Date __/__/__
Page _____

14 Write a program for congestion control using leaky bucket algorithm

```
import java.util.*;

class leakybucket
{
    public static void main (String [])
    args)
    {
        int rem;

Scanner sc = new Scanner (System.in);
    int s=0;

System.out.println ("enter no: of
queues, buffer size, input and
output bucket size");

int q = sc.nextInt();
int bs = sc.nextInt();
int ip = sc.nextInt()
int op = sc.nextInt();

for (int i=0; i < q; i++)
    {
        rem = bs - s;
        if (ip <= (rem))
    {
```

Date __/__/__
Page _____

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

ClientUDP.py

```
from socket import *
ServerName = "127.0.0.1"
serverPort = 12000
clientSocket = socket(AF_INET, SOCK_DGRAM
sentence = input("Input file name: ")

clientSocket.sendto(bytes(sentence, "utf-8"),
        (serverName, serverPort))
fileContents, serverAddress = clientSocket.
                recvfrom(2048)
print("In Reply 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_DG
serverSocket.bind(("127.0.0.1", serverPort
print("The server is ready to receive")
```

Date __/__/__
Page _____

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

Output

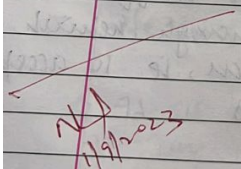
```
server UDP.py
The server is ready to receive

Sent contents of ServerUDP.py
The server is ready to receive
```

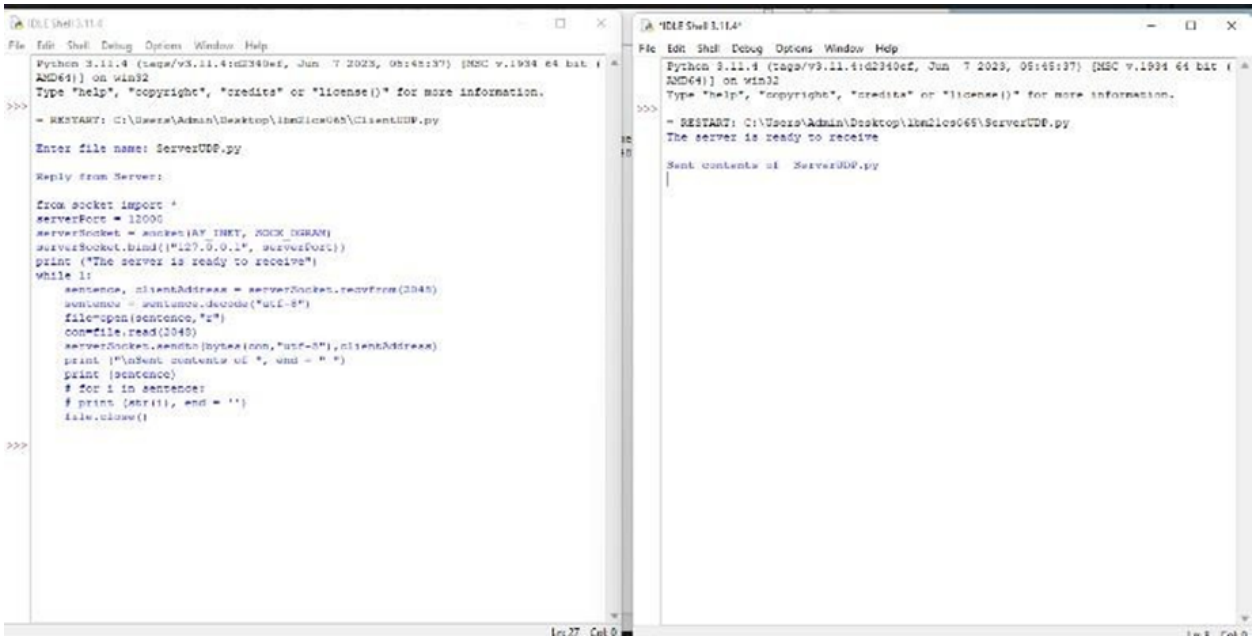
ClientUDP.py

```
Enter file name: serverUDP.py

Reply from Server:

from socket import *
serverPort = 12000
serverSocket = socket(AF_INET, SOCK_D
serverSocket.bind(("127.0.0.1", serverPort))
```

Date __/__/__
Page

```
While 1:
print ("The server is ready to receive")
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 ("\n Sent contents of ", end = "")
print (sentence)
  # for i in sentence:
  # print (str (i), end = "")
    file.close()
```




File Edit Shell Debug Options Window Help
Python 3.11.4 (tags/v3.11.4:d2340ef, Jun 7 2023, 05:45:37) [MSC v.1934 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\Admin\Desktop\ibm21cs065\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()
>>>
'IDLE Shell 3.11.4'
File Edit Shell Debug Options Window Help
Python 3.11.4 (tags/v3.11.4:d2340ef, Jun 7 2023, 05:45:37) [MSC v.1934 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\Admin\Desktop\ibm21cs065\ServerUDP.py
The server is ready to receive
Sent contents of ServerUDP.py