**Name : Pritesh Ratnappagol**

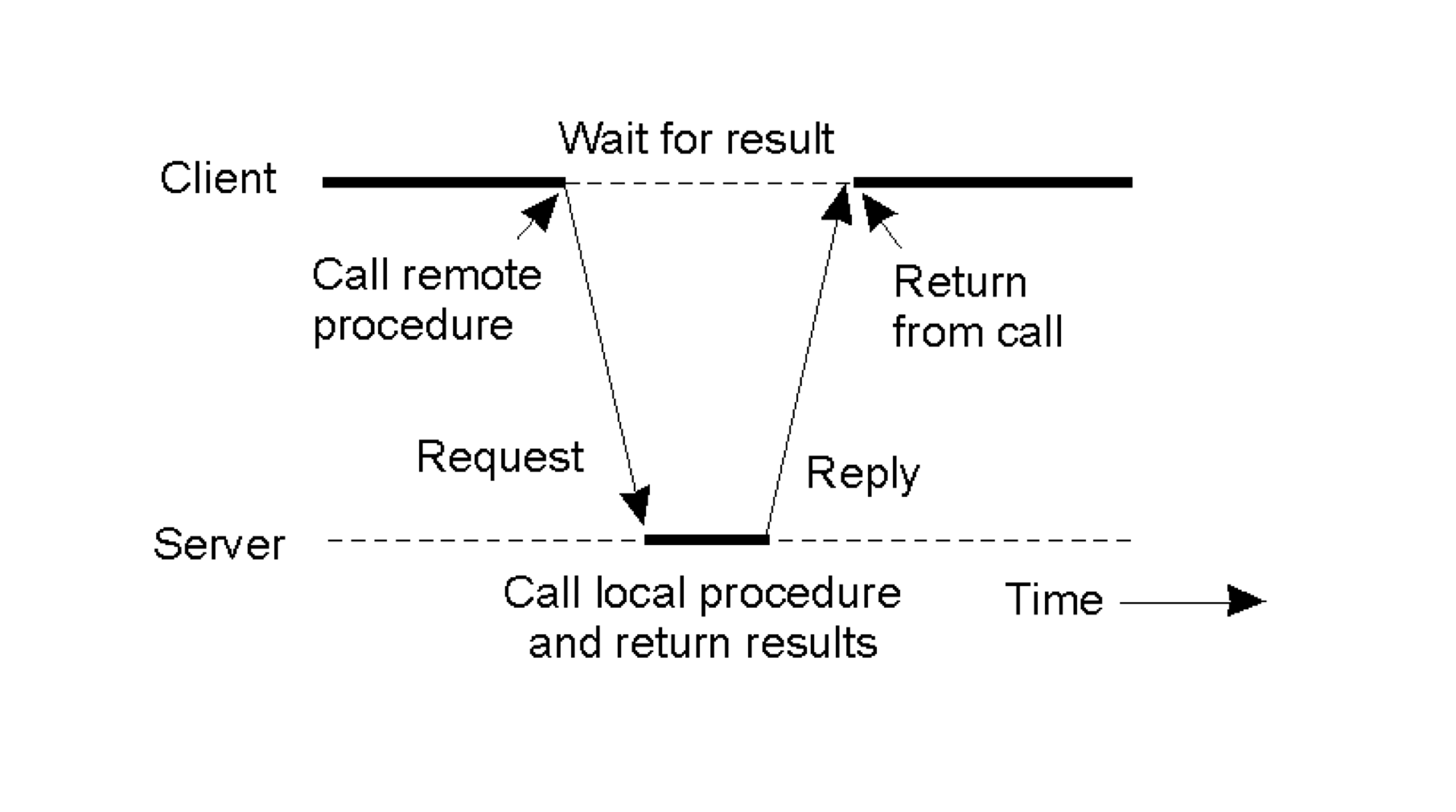
**Aim:**

To understand and implement the RPC system in C language.

**Remote Procedure Call** (**RPC**):

* RPC is a protocol that allows programs to call procedures located on other machines.
* RPC uses the client/server model. The requesting program is a client and the service-providing program is the server.
* The client stub acts as a proxy for the remote procedure.
* The server stub acts as a correspondent to the client stub.
* RPCRuntime is responsible for retransmission, acknowledgements, packets routing and encryption from client to server and vice versa.

**Client and Server Stubs**



**Steps for RPC :**

1. Client procedure calls client stub in normal way
2. Client stub builds message, calls local OS
3. Client's OS sends message to remote OS through RPC Runtime
4. Remote OS gives message to server stub
5. Server stub unpacks parameters, calls server
6. Server does work, returns result to the stub
7. Server stub packs it in message, calls local OS
8. Server's OS sends message to client's OS through RPC Runtime
9. Client's OS gives message to client stub
10. Stub unpacks result, returns to client

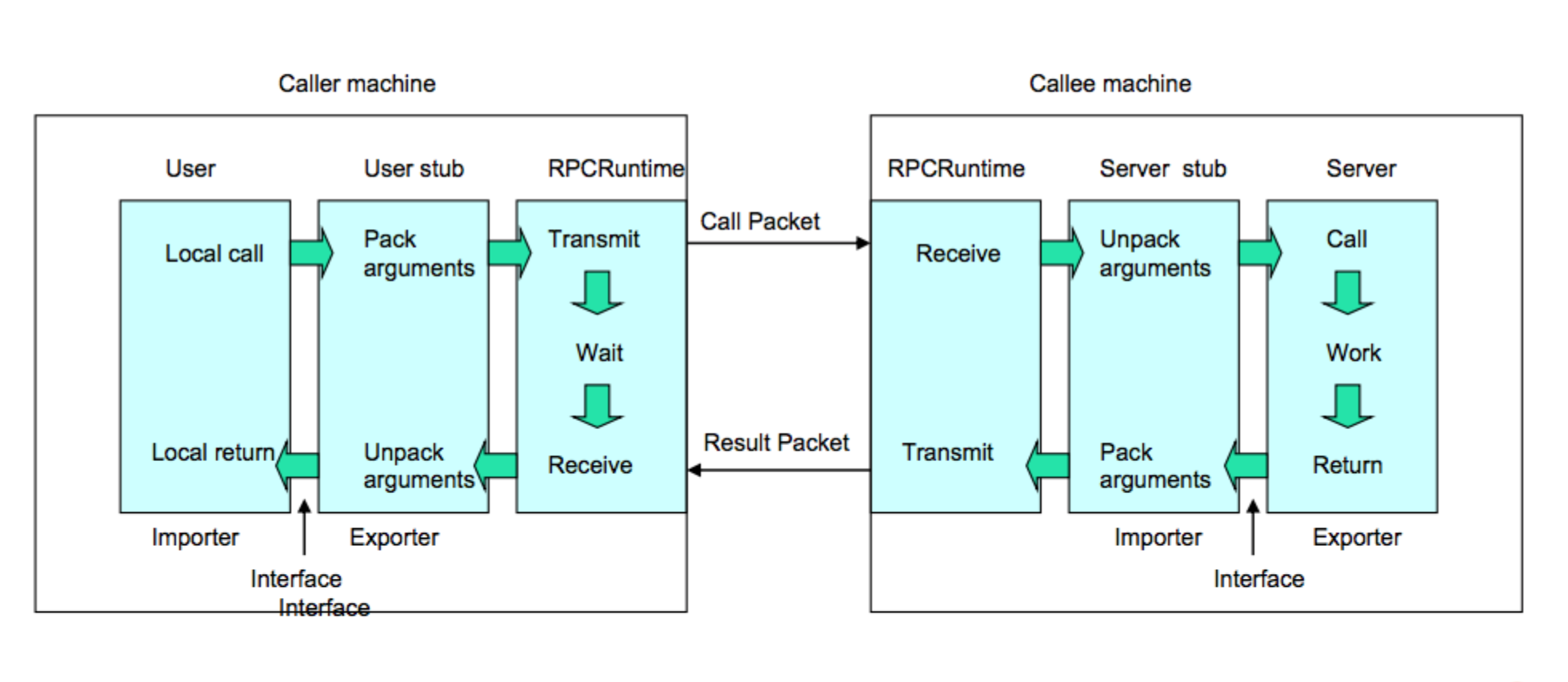


Fig. RPC structure

**Implementation:**

I have used the above concepts for designing and implementing functions in the assignment were client calls the server to perform following actions:

1. Path (pwd command)

2. Echo

3. File Check

4. Sort

5. Matrix Multiplication

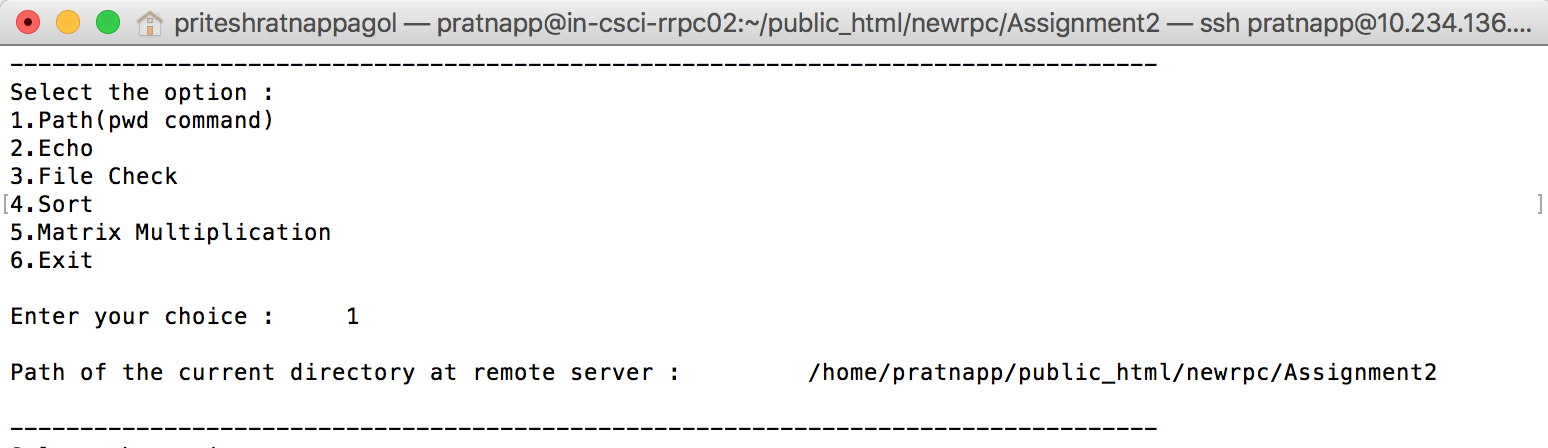
6. Exit

1. Path:

In this option, client calls the server to know the path of the current directory the server is working in, when server gets this call from the client it returns the path of the current directory it is working in.

At the server side to know the current working directory I have used chdir() function.

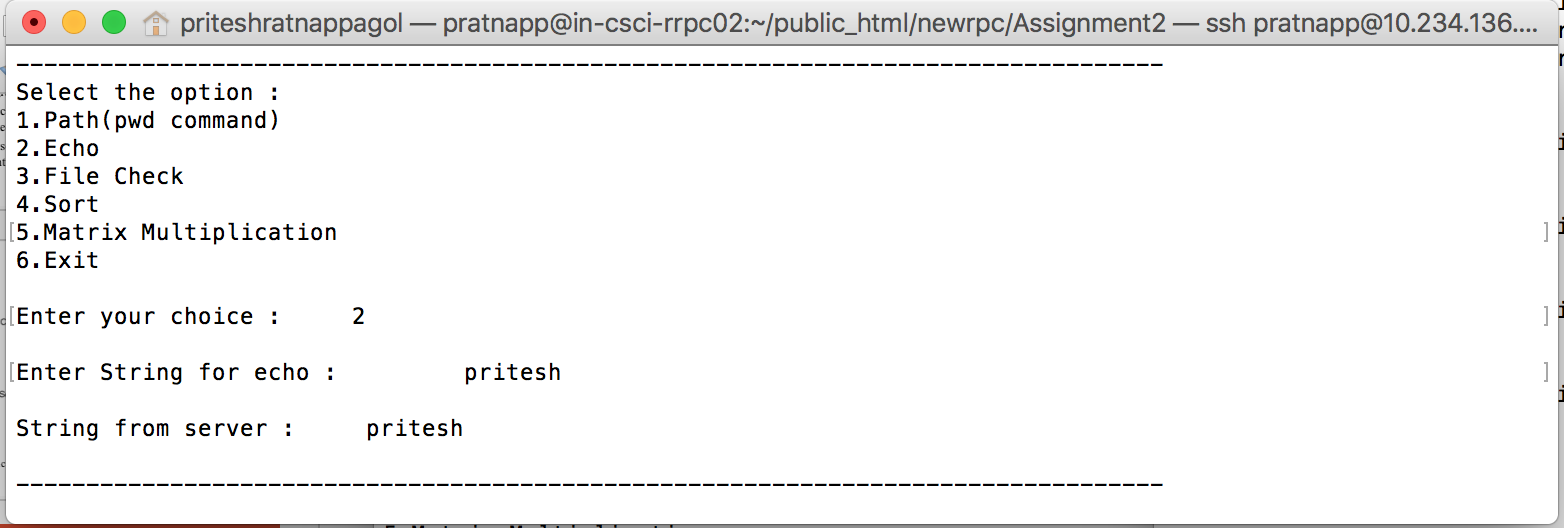
At the client I am just displaying the result of the server.



Screenshot 1: Path (pwd command)

1. Echo

In this option client sends a string to the server and server returns the same string to the client.



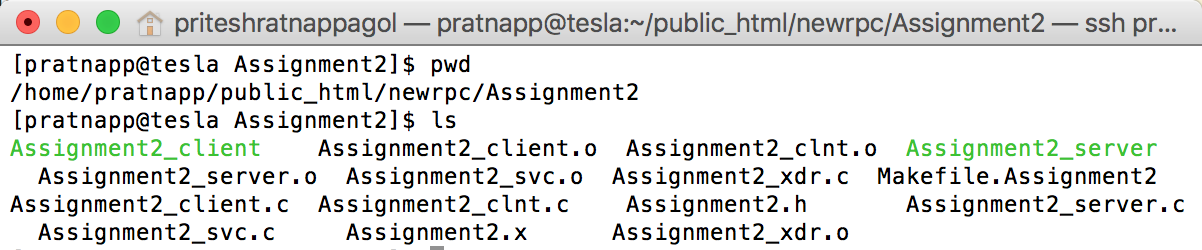
Screenshot 2: Echo

1. File Exists

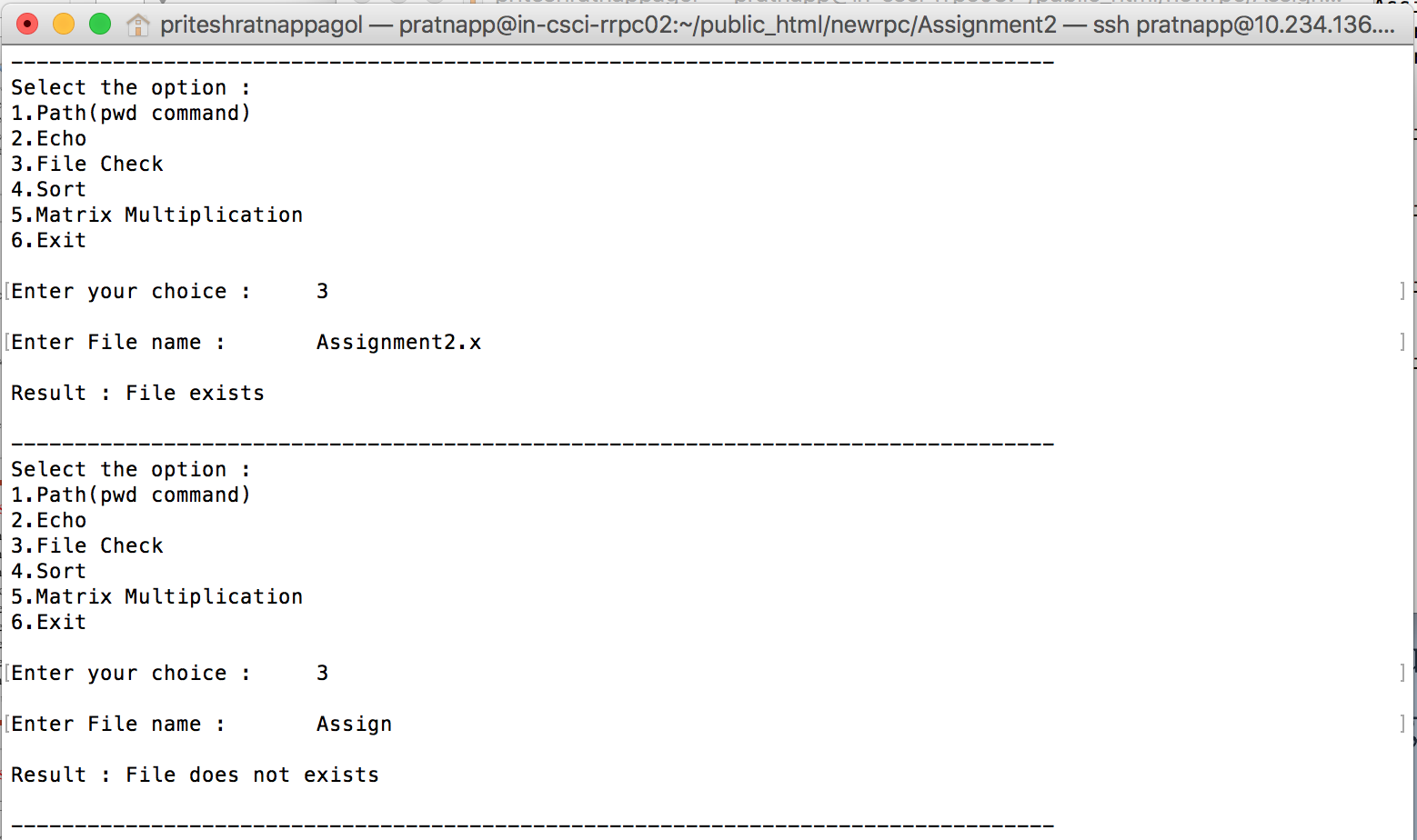
In this option, client calls the server to know if a file exists at the server’s directory by passing the file name, then server checks is file exits or not and returns the answer to the client.

At client, user passes the filename.

At server, I am taking the filename passed by client and checking if I can open the file or not. If I can open the file with the same filename passed by the client then I am replying the client with message “File exists” else “File does not exists”.



Screenshot 3. Files present in the directory



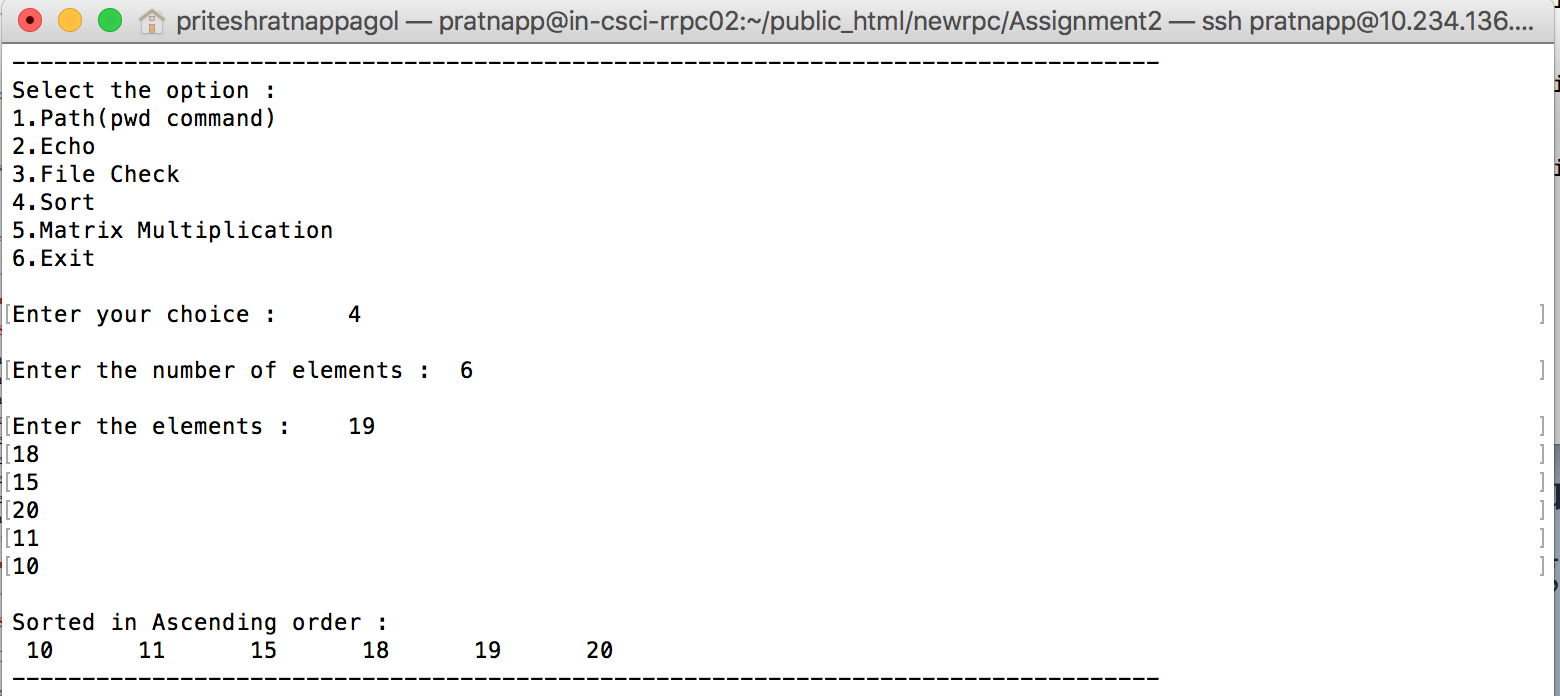
Screenshot 4: Files Exists ; the first part shows the file exists and the second part shows the file does not exists

4) Sort

In this option, at client side user gives the input as the number of elements and the elements he wants to sort in ascending order and sends it to the server where performs the sorting operation sends the results to the client.

At server, I have used bubble sort concept for sorting of numbers.

At client, Displays the sorted elements received from the server.

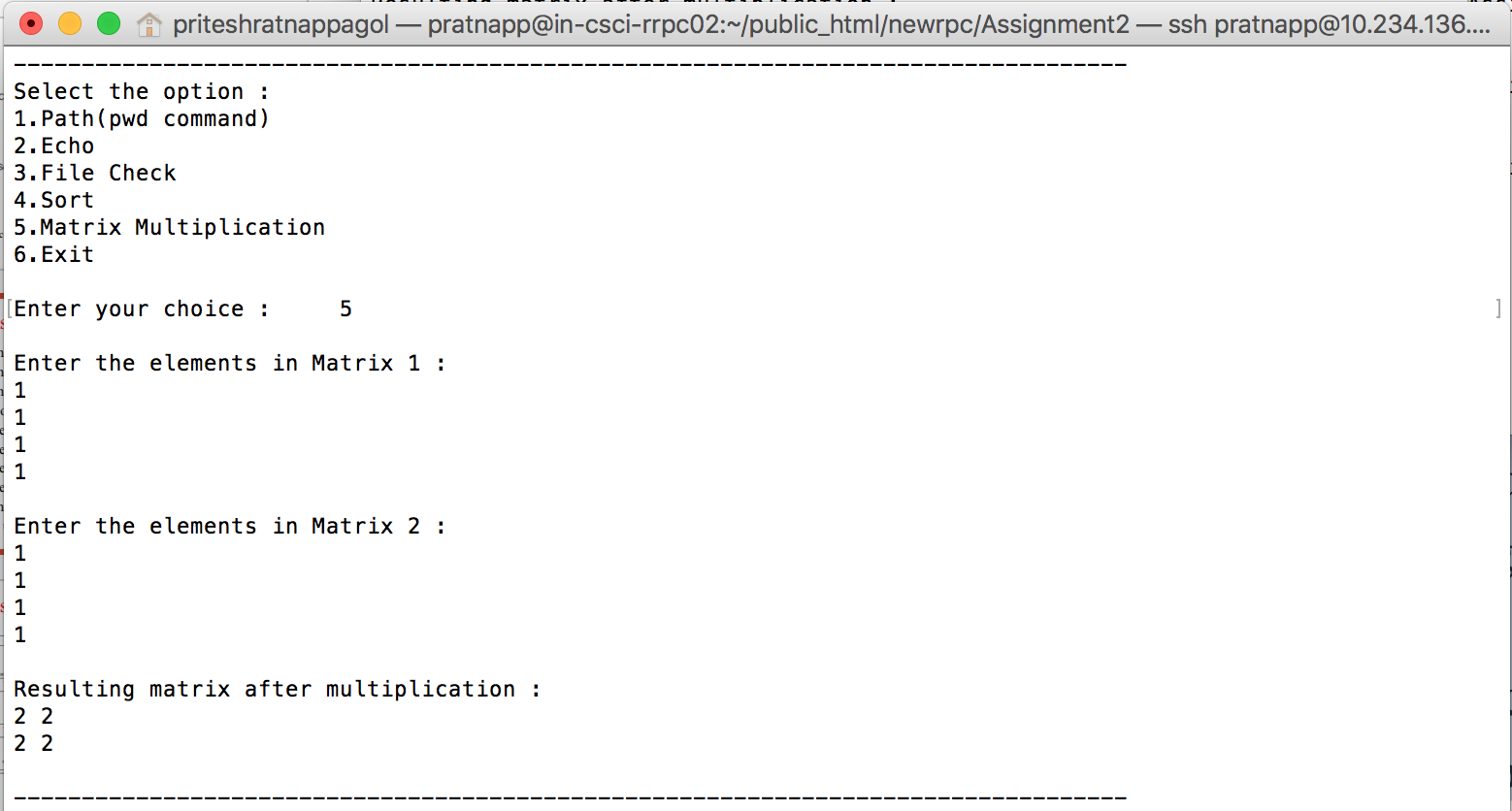


Screenshot 5: Sort

5)Matrix Multiplication

In this option, I am performing the matrix multiplication at the server side. The matrixes for multiplication at the server side are provided at the client side by the user.

At client, I am taking the values of the matrixes and storing it in 1-D array and sending it to the server and after that at the server side I am converting it into the 2-D array(i.e. of matrix form) and then performing the matrix multiplication, then I am storing the resulting values in 1-D array send it to the client side and converting it into a 2-D array (i.e. of matrix form) and displaying the results.



Screenshot 6 : Matrix Multiplication

6) Exit

Exit option is used to stop the client.



Screenshot 7 : Exit

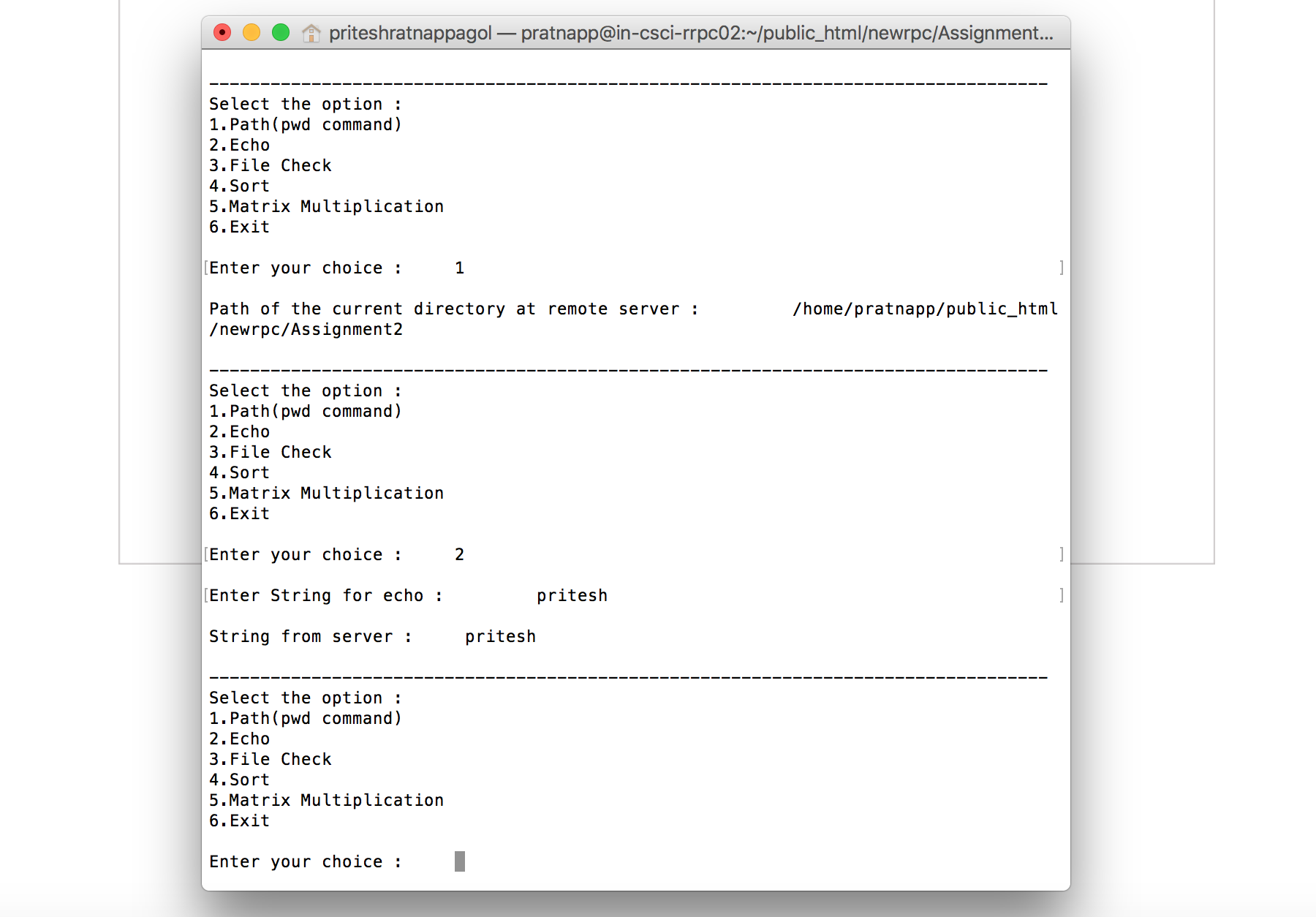
7) Invalid response from the client

If user at the client enters any other number apart from the choices given the program displays the message “Invalid choice. . Please enter again”.



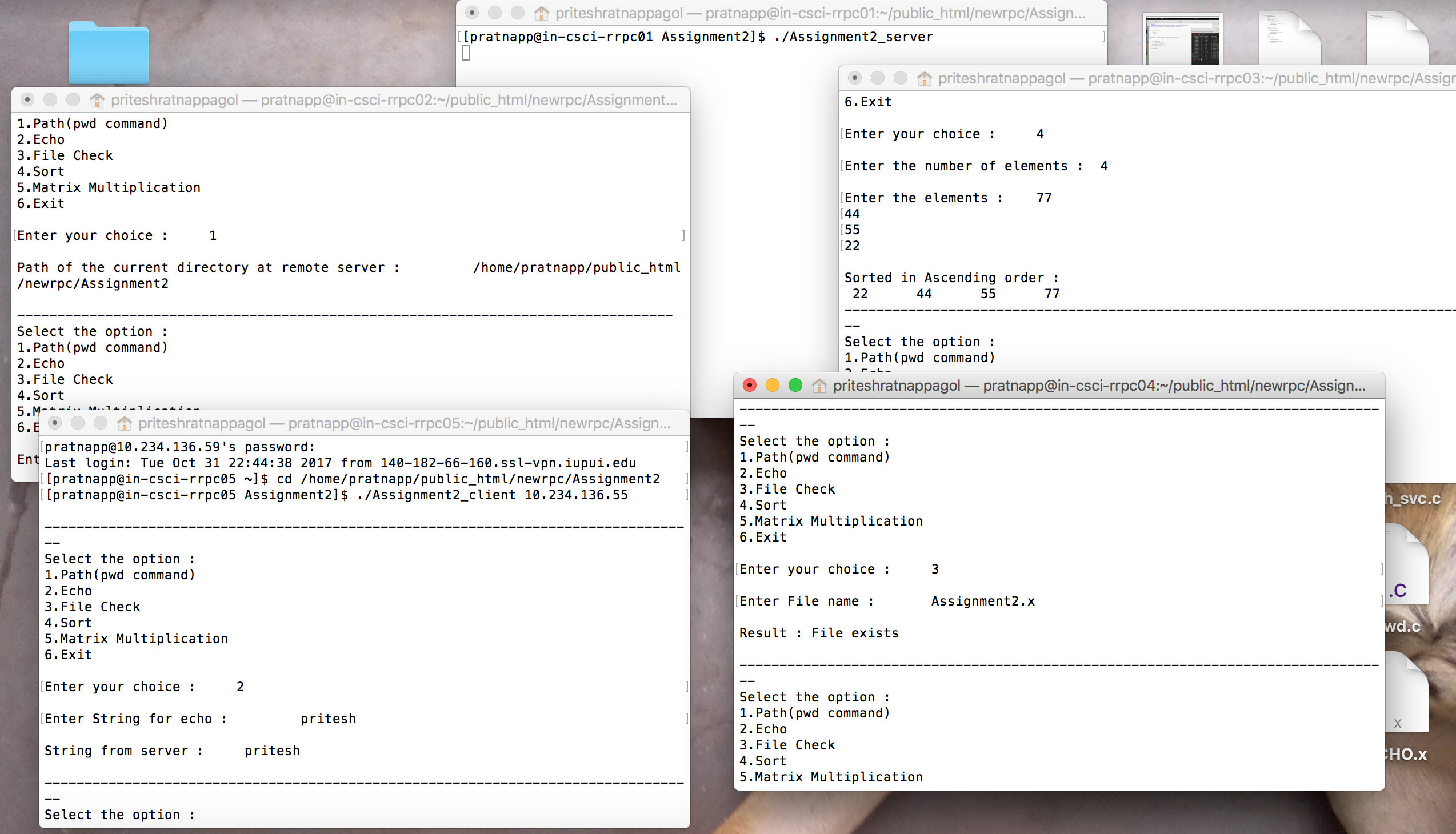
Screenshot 8 : Invalid choice

8) I have used switch case for performing all the function given such that user can make his own choice regarding the execution of the procedures he wants to perform at the server side. And have used while loop to keep accepting the client request so that server must not exit on satisfying one request from the client.



Screenshot 9 : Switch and while loop

9) Multithreading; The implemented program supports multithreading; any number of clients can connect to the server and perform their required operations at the server side.



Screenshot 9 : Multithreading using 4 clients and 1 server