**Network Programming for Engineers (ECE 5650)**

**Lab 2**

**Team Members Names: Anika Tasnim, Li Lin**

**Source Code(s):**

**Source code for Client:**

#!/usr/bin/python

# -\*- coding: UTF-8 -\*-

# File Name：TCPEchoAppClient.py

# Created: 9/16/2020

# Author: Li Lin & Anika Tasnim

**from** socket **import** **\***

serverName **=** 'localhost'

serverPort **=** 12000

**with** socket**(**AF\_INET**,** SOCK\_STREAM**)** **as** clientSocket**:**

#Setup connection with server

clientSocket**.**connect**((**serverName**,**serverPort**))**

isContinue **=** **True**

**while** isContinue**:**

#Ask user to input message

message **=input(**'Input lowercase sentence:'**)**

#Send message to server

clientSocket**.**send**(**message**.**encode**())**

#Receive message from server

modifiedMessage **=** clientSocket**.**recv**(**1024**)**

**print(**'From Server the Sentence in Upper Case:'**,** modifiedMessage**.**decode**())**

messageNum **=** clientSocket**.**recv**(**1024**)**

**print(**'From Server number of words in the sentence:'**,** messageNum**.**decode**())**

#User input confirmation

**while** **True:**

Yes\_No **=input(**'Do you want to send more message? Y/N '**)**

**if** **(**Yes\_No **==** 'Y'**)** **or** **(**Yes\_No **==** 'y'**):**

**break**

**elif** **(**Yes\_No **==** 'N'**)** **or** **(**Yes\_No **==** 'n'**):**

isContinue **=** **False**

**break**

**else:**

**print(**'Please type Y or N only！'**)**

**Source code for Server:**

#!/usr/bin/python

# -\*- coding: UTF-8 -\*-

# File Name：TCPEchoAppServer.py

# Created: 9/16/2020

# Author: Li Lin Anika Tasnim

**from** socket **import** **\***

serverPort **=** 12000

**with** socket**(**AF\_INET**,** SOCK\_STREAM**)** **as** serverSocket**:**

#bind with server address and port

serverSocket**.**bind**((**''**,**serverPort**))**

#Start to monitor

serverSocket**.**listen**(**1**)**

**print(**'The server is ready to receive...'**)**

#wait for client's connection

connectionSocket**,**addr **=** serverSocket**.**accept**()**

**with** connectionSocket**:**

**while** **True:**

#Receive message from client

message **=** connectionSocket**.**recv**(**1024**)**

#seperating the words by using split functions

words **=** message**.**split**()**

#counting the number of words

count **=** **len(**words**)**

**print(**'Received:' **+** message**.**decode**()** **+** ' ;Number of Words = ' **+** **str(**count**))**

#Change all letters to Upper case

modifiedMessage **=** message**.**decode**().**upper**()**

#Count the number of words from received message

messageNum **=** **str(**count**)**

#Send uppder case letters to Client

connectionSocket**.**send**(**modifiedMessage**.**encode**())**

#Send word number to Client

connectionSocket**.**send**(**messageNum**.**encode**())**

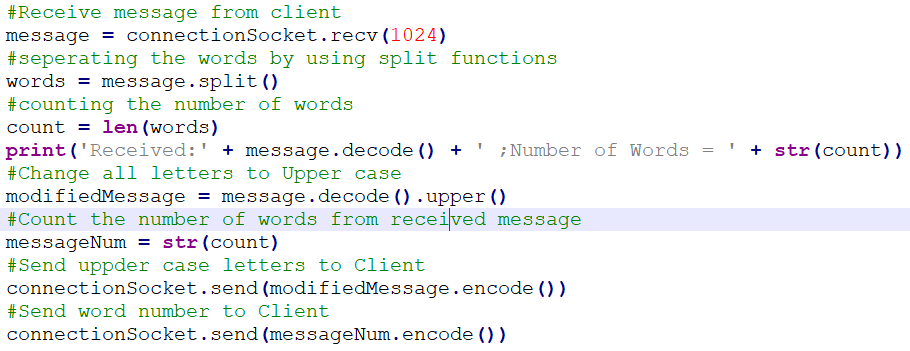
**Testing Procedure, including Description of Inputs**

After writing two codes for the client and server for the TCP Echo Application, we run the program by opening two different terminals (One for the client and one for the server). We have run the server first on the first terminal as a command line and then the client on the other terminal.

The modifications of the codes with screenshots are shown below:

1. The server sends two messages: convert the sentence in upper case and count the number of words in the sentence.

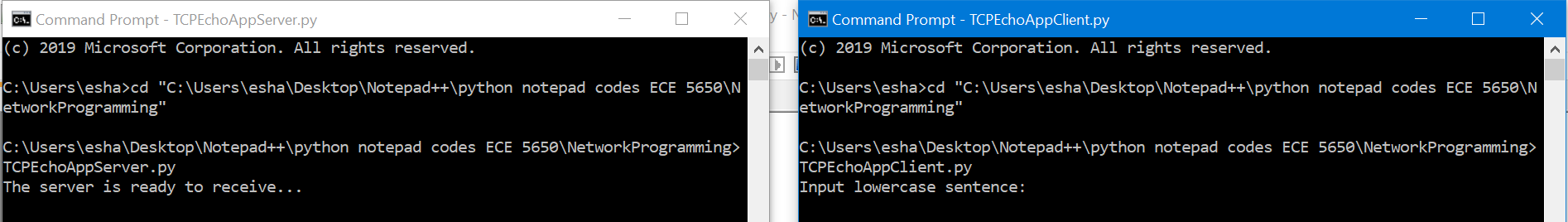
Change in source code: After getting the message from client, in the server code we separate the words in the message using “split function” and count the total number of word using “len function”. After decoding the encoded message from the client end, we convert the message in upper case. Then finally we send the encoded uppercase message with the total number of words to the client.



Testing and debugging the program? (c.)

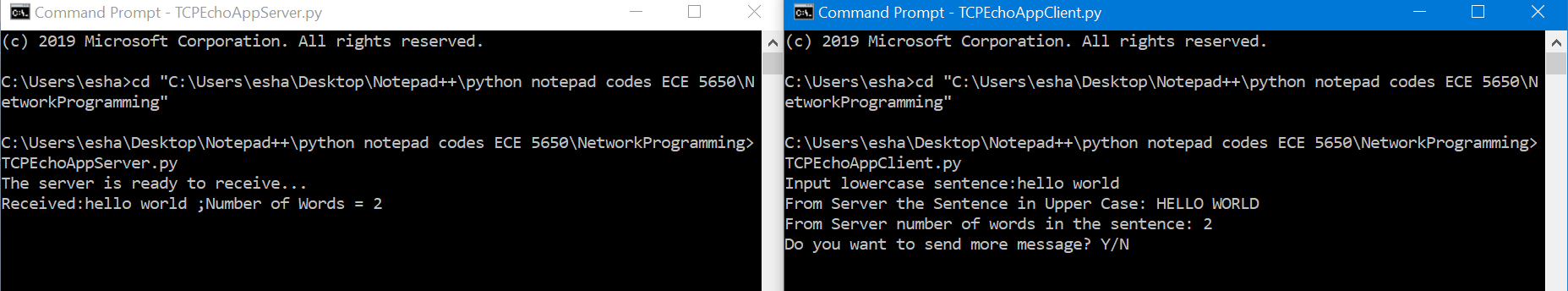
**Screenshots and Their Explanations:**

Opening server and client terminal and preparing for the server to receive:

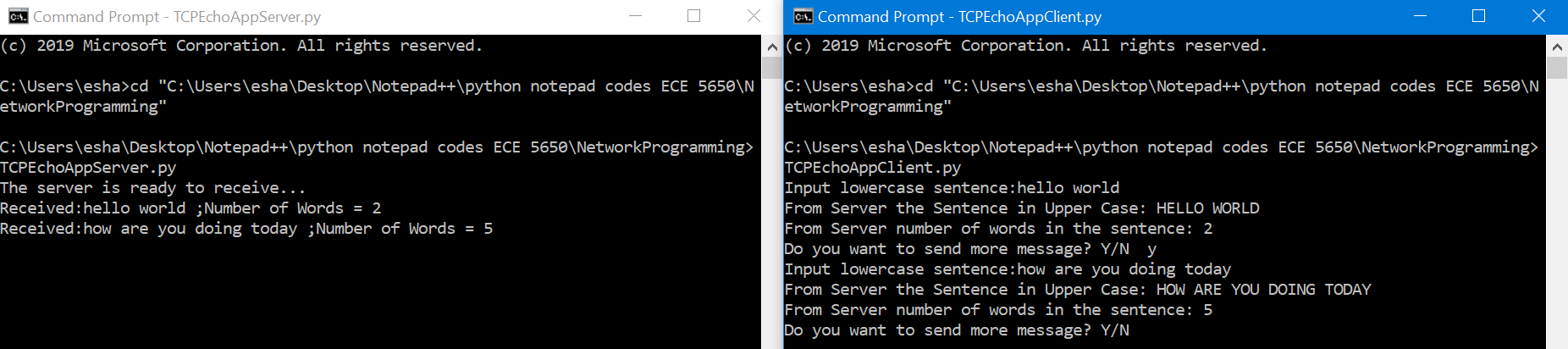


Entering a message in the client window:

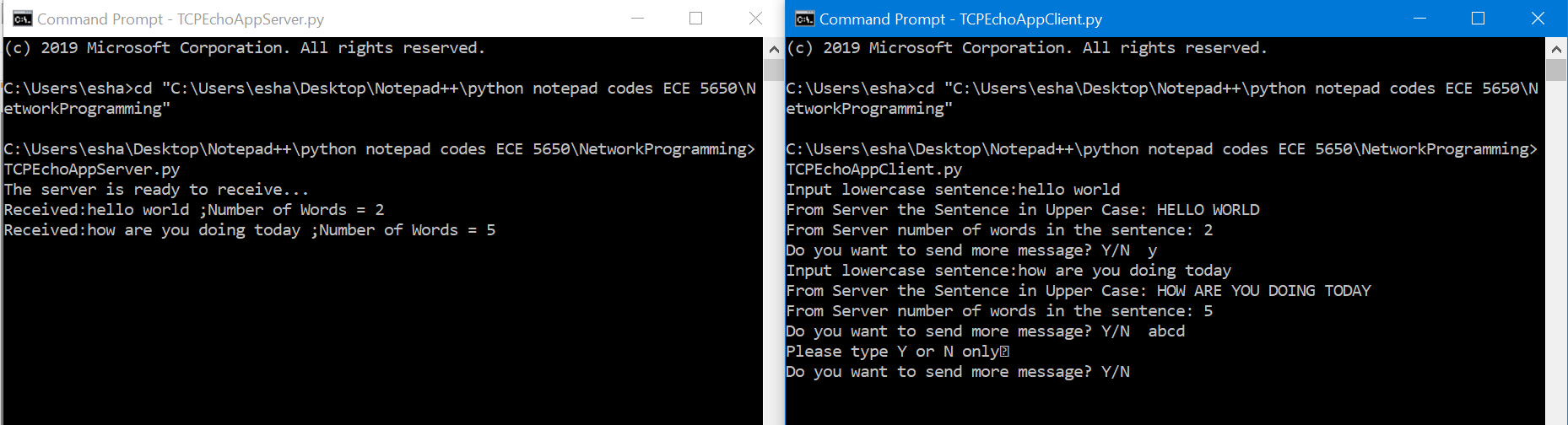
After entering the message in the client window, the message is received by the server. The message is converted to upper case, and the number of words are counted in the server program and sent to client as shown below.



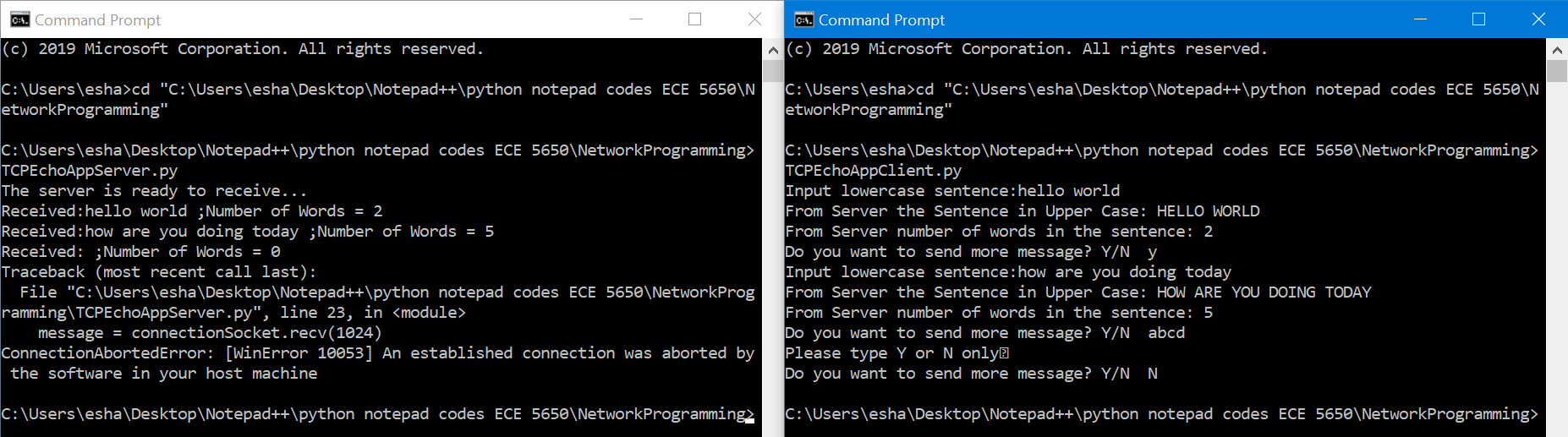
Next in the client window, it is asking if we want to send more messages. If we answer Y/y the program continues to run.



If we enter anything other than Y/y/N/n then it displays a message asking to type the right letters:



The program only terminates when we enter N/n then and we have to connect from the server again



**Completion Status and Self-Critique:**

For program **Client**:

* The program meet all requirements
* The program runs correctly all the time
* We have adequately tested the program
* The program is well documented

For program **Server**:

* The program meet all requirements
* The program runs correctly all the time
* We have adequately tested the program
* The program is well documented