*"""  
Henry Ang  
2/7/17  
CSC 4800  
Chapter 2: TCP Server/Client  
TCP Client: recognize the following commands:  
  
EXITSERVER - sends "EXITSERVER" message to server. Close client socket and terminate.  
date - sends "date" message to server. Prints timestamp.  
os - sends "os" message to server. Prints OS-information.  
ls (serverpath) - sends ls (serverpath) message to server. Prints current or specified directory  
sleep (secs) - sends sleep (secs) message to server. Prints number of seconds slept.  
"""***from** socket **import** \*  
**import** sys, re  
  
HOST = **'127.0.0.1'**PORT = 21567  
BUFSIZ = 1024  
ADDR = (HOST, PORT)  
  
*# server recognized messages*EXIT\_MESSAGE = **"EXITSERVER"**DATE\_MESSAGE = **"date"**OS\_MESSAGE = **"os"**PATH\_MESSAGE = **"ls"**SLEEP\_MESSAGE = **"sleep"**tcpCliSock = socket(AF\_INET, SOCK\_STREAM)  
tcpCliSock.connect(ADDR)  
  
**while True**:  
  
 data = input(**'> '**)  
 **if not** data: *# check if no data inputted* **break** validSleepSearch = re.search(**"^(sleep)(.\*?)\*$"**, data) *# search for message that starts with sleep* sleepMatch = re.match(**"^(sleep)[ ][\d+]+$"**, data) *# match valid sleep input* validPathSearch = re.search(**"^(ls)(.\*?)\*$"**, data) *# search for message that starts with ls* lsSearch = re.match(**"^(ls)[ ]([\d+\w+/.-\_ ]+)$"**, data) *# match valid directory* **if** data == EXIT\_MESSAGE: *# if input messsage is "EXITSERVER"* tcpCliSock.send(bytes(data, **'utf-8'**)) *# send input to server* tcpCliSock.close() *# close socket* sys.exit() *# terimate client* **elif** data == DATE\_MESSAGE: *# if input messsage is "date"* tcpCliSock.send(bytes(data, **'utf-8'**))  
 recieved = tcpCliSock.recv(BUFSIZ)  
 print(**"Date: "** + recieved.decode(**'utf-8'**))  
  
 **elif** data == OS\_MESSAGE: *# if input message is "os"* tcpCliSock.send(bytes(data, **'utf-8'**))  
 recieved = tcpCliSock.recv(BUFSIZ)  
 print(**"OS: "** + recieved.decode(**'utf-8'**))  
  
 **elif** data == PATH\_MESSAGE: *# if input message is "ls"* tcpCliSock.send(bytes(data, **'utf-8'**))  
 recieved = tcpCliSock.recv(BUFSIZ)  
 print(**"ls \"path\": "** + recieved.decode(**'utf-8'**))  
  
 **elif** (validPathSearch != **None**): *# check if input starts with "ls"* **if** (lsSearch != **None**): *# check if input is a valid directory* tcpCliSock.send(bytes(data, **'utf-8'**))  
 recieved = tcpCliSock.recv(BUFSIZ)  
 print(**"ls \"path\": "** + (recieved.decode(**'utf-8'**)))  
 **else**: *# invalid directory* print(**"No such directory"**)  
  
 **elif** (validSleepSearch != **None**): *# check if input starts with "sleep"* **if** sleepMatch != **None**: *# check if input is valid sleep number* tcpCliSock.send(bytes(data, **'utf-8'**))  
 recieved = tcpCliSock.recv(BUFSIZ)  
 print(**"Slept for "** + recieved.decode(**'utf-8'**) + **" seconds"**)  
 **else**: *# default sleep (5 sec)* tcpCliSock.send(bytes(SLEEP\_MESSAGE, **'utf-8'**))  
 recieved = tcpCliSock.recv(BUFSIZ)  
 print(**"Slept for "** + recieved.decode(**'utf-8'**) + **" seconds"**)  
  
 **else**: *# default behavior* tcpCliSock.send(bytes(data, **'utf-8'**))  
 data = tcpCliSock.recv(BUFSIZ)  
 **if not** data:  
 **break** print(data.decode(**'utf-8'**))  
  
tcpCliSock.close()