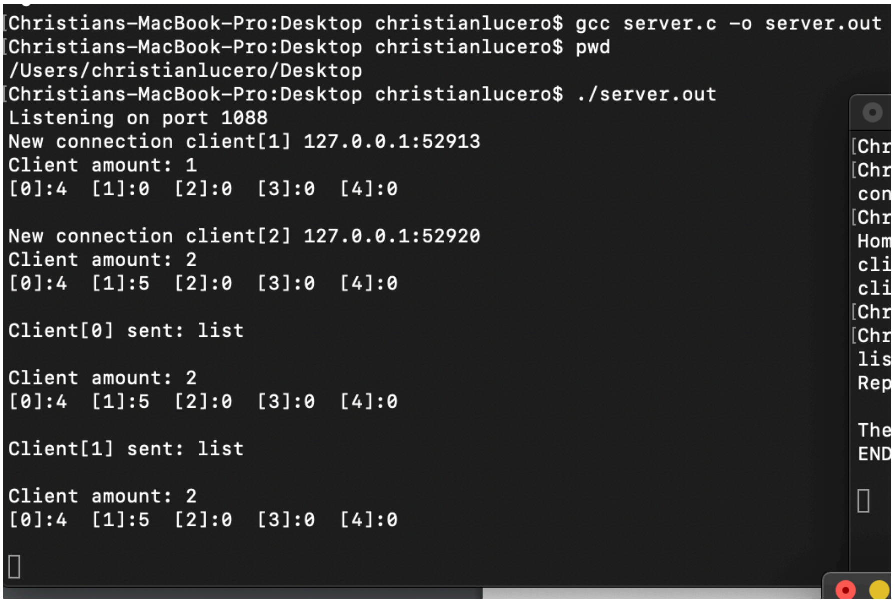
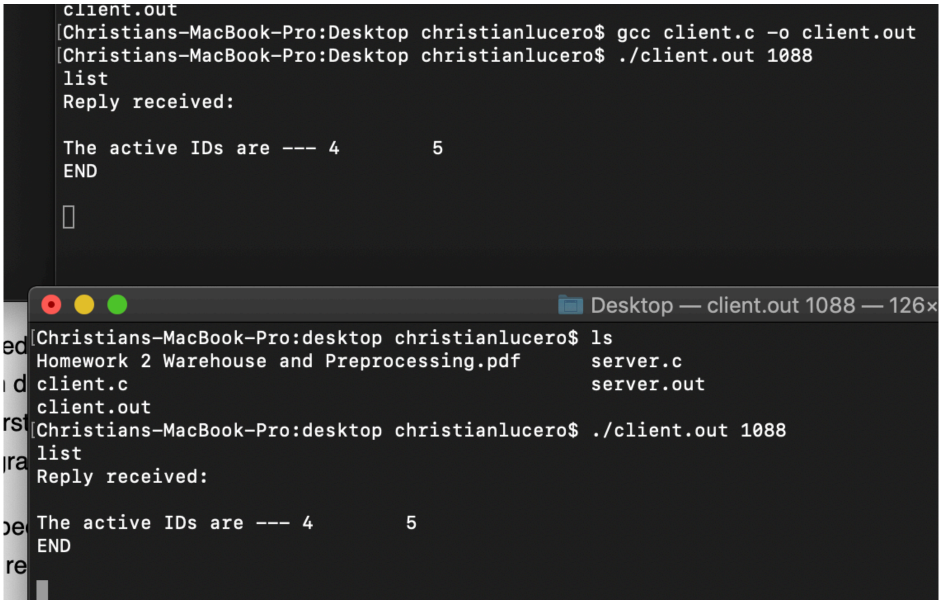
Lab 1 Peer Review

Both Server.c and Client.c successfully compile and run. The first test I performed was to verify the list command was whether or not multiple clients can connect to the server. It does indeedaccept multiple client connections. The next test I performed was to see whether or not the list command functions. It does and returned the id number of both connected clients, see screenshots below.



```
Christians-MacBook-Pro:Desktop christianlucero$ gcc server.c -o server.out
Christians-MacBook-Pro:Desktop christianlucero$ pwd
/Users/christianlucero/Desktop
Christians-MacBook-Pro:Desktop christianlucero$ ./server.out
Listening on port 1088
New connection client[1] 127.0.0.1:52913
Client amount: 1
[0]:4  [1]:0  [2]:0  [3]:0  [4]:0

New connection client[2] 127.0.0.1:52920
Client amount: 2
[0]:4  [1]:5  [2]:0  [3]:0  [4]:0

Client[0] sent: list

Client amount: 2
[0]:4  [1]:5  [2]:0  [3]:0  [4]:0

Client[1] sent: list

Client amount: 2
[0]:4  [1]:5  [2]:0  [3]:0  [4]:0

```

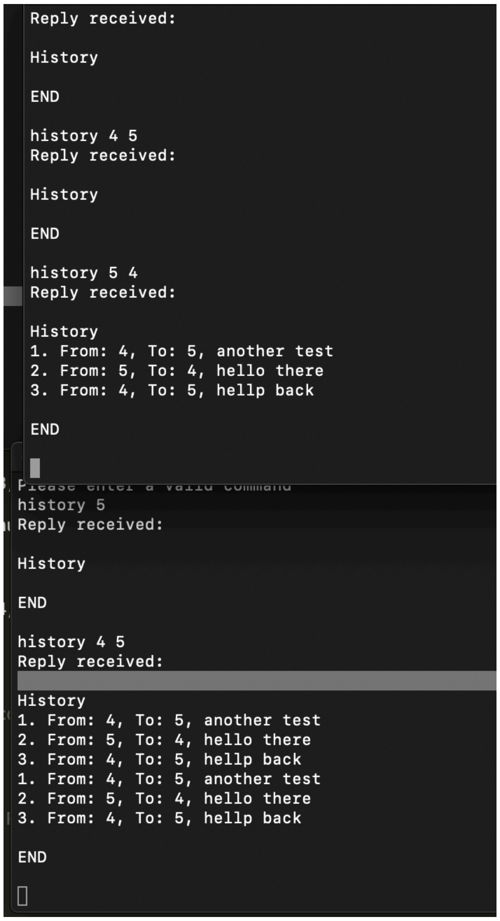
```
Chr
Chr
con
Chr
Hom
cli
cli
Chr
Chr
lis
Rep

The
END
```



```
client.out
Christians-MacBook-Pro:Desktop christianlucero$ gcc client.c -o client.out
Christians-MacBook-Pro:Desktop christianlucero$ ./client.out 1088
list
Reply received:

The active IDs are --- 4        5
END

```

```
Christians-MacBook-Pro:desktop christianlucero$ ls
Homework 2 Warehouse and Preprocessing.pdf      server.c
client.c                                        server.out
client.out
Christians-MacBook-Pro:desktop christianlucero$ ./client.out 1088
list
Reply received:

The active IDs are --- 4        5
END
```

After testing multiple connections and the list command I checked whether or messages worked and you could retrieve the chat history. Both work just fine, message is sent and received between the two clients and recalling the chat history works great too.



```
Reply received:

History

END

history 4 5
Reply received:

History

END

history 5 4
Reply received:

History
1. From: 4, To: 5, another test
2. From: 5, To: 4, hello there
3. From: 4, To: 5, hellp back

END

```

```
Please enter a valid command
history 5
Reply received:

History

END

history 4 5
Reply received:

History
1. From: 4, To: 5, another test
2. From: 5, To: 4, hello there
3. From: 4, To: 5, hellp back
1. From: 4, To: 5, another test
2. From: 5, To: 4, hello there
3. From: 4, To: 5, hellp back

END

```

The last test to perform was checking if the exit command works and it is the only one that does not function properly. Both clients running will get stuck in an infinite loop trying to exit and the only way to stop it is to force quit the program.

The last thing to consider is the style of the code. Looking at the server.c file, the program flows very nicely. The main commands and assignment itself is broken down into several different functions that all sit above the main. The program is properly commented, with a function descriptor above each one detailing what it does as well as comments within the function itself describing smaller bits of code. The only criticism i can make here is that the comments are not very specific or detailed. They are general statements that don't completely describe what is going on and requires the reader to either look closer or not be able to ascertain what the logic is. Other than that the style of the code is easy to understand.

Suggest Grade: A