|  |
| --- |
| TCPserver.py |
| from socket import \*   #takes in the string that was sent to us by the client and steps through #counting the characters on each line that are separated by a space def countBetweenSpace(message):  i = 0  result = ''  for ch in message:  if (chr(ch) == ' ') or (chr(ch) == '\n'):  result += str(i) + ' '  i = 0  if chr(ch) == '\n':  result += '\n'  else:  i += 1  if i > 0:  result += str(i)  return result   server\_port = 12000 #port number server\_socket = socket(AF\_INET, SOCK\_STREAM) #use SOCK\_STREAM for TCP server\_socket.bind(('', server\_port)) #define the port being used for  #comunication server\_socket.listen(1) #open port print('server is listening and ready to receive') while True:  connection\_socket, addr = server\_socket.accept() #connect to client   sentence = connection\_socket.recv(1024) #receive data  result\_message = countBetweenSpace(sentence) #perform manipulation of data  connection\_socket.send(result\_message.encode()) #return the result to the client   connection\_socket.close() #end communication with client |

|  |
| --- |
| TCPclient.py |
| from socket import \*   server\_name = 'localhost' #server ip server\_port = 12000 #server port number client\_socket = socket(AF\_INET, SOCK\_STREAM) #UDP connection using STREAM client\_socket.connect((server\_name, server\_port)) #connect to the server sentence = open('testcases.txt', 'r').read() #get data from local file client\_socket.send(sentence.encode()) #send data to server modified\_sentence = client\_socket.recv(1024) #wait for server reply print(modified\_sentence.decode()) #print result client\_socket.close() #end connection |

|  |
| --- |
| UDPserver.py |
| from socket import \*   #takes in the string that was sent to us by the client and steps through #counting the characters on each line that are separated by a space def countBetweenSpace(message):  i = 0  result = ''  for ch in message:  if (chr(ch) == ' ') or (chr(ch) == '\n'):  result += str(i) + ' '  i = 0  if chr(ch) == '\n':  result += '\n'  else:  i += 1  if i > 0:  result += str(i)  return result   server\_port = 12000 #port number server\_socket = socket(AF\_INET, SOCK\_DGRAM) #use SOCK\_DGRAM for UDP server\_socket.bind(("", server\_port)) #define the port being used  #for communication print('UDP server is ready') while True:  message, client\_Addr = server\_socket.recvfrom(2048) #receive data from a client  result\_message = countBetweenSpace(message) #perform manipulation of data  server\_socket.sendto(result\_message.encode(), client\_Addr) #return the result to the  #client |

|  |
| --- |
| UDPclient.py |
| from socket import \*   server = 'localhost' #server ip server\_port = 12000 #server port number client\_socket = socket(AF\_INET, SOCK\_DGRAM) #UDP connection using DGRAM message = open('testcases.txt', 'r').read() #get data from local file client\_socket.sendto(message.encode(), (server, server\_port)) #send data to server result, server\_Addr = client\_socket.recvfrom(2048) #wait for server reply print(result.decode()) #print result client\_socket.close() #end connection |