Python Programming Assignment 1: SMTP

In this assignment you will get hands on experience implementing a standard protocol using Python: Simple Mail Transfer Protocol (SMTP).

You will be developing a simple mail client using socket programming to establish a TCP connection, do an SMTP handshake, send a message, and finally close the connection. Please note that Python has a module called `smptlib` to send mail via SMTP, **but you will not be using this in this lab – using `smptlib` will result in a 0**.

Many email servers do not accept vanilla TCP connections from unknown IP addresses, so your connection may be rejected. Google (not applicable for NYU mail servers) has also implemented additional authentication measures that will likely reject your request. I was able to send an email via SMTP to NYU's mail server, so if you get stuck I would suggest using NYU's mail server, but there is no limitation to which mail server you use.

Code

Skeleton code has been provided below. Your job is to fill in the code blocks between `#Code Start` and `#Code End`. Between each start and end, it may require one or more lines of code to complete this piece.

Additional Notes

Don't forget to check your spam/junk folders!

In order to get the mail server, you should use the `dig` command that we discussed in class.

The sample code uses Python 2.

What to hand in

In your submission, provide the Python file that you used to run your code. Please make sure to provide the `dig` command that you used to find the mail server resource record as a comment in the code. You should also submit a screenshot proving that you received the email message that was sent from your code.

```
from socket import *
import base64

# Choose a mail server (e.g. NYU. mail server) and call it mailserver
# Code Start (enter your dig command as a comment here) # Code End
mailserver = #Code Start   #Code End
serverPort = 25

# Create socket and establish TCP connection with mailserver
# Code Start

# Code End
tcp_resp = clientSocket.recv(1024).decode()
print(tcp_resp)

# Send HELO command to begin SMTP handshake.
heloCommand = 'HELO Alice\r\n'
clientSocket.send(heloCommand.encode())
helo_resp = clientSocket.recv(1024).decode()
print(helo_resp)


# Send MAIL FROM command and print response.
# Code Start

# Code End


# Send RCPT TO command and print server response.
# Code Start

# Code End

# Send DATA command and print server response.
# Code Start

# Code End


# Send email data.
# Code Start

# Code End
# Send message to close email message.
# Code Start

# Code End

# Send QUIT command and get server response.
# Code Start

# Code End

clientSocket.close()
```