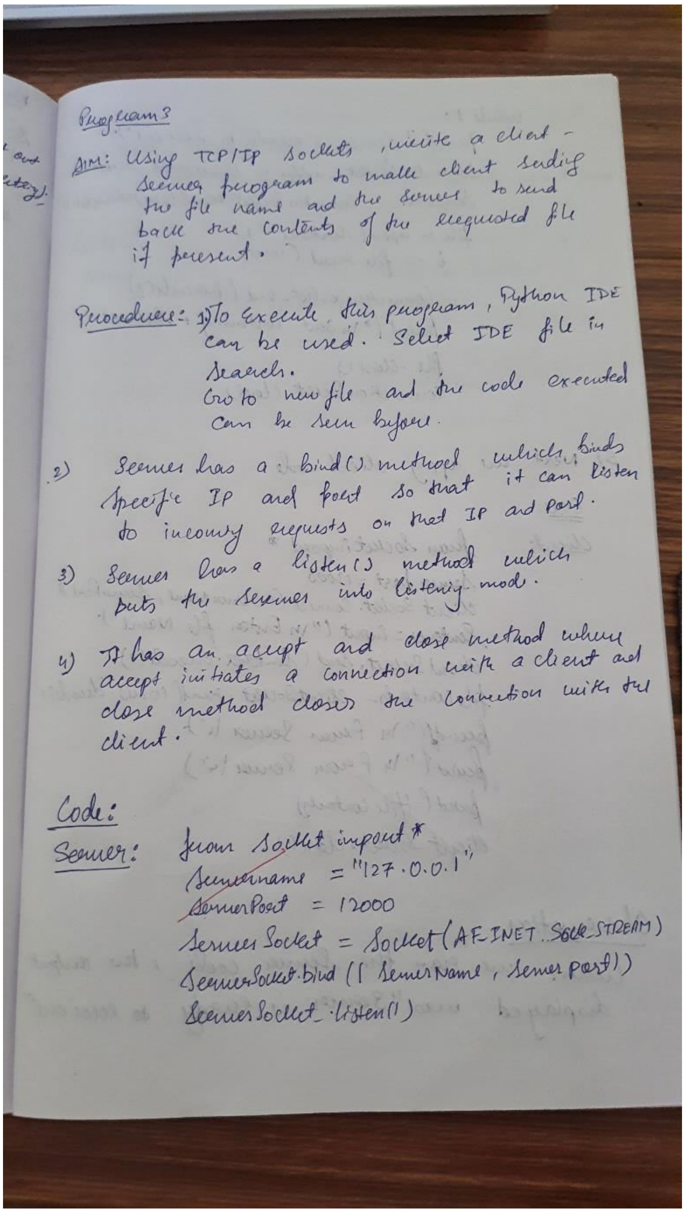
CN LAB 14

Program 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.

Observation:



Program 3

AIM: 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.

Procedure: 1) To Execute this program, Python IDE can be used. Select IDE file in search.
Go to new file and the code executed can be seen before.

2) Server has a bind() method which binds specific IP and port so that it can listen to incoming requests on that IP and port.

3) Server has a listen() method which puts the server into listening mode.

4) It has an accept and close method where accept initiates a connection with a client and close method closes the connection with the client.

Code:

Server:

```
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("Server is ready to recieve");
    connectionSocket, addr = serverSocket.accept()
    sentence = connectionSocket.recv(1024).decode()
    file = open(sentence, "r")
    l = file.read(1024)
     connectionSocket..send(l.encode())
     print("\n Sent contents of + sentence)
     file.close()
     connectionSocket.close()
```

5) Next we apply client code.

<u>Code:</u>

<u>Client:</u>

```
from socket import *
Server port =12000
client Socket. Connect((ServerName, ServerPort))
Sentence =input ("\n Enter file Name").
client Socket. Send (Sentence. encode()).
file contents= clientSocket . read(1024). decode()
printf("\n From Server \n").
print("\n From Server \n")
print(file contents)
client Socket. close()
```

<u>Observation</u>

When we ran the server code, the output displayed was "Server is ready to recieve".

Then we run the client code

→ Enter file name: new.py

→ From Serv: The whole contents was displayed

Output

Server: The Server is ready to recieve
Sent contents of new.py.

CODE:

ClientTCP.py

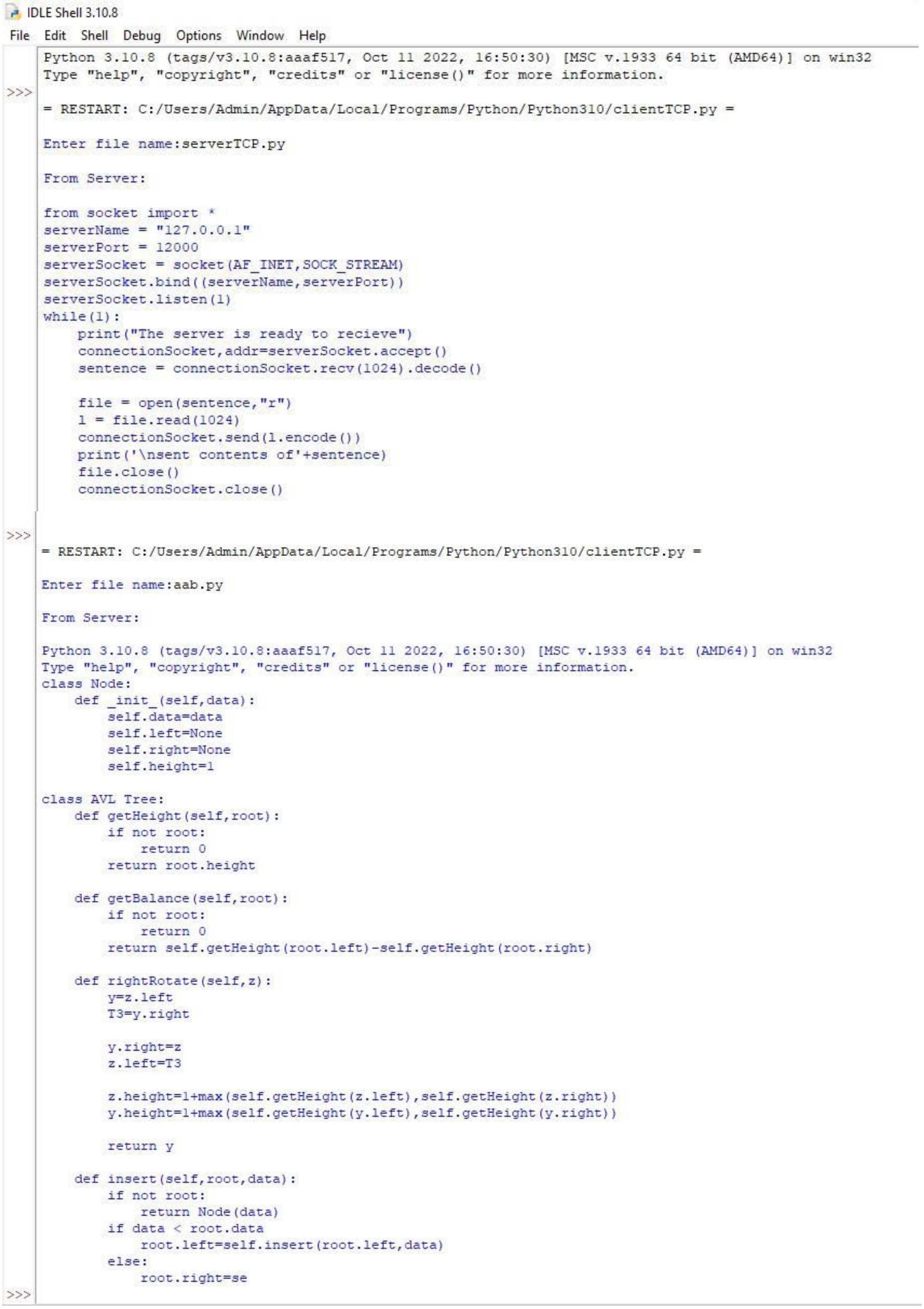
```
from socket import *
serverName = '127.0.0.1'
serverPort = 12000
clientSocket = socket(AF_INET, SOCK_STREAM)
clientSocket.connect((serverName,serverPort))
sentence = input("\nEnter file name: ")
clientSocket.send(sentence.encode())
filecontents = clientSocket.recv(1024).decode()
print ('\nFrom Server:\n') 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()
```

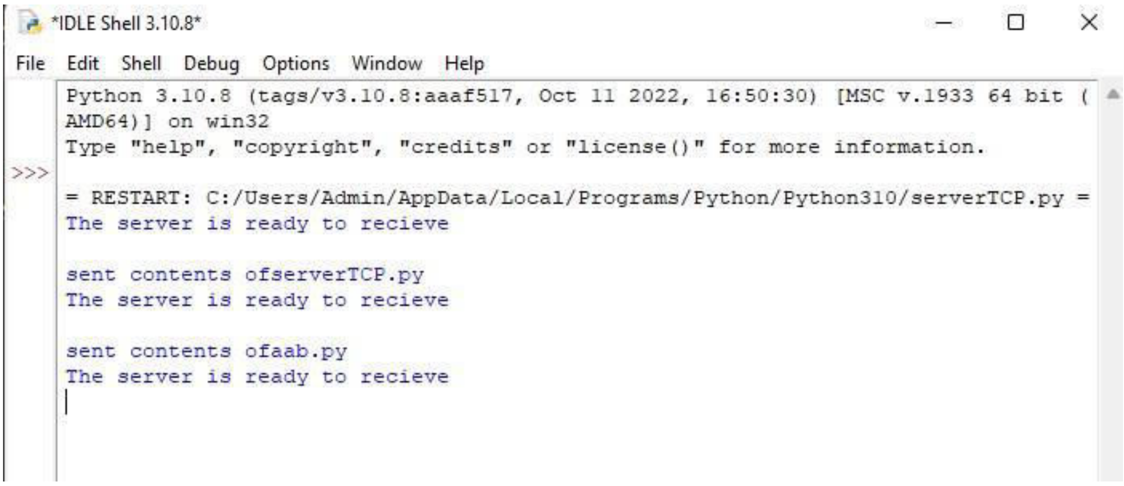
OUTPUT:

Client:



```
    Python 3.10.8 (tags/v3.10.8:aaaf517, Oct 11 2022, 16:50:30) [MSC v.1933 64 bit (AMD64)] on win32
    Type "help", "copyright", "credits" or "license()" for more information.
>>>
    = RESTART: C:/Users/Admin/AppData/Local/Programs/Python/Python310/clientTCP.py =

    Enter file name:serverTCP.py

    From Server:

    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 recieve")
        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()


>>>
    = RESTART: C:/Users/Admin/AppData/Local/Programs/Python/Python310/clientTCP.py =

    Enter file name:aab.py

    From Server:

    Python 3.10.8 (tags/v3.10.8:aaaf517, Oct 11 2022, 16:50:30) [MSC v.1933 64 bit (AMD64)] on win32
    Type "help", "copyright", "credits" or "license()" for more information.
    class Node:
        def _init_(self,data):
            self.data=data
            self.left=None
            self.right=None
            self.height=1

    class AVL Tree:
        def getHeight(self,root):
            if not root:
                return 0
            return root.height

        def getBalance(self,root):
            if not root:
                return 0
            return self.getHeight(root.left)-self.getHeight(root.right)

        def rightRotate(self,z):
            y=z.left
            T3=y.right

            y.right=z
            z.left=T3

            z.height=1+max(self.getHeight(z.left),self.getHeight(z.right))
            y.height=1+max(self.getHeight(y.left),self.getHeight(y.right))

            return y

        def insert(self,root,data):
            if not root:
                return Node(data)
            if data < root.data
                root.left=self.insert(root.left,data)
            else:
                root.right=se
>>>
```

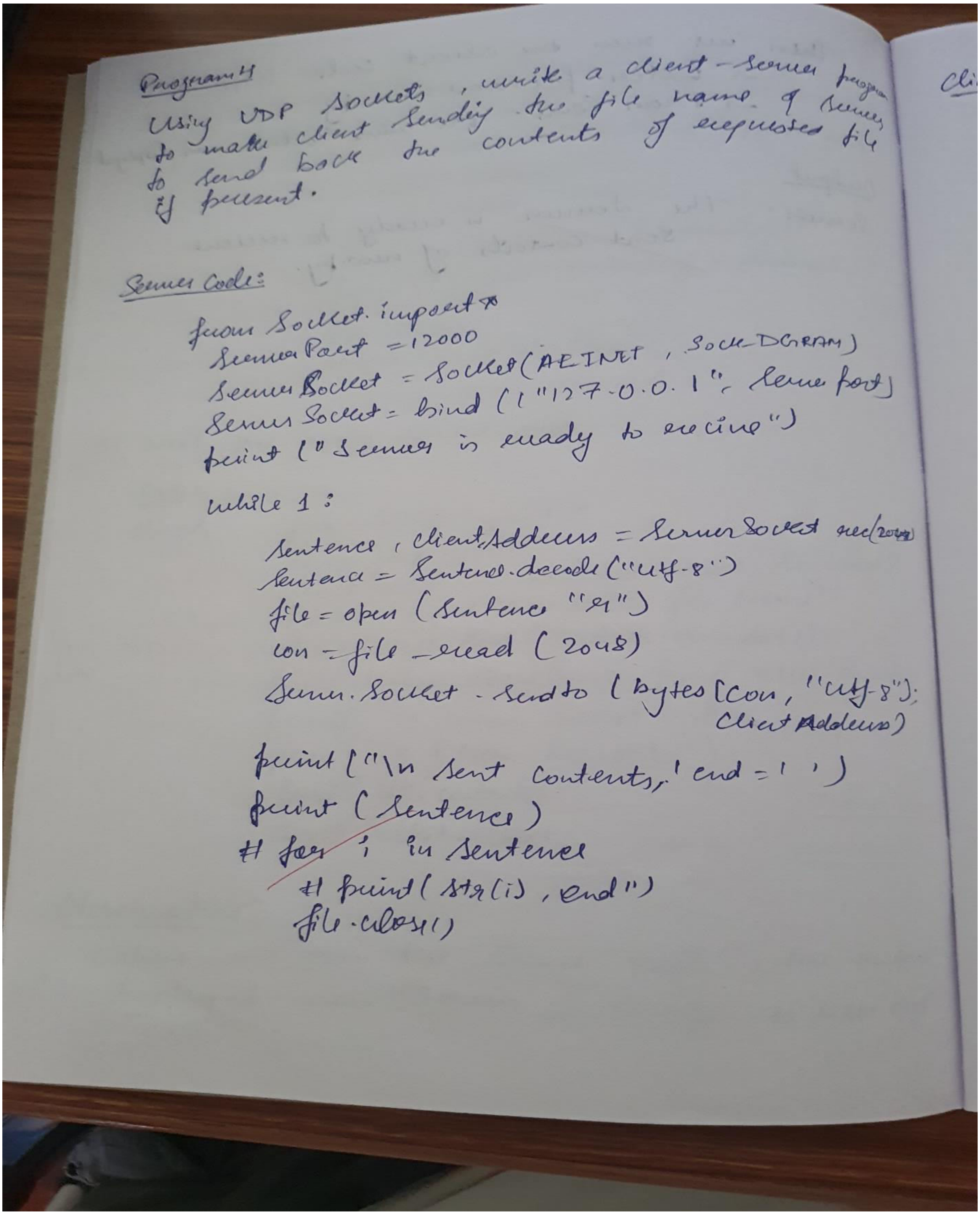
Server:



```
Python 3.10.8 (tags/v3.10.8:aaaf517, Oct 11 2022, 16:50:30) [MSC v.1933 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/Admin/AppData/Local/Programs/Python/Python310/serverTCP.py =
The server is ready to recieve

sent contents ofserverTCP.py
The server is ready to recieve

sent contents ofaab.py
The server is ready to recieve
```

Program 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.

Observation:



Program 4

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

Server Code:

```
from Socket import *
 ServerPort = 12000
 ServerSocket = Socket(AF_INET , SOCK_DGRAM)
 ServerSocket = bind (("127.0.0.1", ServerPort)
 print ("Server is ready to receive")

 while 1 :

     Sentence , clientAddress = ServerSocket recv(2048)
     Sentence = Sentence.decode ("utf-8")
     file = open (Sentence "r")
     con = file_read (2048)
     ServerSocket - sendto (bytes (con, "utf-8"),
                                      clientAddress)

    print ("\n Sent contents,' end = ' ')
    print (Sentence)
   # for i in Sentence
      # print( str(i) , end '')
      file.close()
```

Client code

```
from Socket. import *
Server Port = 12000
ServerSocket = Socket (AF_INET, SOCK_DGRAM)
ServerSocket = Bind ("127.0.0.1", Server port)
print ("Server is ready to recieve")

~~while 1~~:
    Sentence, Client Address = ServerSocket
  Client Socket _ Sendto ( bytes (Sentence, "utf-8");
  ( Server Name , Server Port )
  file contents , Server , Server Address = clientSocket
         recv from (2048)
   print ("\n Reply from Server \n");
   print ( file contents . decode ("utf-8"))
   #  for filecontent;
   #  print ( str (i) . end = '' )
          Client Socket . Close()
          client Socket . Close().
```

Observation

Outputs Client

→ Enter file name : Server . py.

→ Reply from Server : All the Contents were displayed.

Server : The Server ready to recieve sent contents of Server. py.

9/9/23

CODE:

ClientUDP.py

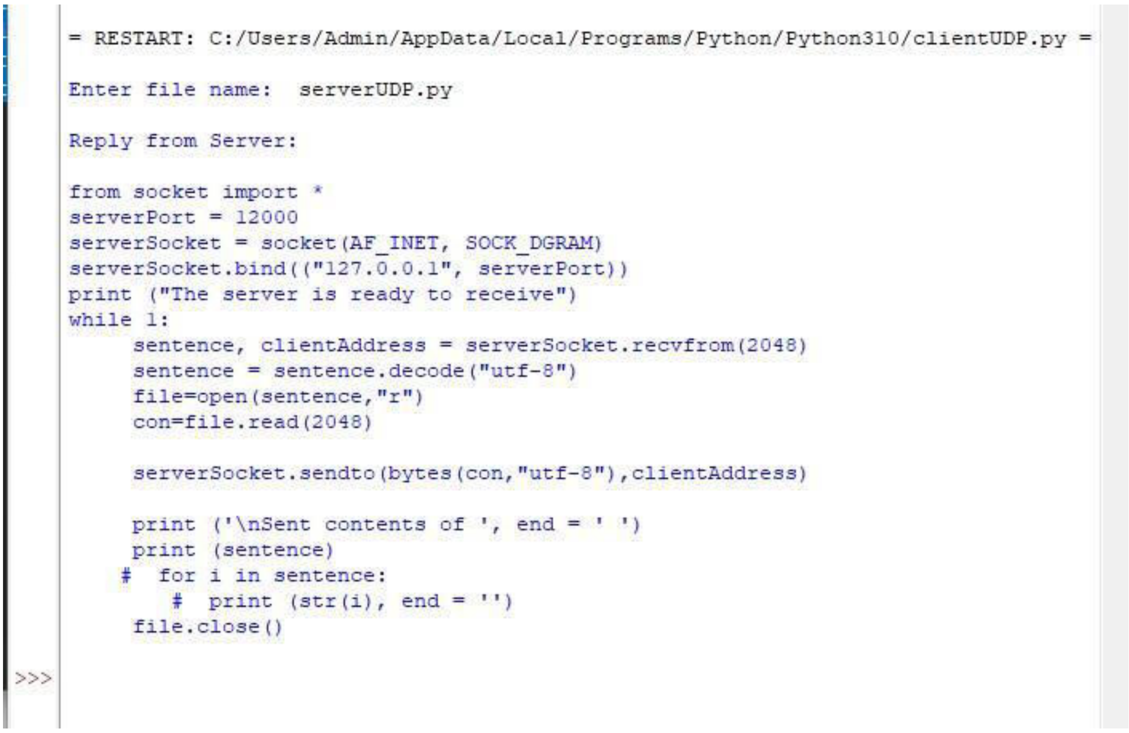
```
from socket import *
serverName = "127.0.0.1"
serverPort = 12000
clientSocket = socket(AF_INET, SOCK_DGRAM)
sentence = input("\nEnter file name: ")
clientSocket.sendto(bytes(sentence,"utf-8"),(serverName, serverPort))
filecontents,serverAddress = clientSocket.recvfrom(2048)
print ('\nReply from Server:\n')
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")
file=open(sentence,"r")
con=file.read(2048)
serverSocket.sendto(bytes(con,"utf-8"),clientAddress)
print ('\nSent contents of ', end = ' ')
print (sentence)
# for i in sentence:
# print (str(i), end = '')
file.close()
```

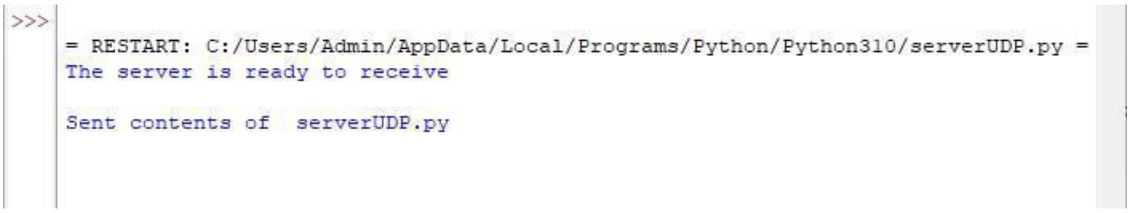
OUTPUT:

Client:



```
= RESTART: C:/Users/Admin/AppData/Local/Programs/Python/Python310/clientUDP.py =

Enter file name:  serverUDP.py

Reply from Server:

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

     serverSocket.sendto(bytes(con,"utf-8"),clientAddress)

     print ('\nSent contents of ', end = ' ')
     print (sentence)
    #  for i in sentence:
        #  print (str(i), end = '')
     file.close()

>>>
```

Server:



```
>>>
= RESTART: C:/Users/Admin/AppData/Local/Programs/Python/Python310/serverUDP.py =
The server is ready to receive

Sent contents of  serverUDP.py
```