto(bytes(sentence,"utf-8"),(serverName, serverPort))
filecontents, serverAddress = clientSocket.recvfrom(2048)
print ("\nReply from Server:\n")
print (filecontents.decode("utf-8"))
# for i in filecontents:
# print(str(i), end = "")
clientSocket.close()
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)
sentence = sentence.decode("utf-8")
```

observation

Event ("the senior is ready to recive")
connection sousses addy - Somersound accept Servena = ConnectionSectual - vice Go 29) - decode; bele = open Csentine " ") connection socket send (1. encoders) Frent (" In sent contents of + sentence) 1910 - CLOSECY connectionspicket, close bugters Server Sede. The server is Ready to recive cleent sede Enter jele name servoitepopy ... From server: From Socketimport * Coode under servortop. Py is . Prented as wretten above) servor sede ? The servin es ready to relive content of server TCP. Py sent Servor o ready to Reuve The

	The state of the s
	Arm : Using DOP Socked write a circuit service to make circuit service Program to make circuit to be service to be service to the service to be serviced by the back the constant of the required by the
	16 pruni
	solution
)	client UPP-Py
	From socket impart * Servingame = "107 voice!" Berver Port = 12000
	client socked = Socket CAF INET, SOCK DONER sentence = input ("In Enter File name") client socket . Send to Chyter (sentence, "U+1-8")
	(Servindame , ServenPort)
	(2048)
	print (" le reply from seven 15") print (" le reply from seven 15") Fan: En fele contents
	prenty (str (8), end ="") cleent sound - close()
	Cleent Souket , close ()

Server upp- py from source empore " server Part = noon Server Source - brood (E" 127 D.D. 1" Server processory) Fronty (" The server is mody locative") 1 21 alles Sentence chent Address - servensour secondarion Sentence - sentence. decode ("or (. 89) file : ppin csentence "4") Server Socked = sended C bytes Con, " ut 1-8" cleen+Addrew) Prenty ("In sent mounts of the ende") Prent (sentince) for: insentence; prent (str (i) end: ') fele dose () bugtero gerver sode The sorvers as ready to posive sunt contents of serveruppe py The Server is I ready to recive