```
<center>
    <h1>
        Computer Network Lab
        <br>
        CSE-325
    </h1>
    <h3>
        Assignment - 5
    </h3>
</center>
```

To implement Remote Command Execution(RCE)

I implemented a program where a client connects to a server and execute commands on the server. I take a command from the client and send it on the server. On the server, I use python's `subprocess` module to run the command and store the output in a pipe. This is equivalent to C's `popen` function. I then take the output from the pipe and send it back to the client.

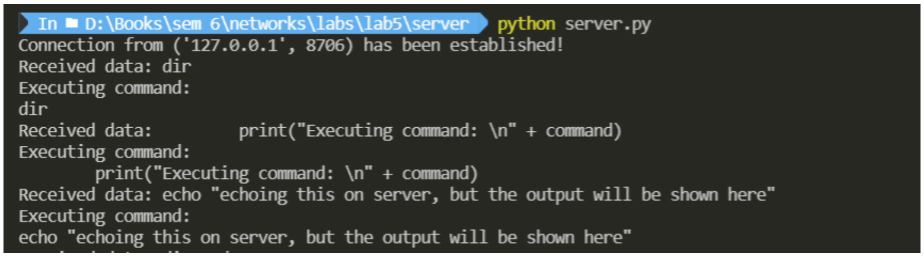
RCE Server

```
import socket
import subprocess

HOST = "127.0.0.1"
PORT = 4204

class Server:
    def __init__(self) -> None:
        self.socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
        self.socket.bind((HOST, PORT))
        self.socket.listen()

    def accept(self):
        """
        This function accpets a client connection and then
        if it recieves any data, it sends it to `execute`
        function and sends back the returned value back
        to the client.
        """
        conn, addr = self.socket.accept()
        print(f"Connection from {addr} has been established!")
        with conn:
            while True:
                data = conn.recv(1024)
                if not data:
                    break
                print(f"Received data: {data.decode('utf-8')}")
                conn.sendall(self.execute(data.decode('utf-8')))

    def execute(self, command):
        """
        This function takes a command as input and executes it,
        it then returns the output of the command.
        """
        print("Executing command: \n" + command)
        x = subprocess.run(command, shell=True, stdout=subprocess.PIPE)
        if x.returncode == 0 and len(x.stdout) > 0:
            return x.stdout
        return "".encode('utf-8')

if __name__ == "__main__":
    server = Server()
    server.accept()
```

Chat Client

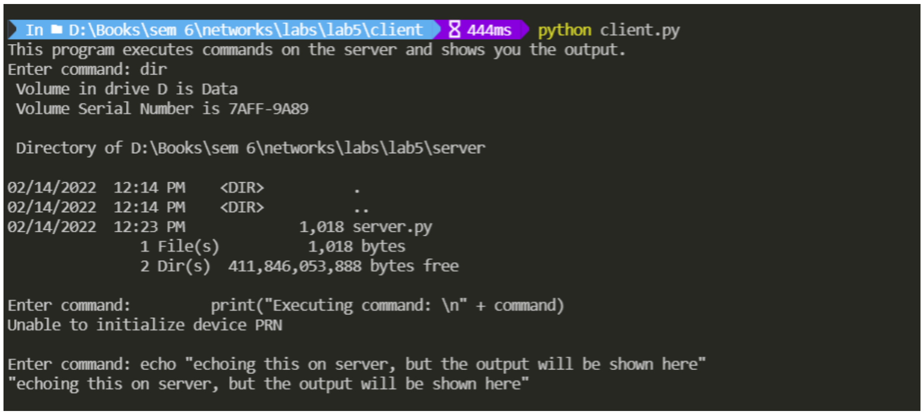
```
import socket
import subprocess

HOST = "127.0.0.1"
PORT = 4204

class Client:
    def __init__(self) -> None:
        self.socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
        self.socket.connect((HOST, PORT))
        print("This program executes commands on the server and shows you the output.")

    def start(self):
        """
        This function takes input from the user and sends it to the server.
        The server then executes the command and returns the output.
        This function then displays that output.
        """
        while True:
            command = input("Enter command: ")
            self.socket.sendall(command.encode('utf-8'))
            data = self.socket.recv(1024)
            print(data.decode('utf-8'))

if __name__ == "__main__":
    server = Client()
    server.start()
```

Output



```
In D:\Books\sem 6\networks\labs\lab5\server  python server.py
Connection from ('127.0.0.1', 8706) has been established!
Received data: dir
Executing command:
dir
Received data:          print("Executing command: \n" + command)
Executing command:
        print("Executing command: \n" + command)
Received data: echo "echoing this on server, but the output will be shown here"
Executing command:
echo "echoing this on server, but the output will be shown here"
```

Fig: RCE Server. It takes in command, executes and sends output to server



```
In D:\Books\sem 6\networks\labs\lab5\client  444ms  python client.py
This program executes commands on the server and shows you the output.
Enter command: dir
 Volume in drive D is Data
 Volume Serial Number is 7AFF-9A89

 Directory of D:\Books\sem 6\networks\labs\lab5\server

02/14/2022  12:14 PM    <DIR>          .
02/14/2022  12:14 PM    <DIR>          ..
02/14/2022  12:23 PM             1,018 server.py
               1 File(s)          1,018 bytes
               2 Dir(s)  411,846,053,888 bytes free

Enter command:         print("Executing command: \n" + command)
Unable to initialize device PRN

Enter command: echo "echoing this on server, but the output will be shown here"
"echoing this on server, but the output will be shown here"
```

Fig: RCE Client. It sends command to server and prints the output from server.