**SIDDHI SINGH**

**17BIT0028**

**ASSESSEMENT-3**

1. **The finance office of VIT wishes to make the transactions more secured. If you are a programmer how you will implement a system to validate the login credentials obtained from the user thereby denying the access to unauthorized users using UDP sockets.**

**c.py**

import socket

UDP\_IP="127.0.0.1"

UDP\_PORT=5007

user\_id=input("Please Enter the user-id:")

sock=socket.socket(socket.AF\_INET,socket.SOCK\_DGRAM)

sock.sendto(user\_id.encode(),(UDP\_IP,UDP\_PORT))

if((sock.recv(1024).decode())=="Exist"):

password=input("Please enter password:")

sock.sendto(password.encode(),(UDP\_IP,UDP\_PORT))

print(sock.recv(1024).decode())

else:

print(sock.recv(1024).decode())

**s.py**

import socket

UDP\_IP="127.0.0.1"

UDP\_PORT=5007

sock=socket.socket(socket.AF\_INET,socket.SOCK\_DGRAM)

sock.bind((UDP\_IP,UDP\_PORT))

**check={'Aradhya':'Mathur','admin':'password','17BIT0146':'aradhya'}**

data,addr=sock.recvfrom(1024)

t1=data.decode()

while True:

if(t1 in check):

print("User\_id Exist")

sock.sendto("Exist".encode(),addr)

password,addr=sock.recvfrom(1024)

pass1=password.decode()

if(check[t1]==pass1):

sock.sendto("Authenticated ".encode(),addr)

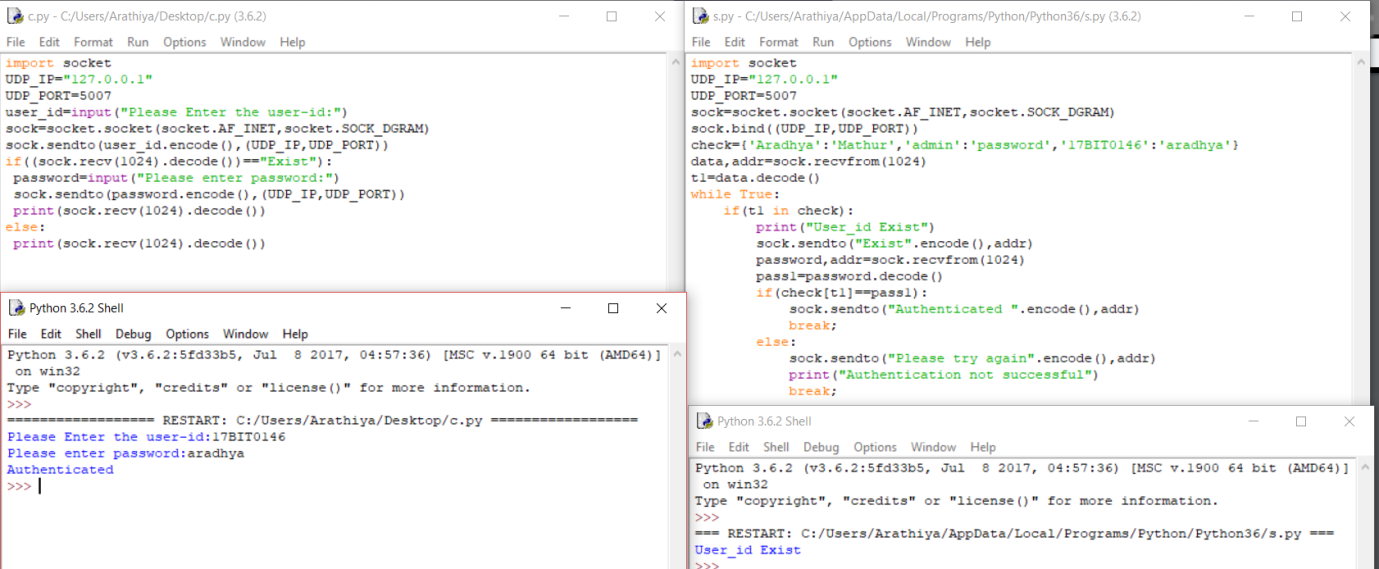
break;

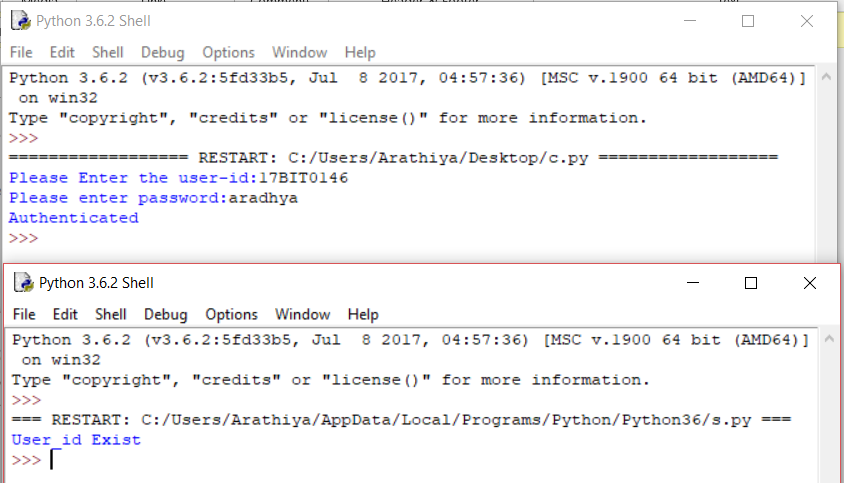
else:

sock.sendto("Please try again".encode(),addr)

print("Authentication not successful")

break;





**2. Write a program to implement a image based message transfer from client to server process**

**using UDP.**

**Client.py**

#17BIT0028

import random

import socket, select

from time import gmtime, strftime

from random import randint

image = "tux.png"

HOST = '127.0.0.1'

PORT = 6666

sock = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM)

server\_address = (HOST, PORT)

sock.connect(server\_address)

try:

# open image

myfile = open(image, 'rb')

bytes = myfile.read()

size = len(bytes)

# send image size to server

sock.sendall("SIZE %s" % size)

answer = sock.recv(4096)

print 'answer = %s' % answer

# send image to server

if answer == 'GOT SIZE':

sock.sendall(bytes)

# check what server send

answer = sock.recv(4096)

print 'answer = %s' % answer

if answer == 'GOT IMAGE' :

sock.sendall("BYE BYE ")

print 'Image successfully send to server'

myfile.close()

finally:

sock.close()

**Server.py**

#17BIT0028

import random

import socket, select

from time import gmtime, strftime

from random import randint

imgcounter = 1

basename = "image%s.png"

HOST = '127.0.0.1'

PORT = 6666

connected\_clients\_sockets = []

server\_socket = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM)

server\_socket.setsockopt(socket.SOL\_SOCKET, socket.SO\_REUSEADDR, 1)

server\_socket.bind((HOST, PORT))

server\_socket.listen(10)

connected\_clients\_sockets.append(server\_socket)

while True:

read\_sockets, write\_sockets, error\_sockets = select.select(connected\_clients\_sockets, [], [])

for sock in read\_sockets:

if sock == server\_socket:

sockfd, client\_address = server\_socket.accept()

connected\_clients\_sockets.append(sockfd)

else:

try:

data = sock.recv(4096)

txt = str(data)

if data:

if data.startswith('SIZE'):

tmp = txt.split()

size = int(tmp[1])

print 'got size'

sock.sendall("GOT SIZE")

elifdata.startswith('BYE'):

sock.shutdown()

else :

myfile = open(basename % imgcounter, 'wb')

myfile.write(data)

data = sock.recv(40960000)

if not data:

myfile.close()

break

myfile.write(data)

myfile.close()

sock.sendall("GOT IMAGE")

sock.shutdown()

except:

sock.close()

connected\_clients\_sockets.remove(sock)

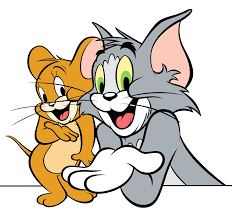
continue

imgcounter += 1

server\_socket.close()

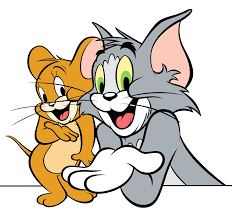
**Sent image**

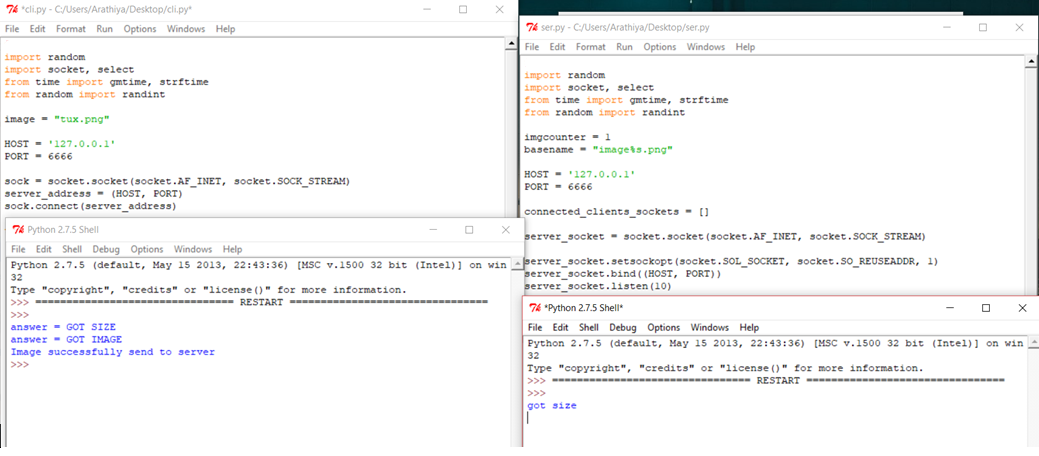
**tux.png**

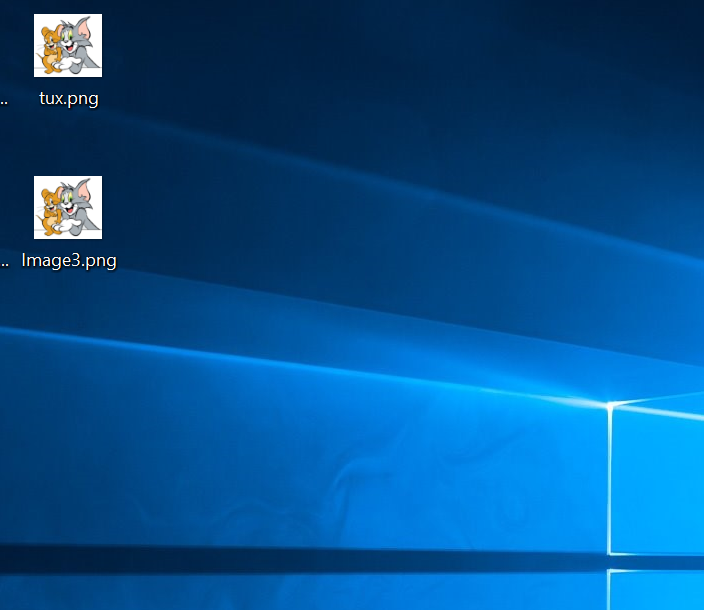


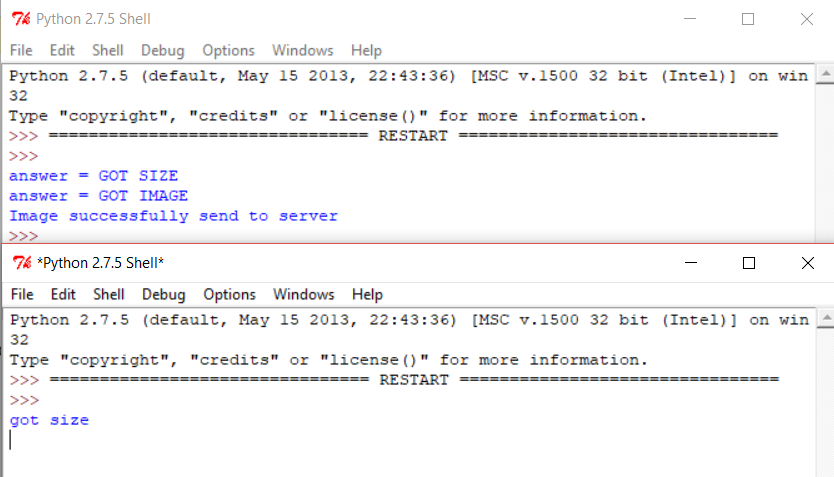
**Received image**

**Image3.png**









**Ques3 Find the logical address of a host when its physical address is known (RARP protocol) using UDP.**

**Server.java**

import java.io.\*; import java.net.\*; import java.util.\*; public class Server{

public static void main(String args[]) { try{

DatagramSocket server = new DatagramSocket(1309); while(true){

byte[] sendByte = new byte[1204]; byte[] receiveByte = new byte[1204]; DatagramPacket receiver = new

DatagramPacket(receiveByte,receiveByte.length); server.receive(receiver);

String str = new String(receiver.getData()); String s = str.trim();

InetAddress addr = receiver.getAddress(); int port = receiver.getPort();

String ip[] = {"108.10.31.68"};

String mac[] = {"A4:9D:E6:121:A8:D1"};

for (int i = 0; i < ip.length; i++) { if(s.equals(mac[i]))

{

sendByte = ip[i].getBytes(); DatagramPacket sender = new DatagramPacket(sendByte,sendByte.length,addr,port);

server.send(sender); break;

}

}

break;

}

}catch(Exception e)

{

System.out.println(e);

}

}

}

# Client.java

import java.io.\*; import java.net.\*; import java.util.\*;

public class RARPClient {

public static void main(String args[]){ try{

DatagramSocket client = new DatagramSocket(); InetAddress addr = InetAddress.getByName("127.0.0.1"); byte[] sendByte = new byte[1204]; byte[] receiveByte = new byte[1024];

BufferedReader in = new BufferedReader(new InputStreamReader(System.in)); System.out.println("Enter the Physical Address "); String str = in.readLine(); sendByte = str.getBytes();

DatagramPacket sender = new DatagramPacket(sendByte,sendByte.length,addr,1309); client.send(sender);

DatagramPacket receiver = new DatagramPacket(receiveByte,receiveByte.length); client.receive(receiver);

String s = new String(receiver.getData()); System.out.println("The Logical Address is :" + s.trim()); client.close();

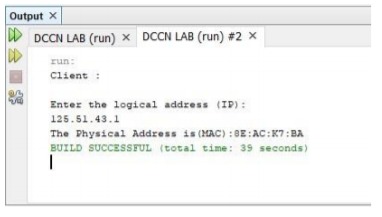
}

catch(Exception e){ System.out.println(e);

}

}

}



**4. Implement a UDP based socket program for the game “Guess It”. The game is played by client**

**and server. The game proceeds as follows: The server will think a “magic number”. Magic numberis greater than zero and less than 100. Prompt for the “magic number” from the client. The servershould print out “Higher” if the magic number is higher than the guess and “Lower” if the magicnumber is less than the guess. Give 10 chances for the client to guess the number. When the magicnumber has been guessed print "Great" and then end. If the client cannot guess the number in 10attempts then print the message "Better luck next time:)" and end.**

**ANSWER 4**

**Client.py**

#17BIT0028

import socket

s=socket.socket()

host=socket.gethostname()

port=12345

magic = str(input("Enter the magic number within 0 and 100: "))

s.connect((host,port))

s.send(magic.encode())

res = s.recv(1024).decode()

print(res)

s.close

**Server.py**

#17BIT0028

import socket

import random

s=socket.socket()

host=socket.gethostname()

port=12345

s.bind((host,port))

s.listen(5)

count = 0

while True:

c,addr = s.accept()

r = c.recv(1024).decode()

print("Got connection from ",addr)

rand = random.choice(range(0,100))

print("Random magic number within 0 and 100 ",rand)

if(int(r)==rand):

msg = "\nGreat"

break

if(int(r)>rand):

msg = "\nHigher"

if(int(r)<rand):

msg = "\nLower"

count=count+1

if(count==10):

msg = "\nBetter luck next time"

c.send(msg.encode())

c.close()

