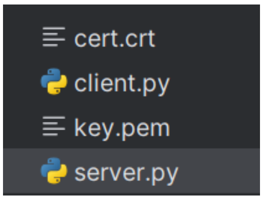
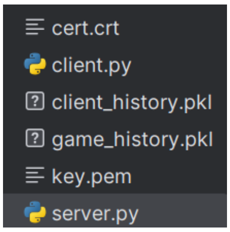
20202904 문준혁 HW.2

Before connecting the client and server.



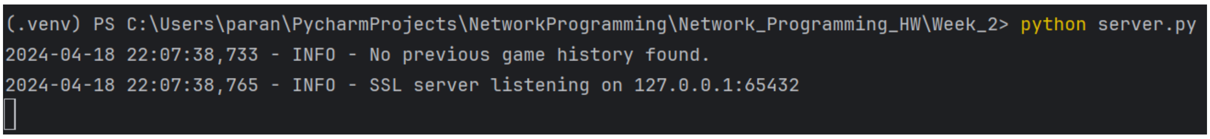
```
cert.crt
client.py
key.pem
server.py
```

After a TCP connection successfully ends between server and client -> pickle files compressed with zlib for match histories are created.



```
cert.crt
client.py
client_history.pkl
game_history.pkl
key.pem
server.py
```

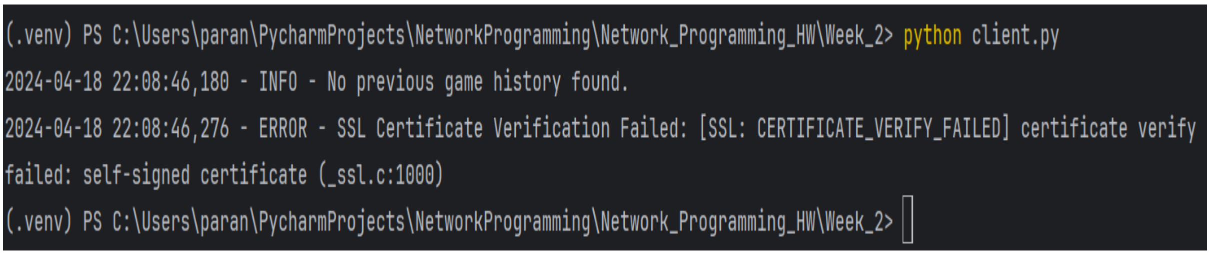
Server listening for Client



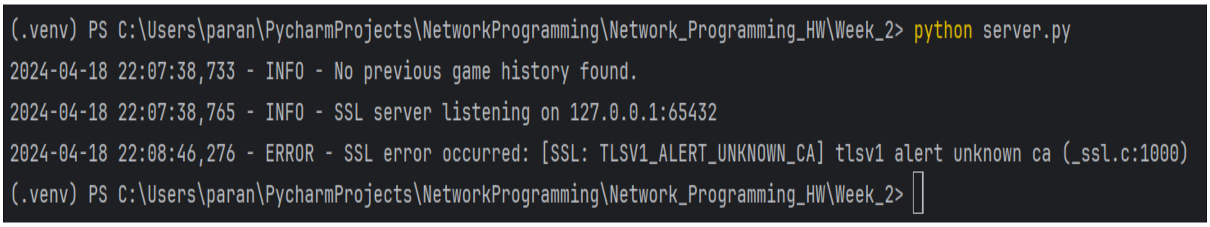
```
(.venv) PS C:\Users\paran\PycharmProjects\NetworkProgramming\Network_Programming_HW\Week_2> python server.py
2024-04-18 22:07:38,733 - INFO - No previous game history found.
2024-04-18 22:07:38,765 - INFO - SSL server listening on 127.0.0.1:65432
```

Client input "python client.py"

Connection was terminated because the certificate was not approved by the client. This was to demonstrate that invalid certificates outputs an error and terminates the connection. I followed the lecture material and used option -a for approval of the cert file.



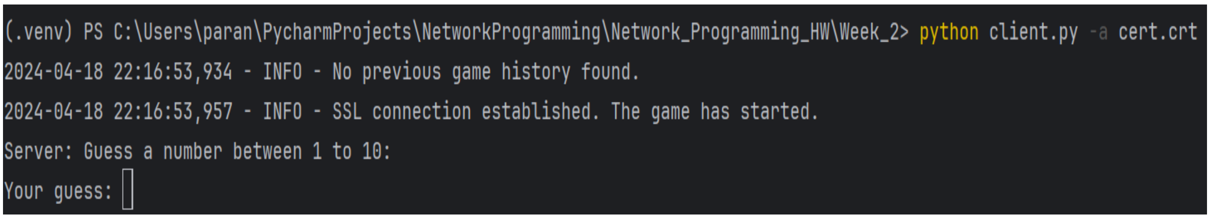
```
(.venv) PS C:\Users\paran\PycharmProjects\NetworkProgramming\Network_Programming_HW\Week_2> python client.py
2024-04-18 22:08:46,180 - INFO - No previous game history found.
2024-04-18 22:08:46,276 - ERROR - SSL Certificate Verification Failed: [SSL: CERTIFICATE_VERIFY_FAILED] certificate verify failed: self-signed certificate (_ssl.c:1000)
(.venv) PS C:\Users\paran\PycharmProjects\NetworkProgramming\Network_Programming_HW\Week_2>
```



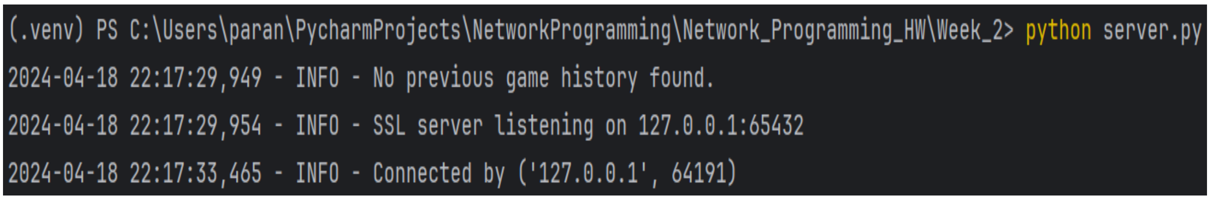
```
(.venv) PS C:\Users\paran\PycharmProjects\NetworkProgramming\Network_Programming_HW\Week_2> python server.py
2024-04-18 22:07:38,733 - INFO - No previous game history found.
2024-04-18 22:07:38,765 - INFO - SSL server listening on 127.0.0.1:65432
2024-04-18 22:08:46,276 - ERROR - SSL error occurred: [SSL: TLSV1_ALERT_UNKNOWN_CA] tlsv1 alert unknown ca (_ssl.c:1000)
(.venv) PS C:\Users\paran\PycharmProjects\NetworkProgramming\Network_Programming_HW\Week_2>
```

So we will try again with the option -a cert.crt for the client.

Now the connection was established with the cert.crt (key.pem was used in the server.py code). The pickle files are empty, since there is no previous match history, so a log outputs that no previous game history has been found. All the “INFO –“ are logs.

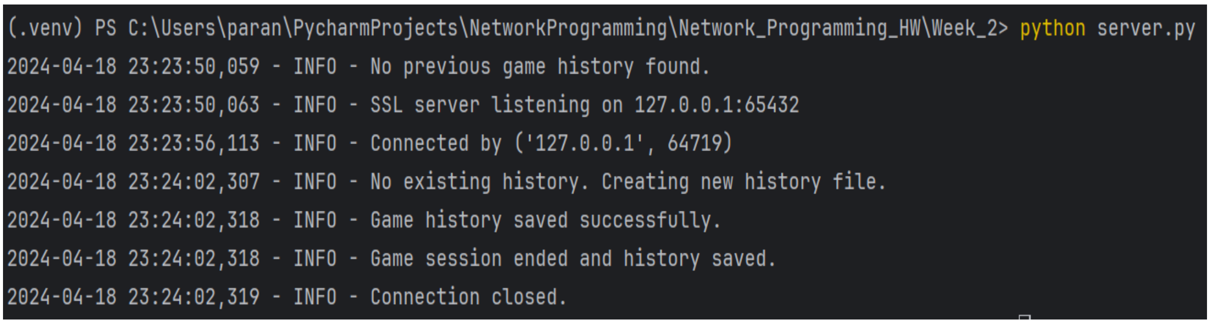


```
(.venv) PS C:\Users\paran\PycharmProjects\NetworkProgramming\Network_Programming_HW\Week_2> python client.py -a cert.crt
2024-04-18 22:16:53,934 - INFO - No previous game history found.
2024-04-18 22:16:53,957 - INFO - SSL connection established. The game has started.
Server: Guess a number between 1 to 10:
Your guess: 
```

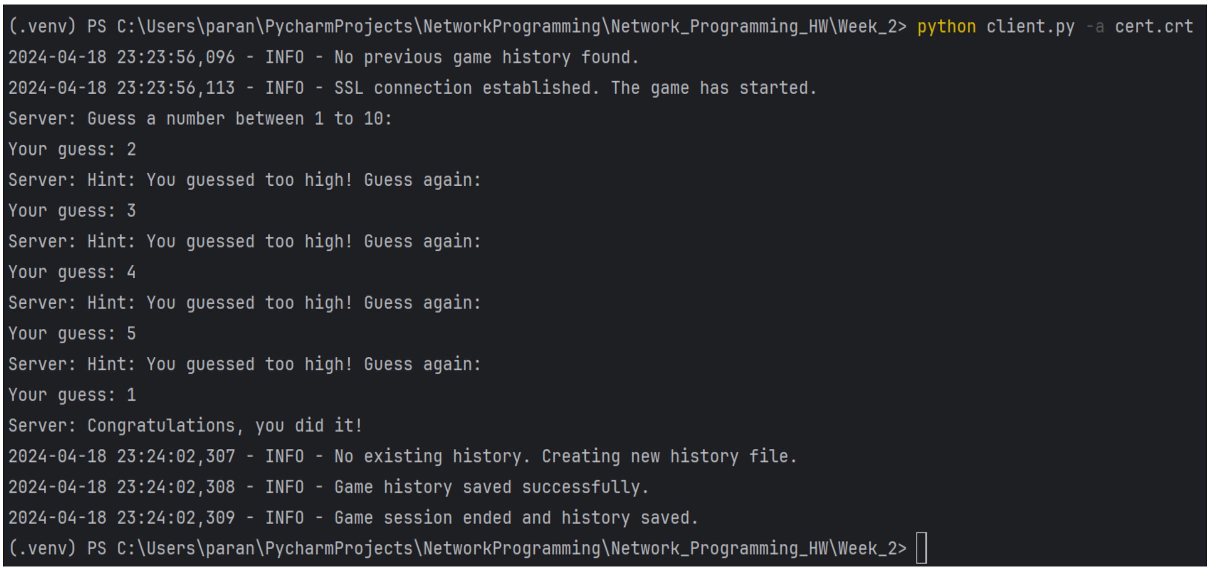


```
(.venv) PS C:\Users\paran\PycharmProjects\NetworkProgramming\Network_Programming_HW\Week_2> python server.py
2024-04-18 22:17:29,949 - INFO - No previous game history found.
2024-04-18 22:17:29,954 - INFO - SSL server listening on 127.0.0.1:65432
2024-04-18 22:17:33,465 - INFO - Connected by ('127.0.0.1', 64191)
```

This is a case where the client got the correct answer.

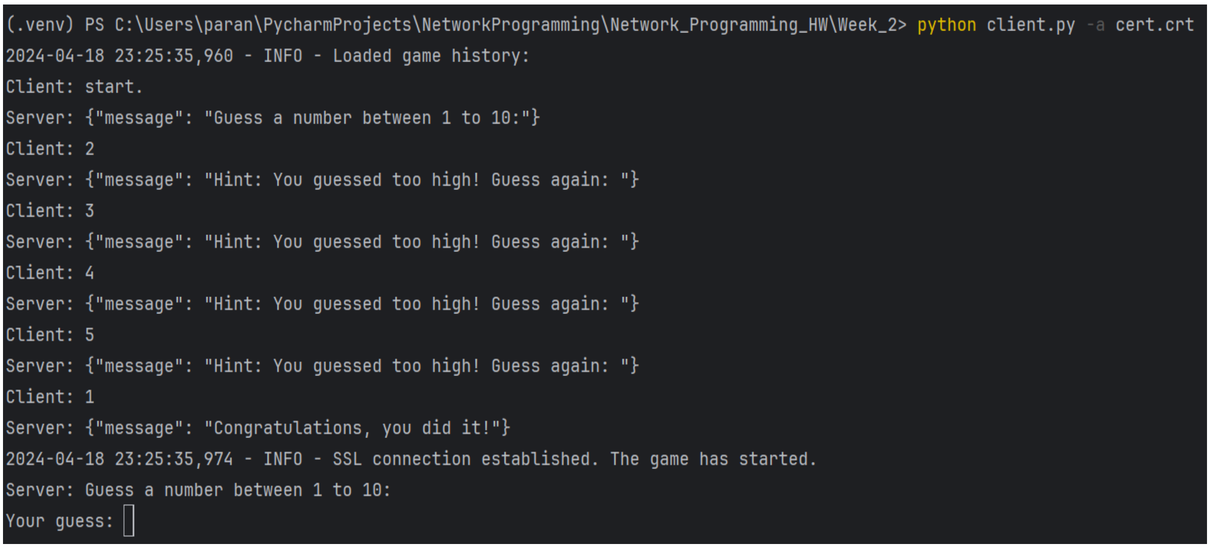


```
(.venv) PS C:\Users\paran\PycharmProjects\NetworkProgramming\Network_Programming_HW\Week_2> python server.py
2024-04-18 23:23:50,059 - INFO - No previous game history found.
2024-04-18 23:23:50,063 - INFO - SSL server listening on 127.0.0.1:65432
2024-04-18 23:23:56,113 - INFO - Connected by ('127.0.0.1', 64719)
2024-04-18 23:24:02,307 - INFO - No existing history. Creating new history file.
2024-04-18 23:24:02,318 - INFO - Game history saved successfully.
2024-04-18 23:24:02,318 - INFO - Game session ended and history saved.
2024-04-18 23:24:02,319 - INFO - Connection closed.
```

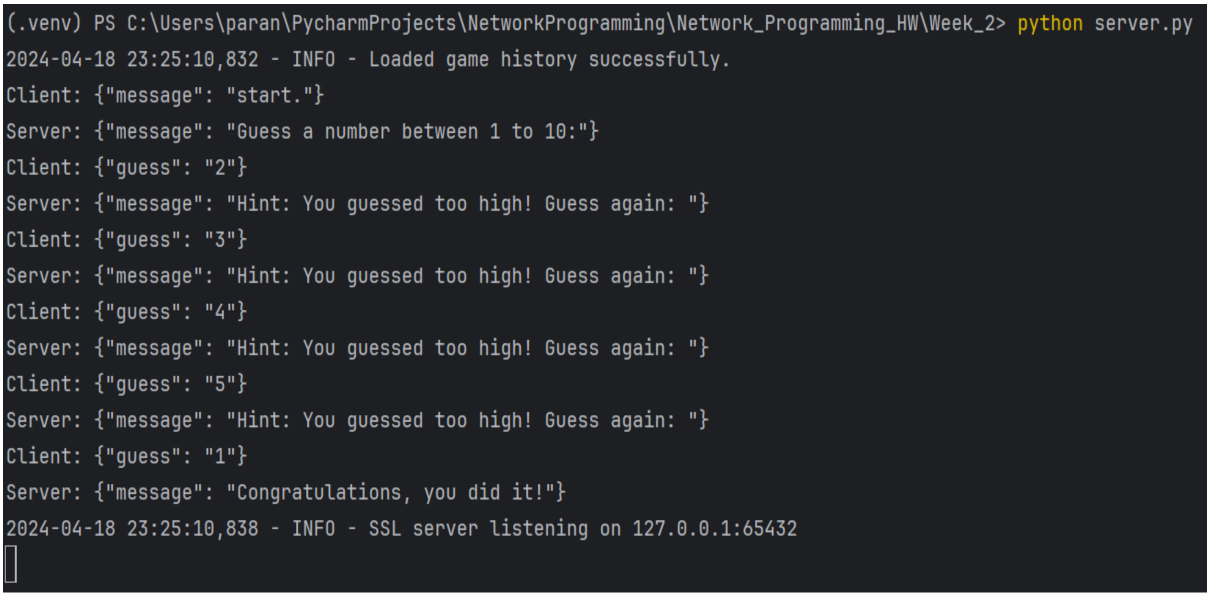


```
(.venv) PS C:\Users\paran\PycharmProjects\NetworkProgramming\Network_Programming_HW\Week_2> python client.py -a cert.crt
2024-04-18 23:23:56,096 - INFO - No previous game history found.
2024-04-18 23:23:56,113 - INFO - SSL connection established. The game has started.
Server: Guess a number between 1 to 10:
Your guess: 2
Server: Hint: You guessed too high! Guess again:
Your guess: 3
Server: Hint: You guessed too high! Guess again:
Your guess: 4
Server: Hint: You guessed too high! Guess again:
Your guess: 5
Server: Hint: You guessed too high! Guess again:
Your guess: 1
Server: Congratulations, you did it!
2024-04-18 23:24:02,307 - INFO - No existing history. Creating new history file.
2024-04-18 23:24:02,308 - INFO - Game history saved successfully.
2024-04-18 23:24:02,309 - INFO - Game session ended and history saved.
(.venv) PS C:\Users\paran\PycharmProjects\NetworkProgramming\Network_Programming_HW\Week_2> 
```

A new client_history.pkl file and game_history.pkl file is created after connection ends with the data that was transmitted between. If I start another game, the log prints. Below is the start of another game, which prints the history of the matches.

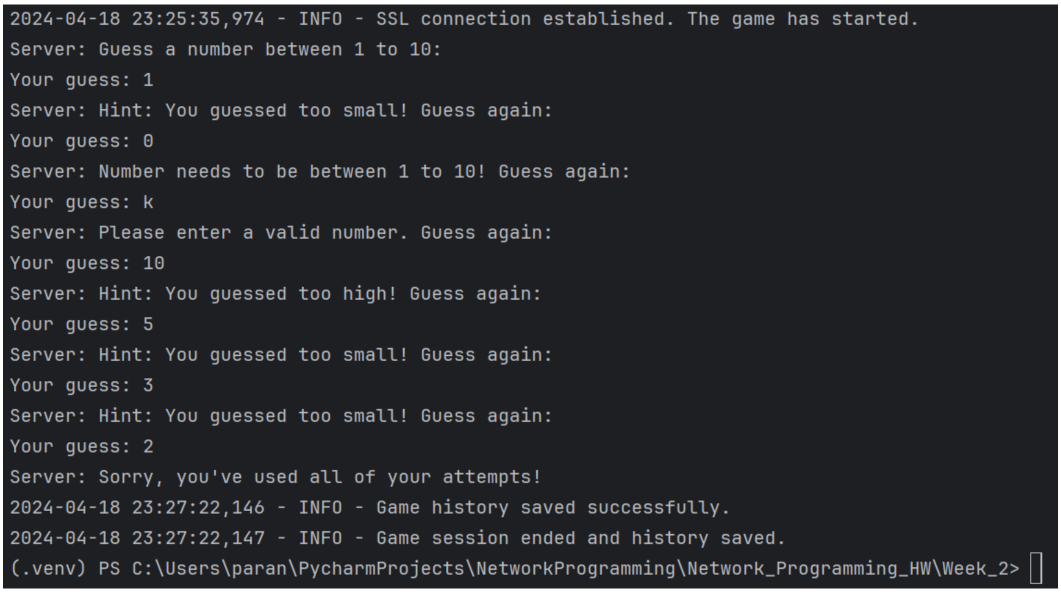


```
(.venv) PS C:\Users\paran\PycharmProjects\NetworkProgramming\Network_Programming_HW\Week_2> python client.py -a cert.crt
2024-04-18 23:25:35,960 - INFO - Loaded game history:
Client: start.
Server: {"message": "Guess a number between 1 to 10:"}
Client: 2
Server: {"message": "Hint: You guessed too high! Guess again: "}
Client: 3
Server: {"message": "Hint: You guessed too high! Guess again: "}
Client: 4
Server: {"message": "Hint: You guessed too high! Guess again: "}
Client: 5
Server: {"message": "Hint: You guessed too high! Guess again: "}
Client: 1
Server: {"message": "Congratulations, you did it!"}
2024-04-18 23:25:35,974 - INFO - SSL connection established. The game has started.
Server: Guess a number between 1 to 10:
Your guess: 
```



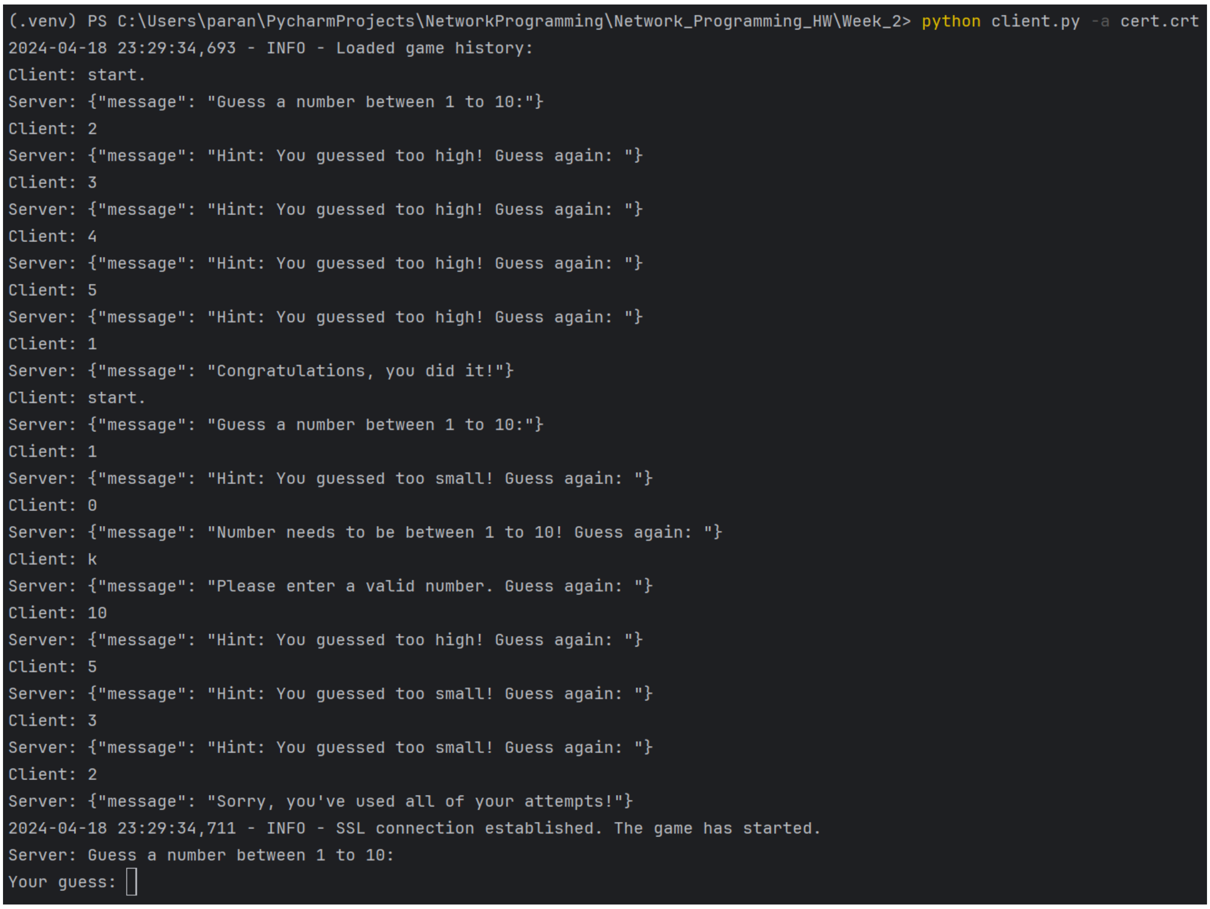
```
(.venv) PS C:\Users\paran\PycharmProjects\NetworkProgramming\Network_Programming_HW\Week_2> python server.py
2024-04-18 23:25:10,832 - INFO - Loaded game history successfully.
Client: {"message": "start."}
Server: {"message": "Guess a number between 1 to 10:"}
Client: {"guess": "2"}
Server: {"message": "Hint: You guessed too high! Guess again: "}
Client: {"guess": "3"}
Server: {"message": "Hint: You guessed too high! Guess again: "}
Client: {"guess": "4"}
Server: {"message": "Hint: You guessed too high! Guess again: "}
Client: {"guess": "5"}
Server: {"message": "Hint: You guessed too high! Guess again: "}
Client: {"guess": "1"}
Server: {"message": "Congratulations, you did it!"}
2024-04-18 23:25:10,838 - INFO - SSL server listening on 127.0.0.1:65432
```

If I run another game, the history of the matches will be appended to the previous history, which will print all match histories. For example, I ran another game from the previous match where the client doesn't get the answer.

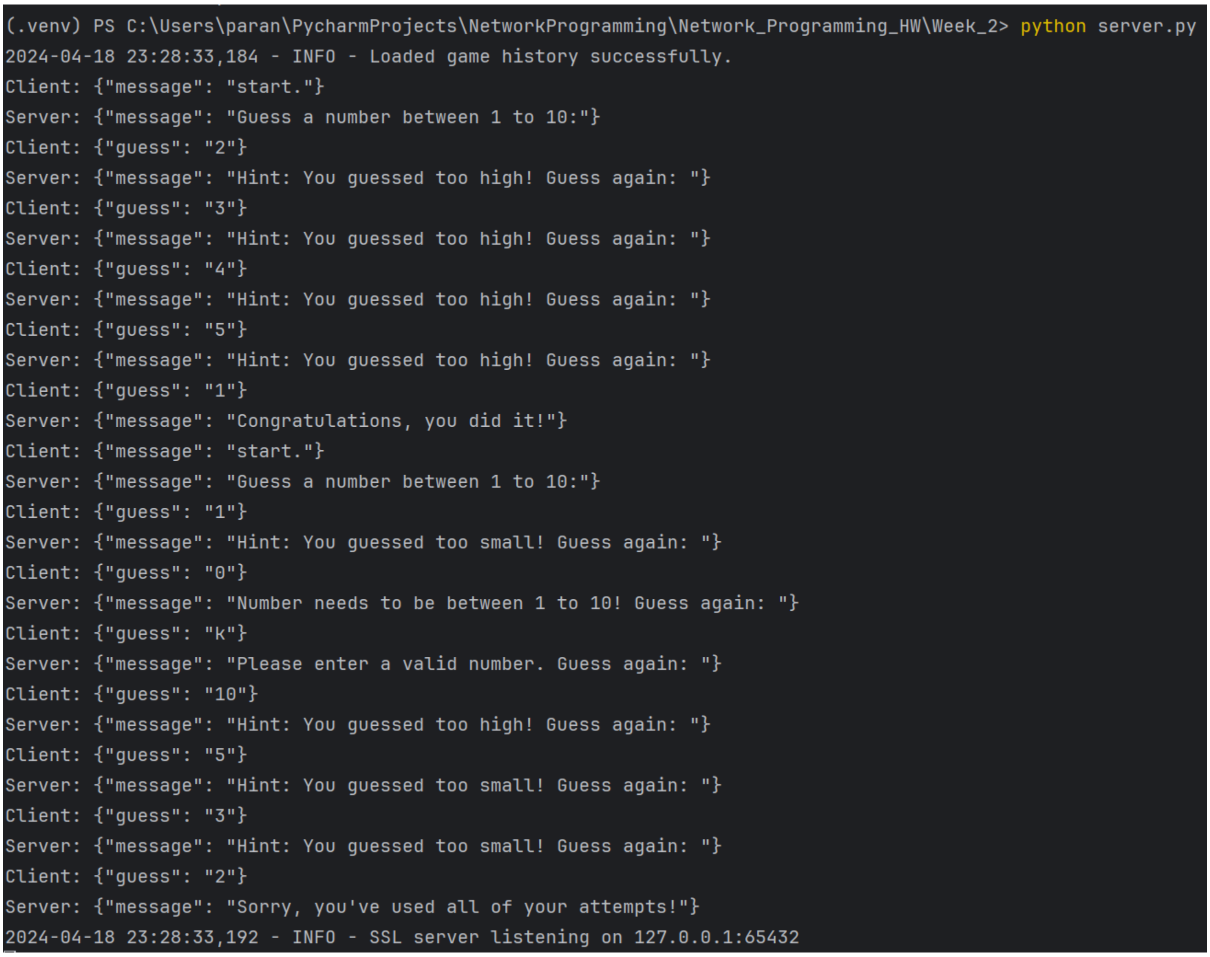


```
2024-04-18 23:25:35,974 - INFO - SSL connection established. The game has started.
Server: Guess a number between 1 to 10:
Your guess: 1
Server: Hint: You guessed too small! Guess again:
Your guess: 0
Server: Number needs to be between 1 to 10! Guess again:
Your guess: k
Server: Please enter a valid number. Guess again:
Your guess: 10
Server: Hint: You guessed too high! Guess again:
Your guess: 5
Server: Hint: You guessed too small! Guess again:
Your guess: 3
Server: Hint: You guessed too small! Guess again:
Your guess: 2
Server: Sorry, you've used all of your attempts!
2024-04-18 23:27:22,146 - INFO - Game history saved successfully.
2024-04-18 23:27:22,147 - INFO - Game session ended and history saved.
(.venv) PS C:\Users\paran\PycharmProjects\NetworkProgramming\Network_Programming_HW\Week_2>
```

If I run the game again, the server and client will print the full match histories.



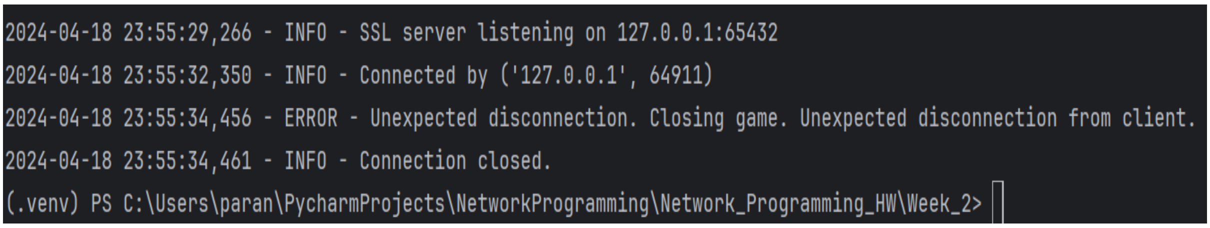
```
(.venv) PS C:\Users\paran\PycharmProjects\NetworkProgramming\Network_Programming_HW\Week_2> python client.py -a cert.crt
2024-04-18 23:29:34,693 - INFO - Loaded game history:
Client: start.
Server: {"message": "Guess a number between 1 to 10:"}
Client: 2
Server: {"message": "Hint: You guessed too high! Guess again: "}
Client: 3
Server: {"message": "Hint: You guessed too high! Guess again: "}
Client: 4
Server: {"message": "Hint: You guessed too high! Guess again: "}
Client: 5
Server: {"message": "Hint: You guessed too high! Guess again: "}
Client: 1
Server: {"message": "Congratulations, you did it!"}
Client: start.
Server: {"message": "Guess a number between 1 to 10:"}
Client: 1
Server: {"message": "Hint: You guessed too small! Guess again: "}
Client: 0
Server: {"message": "Number needs to be between 1 to 10! Guess again: "}
Client: k
Server: {"message": "Please enter a valid number. Guess again: "}
Client: 10
Server: {"message": "Hint: You guessed too high! Guess again: "}
Client: 5
Server: {"message": "Hint: You guessed too small! Guess again: "}
Client: 3
Server: {"message": "Hint: You guessed too small! Guess again: "}
Client: 2
Server: {"message": "Sorry, you've used all of your attempts!"}
2024-04-18 23:29:34,711 - INFO - SSL connection established. The game has started.
Server: Guess a number between 1 to 10:
Your guess:
```



```
(.venv) PS C:\Users\paran\PycharmProjects\NetworkProgramming\Network_Programming_HW\Week_2> python server.py
2024-04-18 23:28:33,184 - INFO - Loaded game history successfully.
Client: {"message": "start."}
Server: {"message": "Guess a number between 1 to 10:"}
Client: {"guess": "2"}
Server: {"message": "Hint: You guessed too high! Guess again: "}
Client: {"guess": "3"}
Server: {"message": "Hint: You guessed too high! Guess again: "}
Client: {"guess": "4"}
Server: {"message": "Hint: You guessed too high! Guess again: "}
Client: {"guess": "5"}
Server: {"message": "Hint: You guessed too high! Guess again: "}
Client: {"guess": "1"}
Server: {"message": "Congratulations, you did it!"}
Client: {"message": "start."}
Server: {"message": "Guess a number between 1 to 10:"}
Client: {"guess": "1"}
Server: {"message": "Hint: You guessed too small! Guess again: "}
Client: {"guess": "0"}
Server: {"message": "Number needs to be between 1 to 10! Guess again: "}
Client: {"guess": "k"}
Server: {"message": "Please enter a valid number. Guess again: "}
Client: {"guess": "10"}
Server: {"message": "Hint: You guessed too high! Guess again: "}
Client: {"guess": "5"}
Server: {"message": "Hint: You guessed too small! Guess again: "}
Client: {"guess": "3"}
Server: {"message": "Hint: You guessed too small! Guess again: "}
Client: {"guess": "2"}
Server: {"message": "Sorry, you've used all of your attempts!"}
2024-04-18 23:28:33,192 - INFO - SSL server listening on 127.0.0.1:65432
```

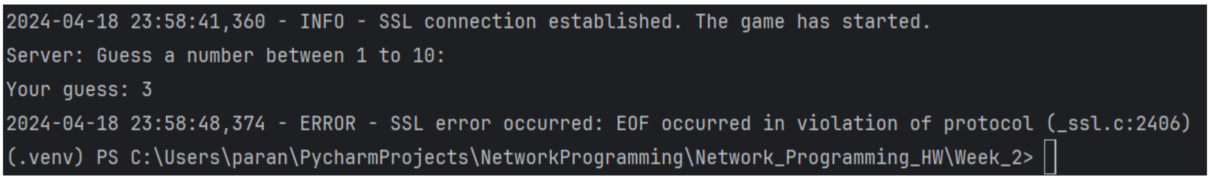
Below is an instance where the client disconnects during the connection.

The unexpected disconnection is logged. The history match that was going on isn't saved.



```
2024-04-18 23:55:29,266 - INFO - SSL server listening on 127.0.0.1:65432
2024-04-18 23:55:32,350 - INFO - Connected by ('127.0.0.1', 64911)
2024-04-18 23:55:34,456 - ERROR - Unexpected disconnection. Closing game. Unexpected disconnection from client.
2024-04-18 23:55:34,461 - INFO - Connection closed.
(.venv) PS C:\Users\paran\PycharmProjects\NetworkProgramming\Network_Programming_HW\Week_2>
```

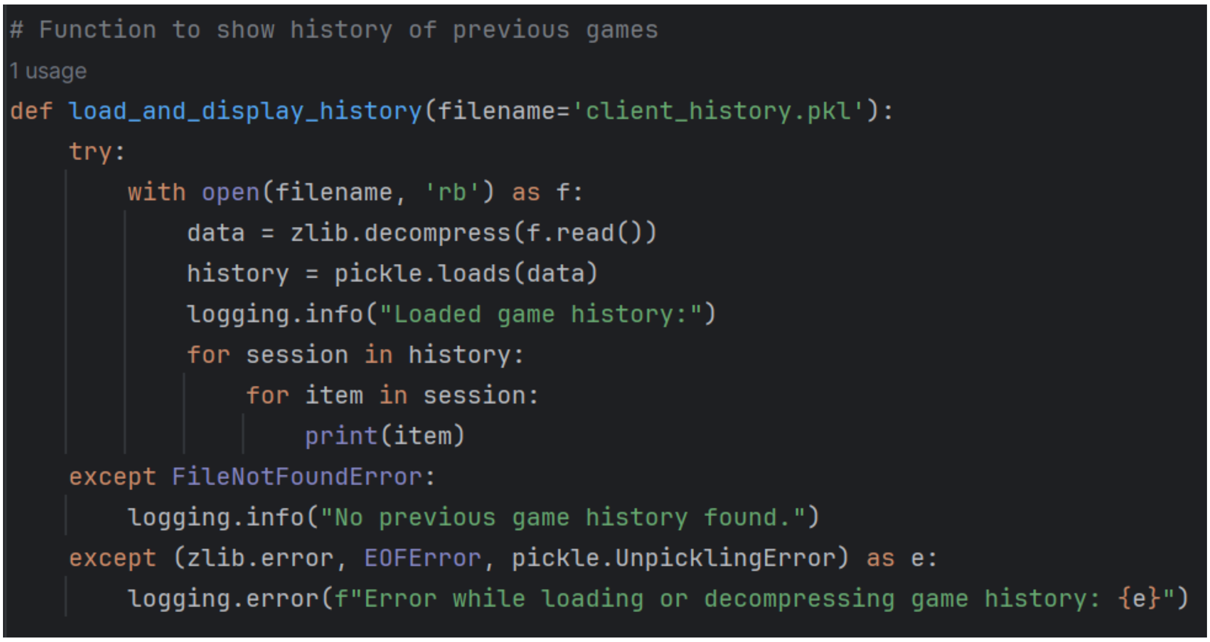
If the server disconnects, the client was in typing mode, so if it tries to send the guess to the server, an SSL error is raised since we shouldn't be sending anything when the SSL connection was severed.



```
2024-04-18 23:58:41,360 - INFO - SSL connection established. The game has started.
Server: Guess a number between 1 to 10:
Your guess: 3
2024-04-18 23:58:48,374 - ERROR - SSL error occurred: EOF occurred in violation of protocol (_ssl.c:2406)
(.venv) PS C:\Users\paran\PycharmProjects\NetworkProgramming\Network_Programming_HW\Week_2>
```

load_and_display_history(filename)

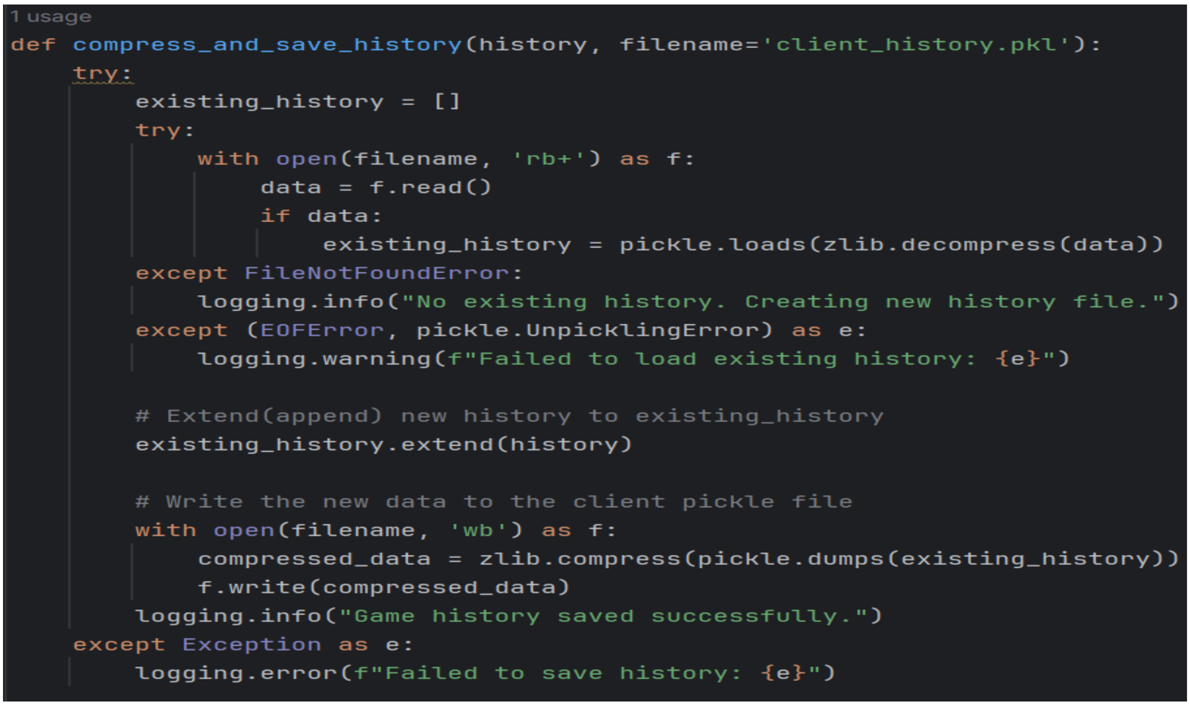
If the file name ('client_history.pkl' or 'game_history.pkl') exists, decompress the zlib and use loads(), and print all items within using a double for loop. If file doesn't exist, print no game history found. If error decompressing or loading the pkl file, print error.



```
# Function to show history of previous games
1 usage
def load_and_display_history(filename='client_history.pkl'):
    try:
        with open(filename, 'rb') as f:
            data = zlib.decompress(f.read())
            history = pickle.loads(data)
            logging.info("Loaded game history:")
            for session in history:
                for item in session:
                    print(item)
    except FileNotFoundError:
        logging.info("No previous game history found.")
    except (zlib.error, EOFError, pickle.UnpicklingError) as e:
        logging.error(f"Error while loading or decompressing game history: {e}")
```

compress_and_save_history(history, filename)

If file exists and game successfully ended, append the current match history to the previous history



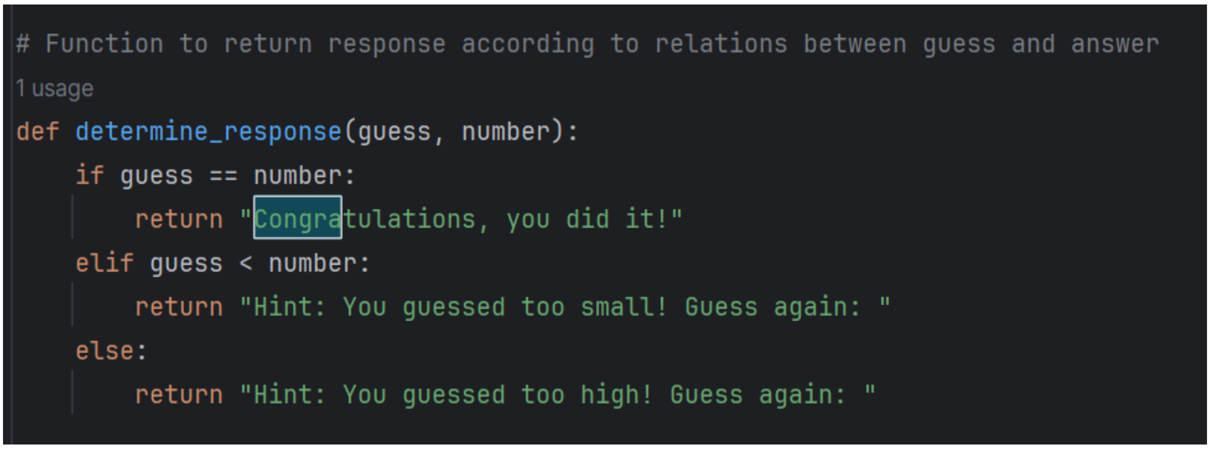
```
1 usage
def compress_and_save_history(history, filename='client_history.pkl'):
    try:
        existing_history = []
        try:
            with open(filename, 'rb+') as f:
                data = f.read()
                if data:
                    existing_history = pickle.loads(zlib.decompress(data))
        except FileNotFoundError:
            logging.info("No existing history. Creating new history file.")
        except (EOFError, pickle.UnpicklingError) as e:
            logging.warning(f"Failed to load existing history: {e}")

        # Extend(append) new history to existing_history
        existing_history.extend(history)

        # Write the new data to the client pickle file
        with open(filename, 'wb') as f:
            compressed_data = zlib.compress(pickle.dumps(existing_history))
            f.write(compressed_data)
        logging.info("Game history saved successfully.")
    except Exception as e:
        logging.error(f"Failed to save history: {e}")
```

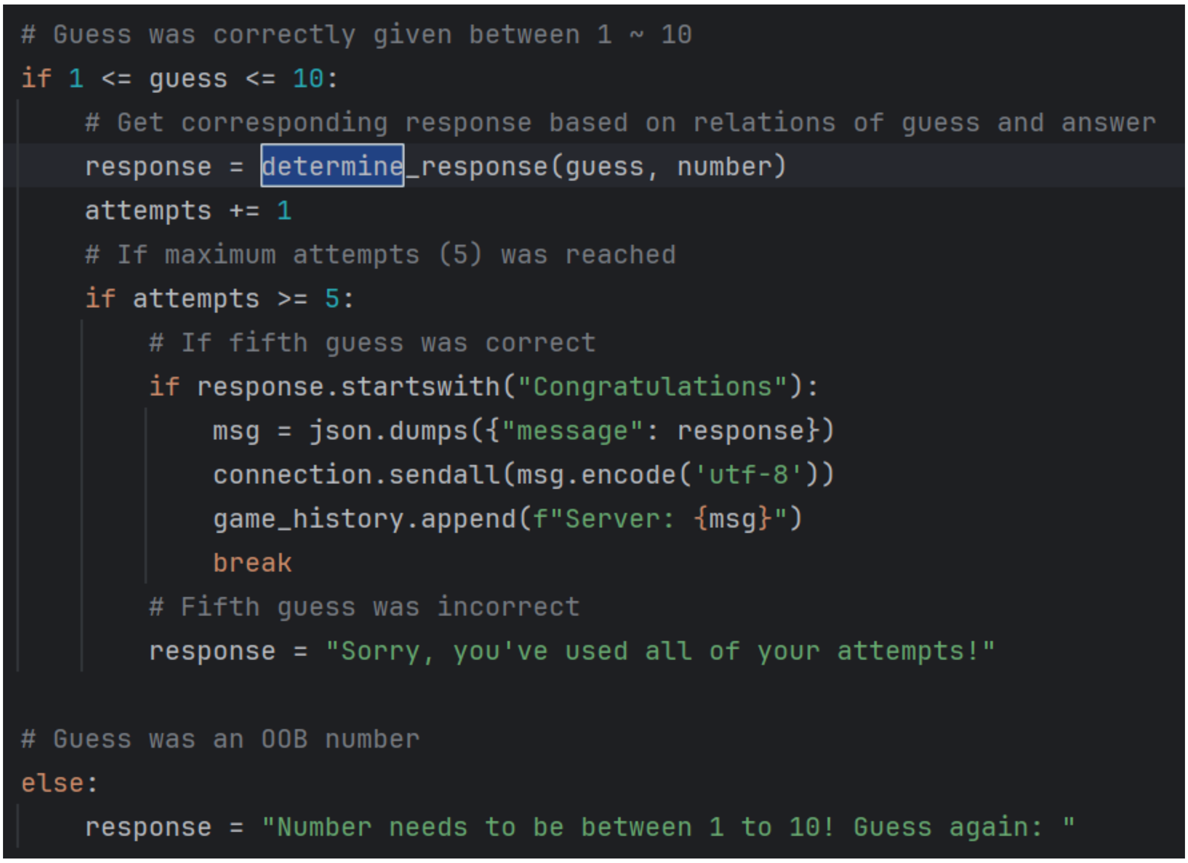
determine_response(guess, number)

Returns response based on guess and number



```
# Function to return response according to relations between guess and answer
1 usage
def determine_response(guess, number):
    if guess == number:
        return "Congratulations, you did it!"
    elif guess < number:
        return "Hint: You guessed too small! Guess again: "
    else:
        return "Hint: You guessed too high! Guess again: "
```

Implemented the code so that for the fifth attempt, if the response had “congratulations” in it, it would be a success. Otherwise, the “Sorry, ~” is set for response.



```
# Guess was correctly given between 1 ~ 10
if 1 <= guess <= 10:
    # Get corresponding response based on relations of guess and answer
    response = determine_response(guess, number)
    attempts += 1
    # If maximum attempts (5) was reached
    if attempts >= 5:
        # If fifth guess was correct
        if response.startswith("Congratulations"):
            msg = json.dumps({"message": response})
            connection.sendall(msg.encode('utf-8'))
            game_history.append(f"Server: {msg}")
            break
        # Fifth guess was incorrect
        response = "Sorry, you've used all of your attempts!"

# Guess was an OOB number
else:
    response = "Number needs to be between 1 to 10! Guess again: "
```