Please note that these are only high level instructions to help you navigate through the lab

1. Basics of SMTP (Simple Mail Transfer Protocol)
    a. An SMTP server is an application that's primary purpose is to send, receive, and/or relay outgoing mail between email senders and receivers.
    b. When you send an email, the SMTP server processes your email, decides which server to send the message to, and relays the message to that server. The recipient's inbox service provider, such as Gmail or AOL then downloads the message and places it in the recipient's inbox.
    c. SMTP server vs regular server
        i. Server is a generic term, which means a program on machine serving the clients request. The purpose of SMTP server is specifically to serve request pertaining to emails.
    d. Your mail client has a conversation with our server, first verifying that the username and password used for authentication correspond to an active account, then relaying the message information, such as sender address, recipient, and message content.
    e. We are supposed to send an email using the SMTP server provided by Linux.

We need to work on this segment

Client -> SMTP server -> Recipient

2. Part 1:
    a. Run the SMTP server on Linux in debugging mode
        i. python -m smtpd -c DebuggingServer -n localhost:6000
    b. Connect to the server
    c. In general: Prepare the message -> Send the encoded SMTP message thought the socket created-> Receive the SMTP response through the socket.
    d. Always encode the messages while sending it to the server, and decode the messages while receiving them
    e. Always print the messages sent and received. This will help you debug, in case of any errors.
    f. Send HELO along with the name of the server
        ii. For example, 'HELO localhost.com\r\n'
    g. Similarly, Send *MAIL FROM* (Refer to the diagram given the Lab manual)
    h. Send *RCPT TO*
    i. Send DATA
    j. Send QUIT
    k. Check if the server sends correct responses

3. Part 2
    a. Create a dummy gmail account for the purposes of this lab.
    b. Go to https://myaccount.google.com/lesssecureapps, and toggle the switch to allow sign ins from less secure apps.
    c. Get the inputs from the user. For example, username, password, and the email address of the recipient.
        i. Should we use input() or raw_input() ?
    d. Get output of ping command using *subprocess* module
        i. ping_response = subprocess.Popen(["the name of the command", "argument 1", "argument 2", "ping destination address/name"], stdout=subprocess.PIPE).stdout.read()

e. Try communicating with the gmail's SMTP server
    i. create a server connection
        1. server = smtplib.SMTP_SSL('smtp.gmail.com', 465)
    ii. send HELO message
        1. server.ehlo()
    iii. login using username and password of the dummy gmail account you created
f. Initialize the SUBJECT and TEXT of the mail you want to send
g. Create a message using SUBJECT and TEXT
    i. message = 'Subject: {}\n\n{}'.format(SUBJECT, TEXT)
h. The whole message including the subject and contents should look like this:

```
Subject: Server ping result!

I have pinged google.com and the result is attached to this email:

PING google.com (172.217.164.110): 56 data bytes
64 bytes from 172.217.164.110: icmp_seq=0 ttl=54 time=10.425 ms
64 bytes from 172.217.164.110: icmp_seq=1 ttl=54 time=10.823 ms

--- google.com ping statistics ---
2 packets transmitted, 2 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 10.425/10.624/10.823/0.199 ms
```

i. Send the message
    i. server.sendmail(email_address, receiver_email, message)

4. For Demo:
    a. Send an email to coen146lab5@gmail.com, including the ping results, and your full name as the subject