Name: Preyash Date: 03-02-2022

Registration Number: 20BPS1022

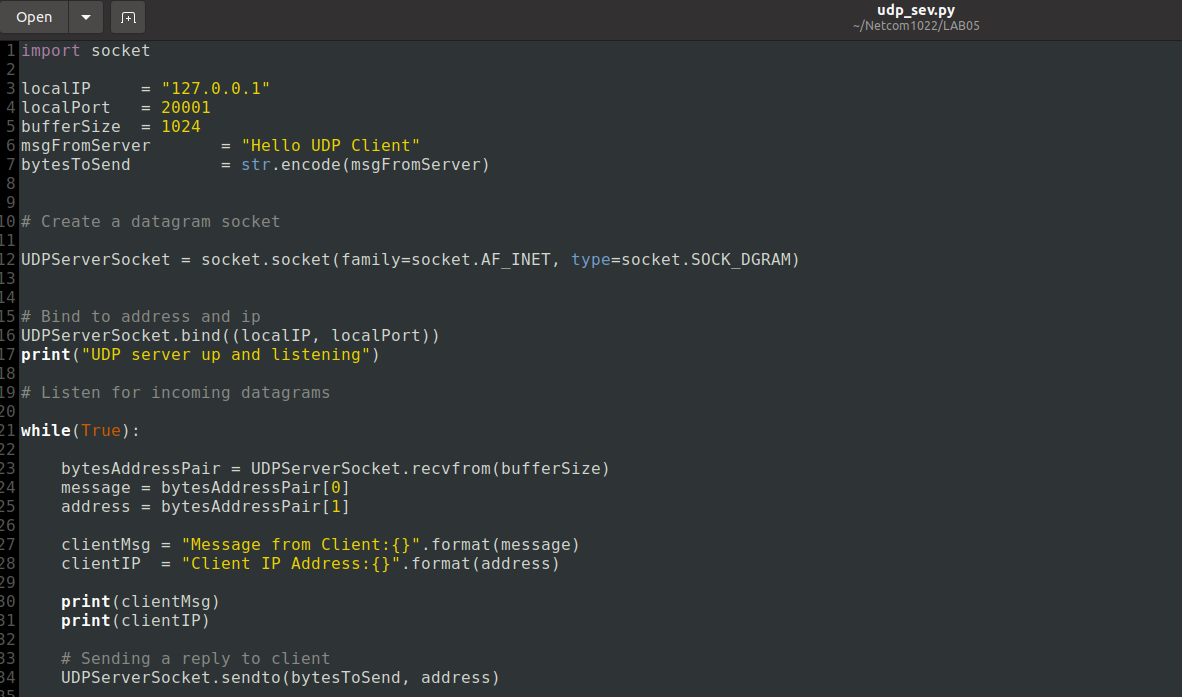
**LAB-05**

**UDP Socket Programming**

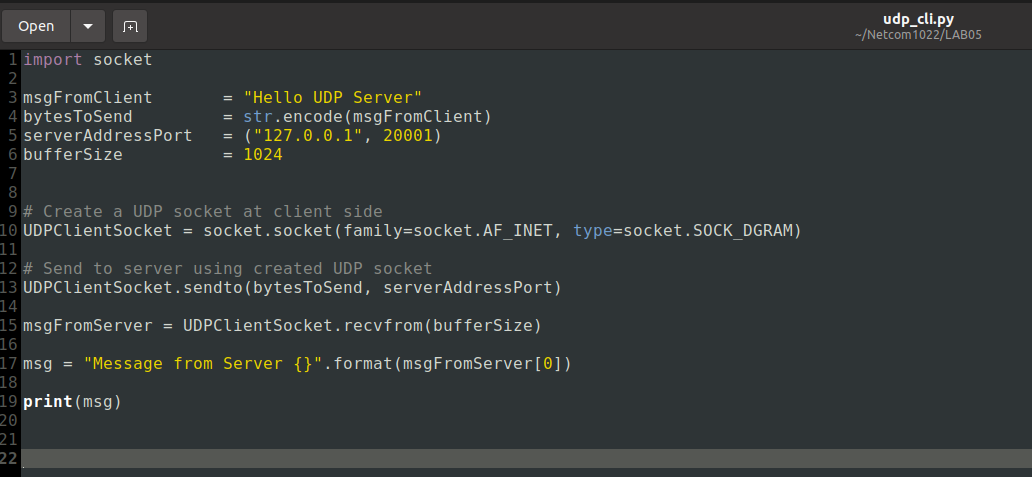
**AIM:** To get familiar with UDP socket programming and to build a key server that serves clients with a client-supplied encoded key.

**Basic program:** Basic program to get familiar with UDP concepts.

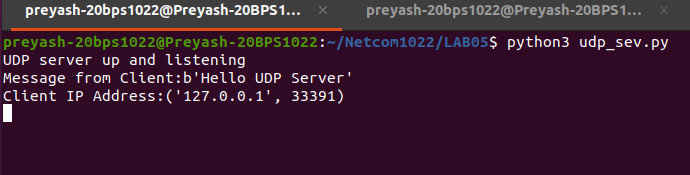
**Server-side code:**



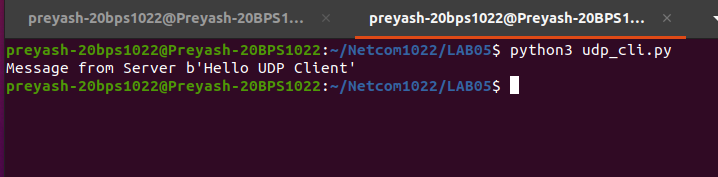
**Client-side code:**



**Server-side output:**



**Client-side output:**



**TASK:**

**Send the following statement from the client and do the message encoding on the server side and display it on the client side.**

**“Please Switch Off the Lights and Fan When Not in Use” replace the vowels with their next vowel for example “a” with “e”, “e” with “i” and so on. Similarly replace the consonants with the next consonant in order. Retain the Capital letters even after encoding.**

**ALGORITHM**

**Server-side Algorithm:**

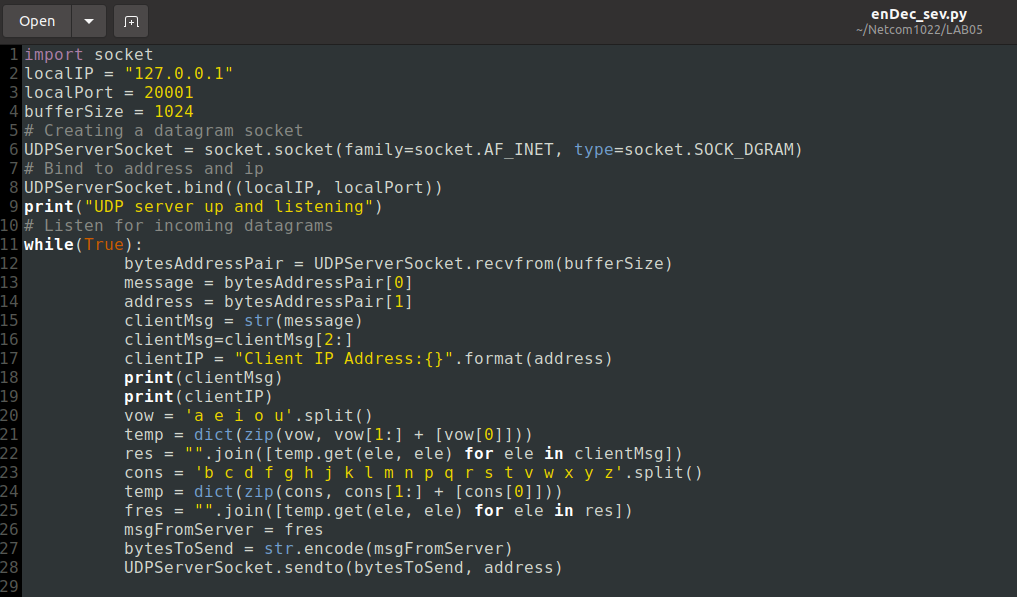
* Connect to a port
* Wait for the client to connect.
* Once connected, receive the client's key and encode it before returning the encoded key.
* Disconnect

**Client-Side Algorithm:**

* Connect to port
* Send key
* Wait for response and receive it
* Print the response

**Server Program Source Code:**

**Code window:**

****

**Code:**

import socket

localIP = "127.0.0.1"

localPort = 20001

bufferSize = 1024

# Creating a datagram socket

UDPServerSocket = socket.socket(family=socket.AF\_INET, type=socket.SOCK\_DGRAM)

# Bind to address and ip

UDPServerSocket.bind((localIP, localPort))

print("UDP server up and listening")

# Listen for incoming datagrams

while(True):

bytesAddressPair = UDPServerSocket.recvfrom(bufferSize)

message = bytesAddressPair[0]

address = bytesAddressPair[1]

clientMsg = str(message)

clientMsg=clientMsg[2:]

clientIP = "Client IP Address:{}".format(address)

print(clientMsg)

print(clientIP)

vow = 'a e i o u'.split()

temp = dict(zip(vow, vow[1:] + [vow[0]]))

res = "".join([temp.get(ele, ele) for ele in clientMsg])

cons = 'b c d f g h j k l m n p q r s t v w x y z'.split()

temp = dict(zip(cons, cons[1:] + [cons[0]]))

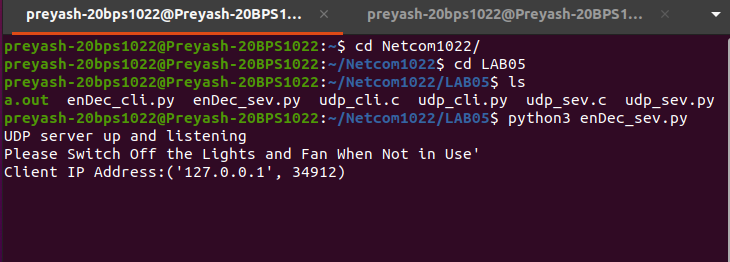
fres = "".join([temp.get(ele, ele) for ele in res])

msgFromServer = fres

bytesToSend = str.encode(msgFromServer)

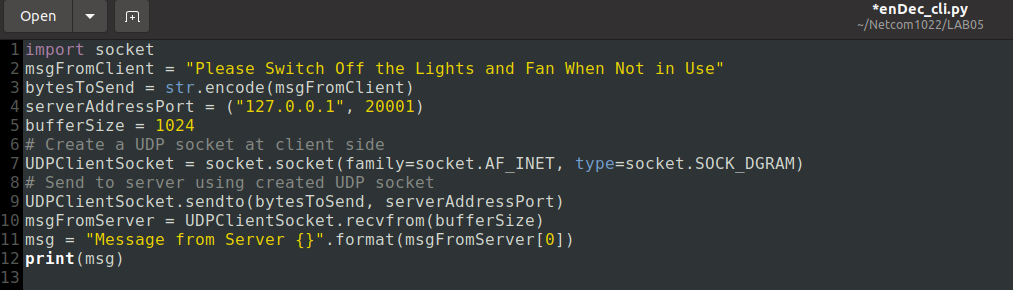
UDPServerSocket.sendto(bytesToSend, address)

**Output:**

****

**Client Program Source Code:**

**Code window:**

****

**Code:**

import socket

msgFromClient = "Please Switch Off the Lights and Fan When Not in Use"

bytesToSend = str.encode(msgFromClient)

serverAddressPort = ("127.0.0.1", 20001)

bufferSize = 1024

# Create a UDP socket at client side

UDPClientSocket = socket.socket(family=socket.AF\_INET, type=socket.SOCK\_DGRAM)

# Send to server using created UDP socket

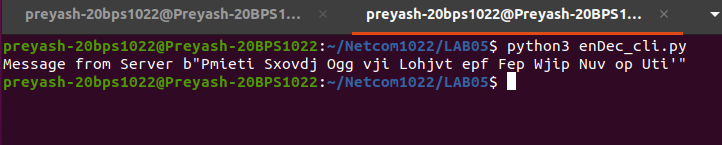
UDPClientSocket.sendto(bytesToSend, serverAddressPort)

msgFromServer = UDPClientSocket.recvfrom(bufferSize)

msg = "Message from Server {}".format(msgFromServer[0])

print(msg)

**Output:**

****

**Result:** We successfully created a program to perform the required output.