

Name: _____

Directions: MAKE SURE TO COPY YOUR ANSWERS TO A SEPARATE SHEET FOR SENDING ME AN ELECTRONIC COPY LATER.

1. (100) Below is a variant of the client-server example in Section 5.1.1.1, again using Python's **thread** module. Here, the number of clients increases and decreases over time, starting with none. When we again reach a situation with no clients, the server prints **v** and exits.

Fill in the blanks.

```
# multiple clients connect to server; clients come and go,
# starting with none, but the server quits later when
# wthere are no clients left; each client repeatedly sends a
# letter k, which the server adds to a global string v and
# echos back to the client; k = '' means # the client is
# dropping out; when all clients are gone, server prints
# final value of v

# this is the server

import socket
import sys

import blank (a)

class gb:
    blank (b)
    v = ''
    threadlist = [] # client sockets
    firstclntyet = False
    port = int(sys.argv[1])

def serveclient(sock):
    if blank (c) :
        while True:
            k = sock.recv(1)
            if k == '': break
            gb.vlock.acquire()
            gb.v += k
            gb.vlock.release()
            blank (d)
            sock.close()
            blank (e)
    else:
        lstn = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
        lstn.bind(('', gb.port))
        lstn.listen(5)
        while True:
            (clnt, ap) = lstn.accept()
            blank (f)
            gb.firstclntyet = True
            thread.start_new_thread(serveclient, (clnt,))
        lstn.close()

def main():
    thread.start_new_thread(serveclient, (None,))
    while blank (g) : pass
    while blank (h) : pass
    print 'the final value of v is ', gb.v

if __name__ == '__main__': main()
```

Solutions:

1.

```
# multiple clients connect to server;
# clients come and go, but the # server quits
# when there are no clients; each client
# repeatedly sends a letter k, which the
# server adds to a global string v and
```

```

# echos back to the client; k = '' means
# the client is dropping out; when all clients
# are gone, server prints final value of v

# this is the server

import socket
import sys

import thread

class gb:
    vlock = thread.allocate_lock()
    v = ''
    threadlist = []
    firstclntyet = False
    port = int(sys.argv[1])

def serveclient(sock):
    if sock:
        while True:
            k = sock.recv(1)
            if k == '': break
            gb.vlock.acquire()
            gb.v += k
            gb.vlock.release()
            sock.send(gb.v)
        sock.close()
        gb.threadlist.remove(sock)
    else:
        lstn = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
        lstn.bind(('', gb.port))
        lstn.listen(5)
        while True:
            (clnt, ap) = lstn.accept()
            gb.threadlist.append(clnt)
            gb.firstclntyet = True
            thread.start_new_thread(serveclient, (clnt,))
        lstn.close()

def main():
    thread.start_new_thread(serveclient, (None,))
    while not gb.firstclntyet: pass
    while gb.threadlist: pass
    print 'the final value of v is ', gb.v

if __name__ == '__main__': main()

```