

## Protocol Specification:

### ----- Server Commands: -----

Execution of the server is done using the following command: `./numbers_server.py users_file [port]` "port" is the port number on which the server listens for connections. If not provided, the program will default to 1337.

1. User Login: Command: User: username  
Password: password (including enter between the lines) Response: If success: Hi username, good to see you. If failure: Failed to login.
2. Calculation: Command: calculate: X operator Z (e.g., calculate: 5 + 3), when, X and Z are integers and operator is one of the following: +, -, x, /. If the input received from the client is in the correct format: Server calculates the result (result = X op Z) and responds with the message: response : result. Else, the server closes the connection with the client.
3. Palindrome Check: Command: is\_palindrome: number (e.g., is\_palindrome: 12321) Response: If number is palindrom: response: Yes Else: response: No If the data received from the client does not adhere to the expected format (e.g., the input is not valid integer), the server terminates the connection with the client.
4. Prime Number Check: Command: is\_primary: number (e.g., is\_primary: 13) Response: If number is primry: response: Yes Else: response: No If the data received from the client does not adhere to the expected format (e.g., the input is not valid integer), the server terminates the connection with the client.
5. Quit Command: Command: quit. Response: Close the connection with the client. Remove the client from the list of active sockets. If the client was authenticated, remove it from the list of authenticated sockets.

### ----- Client Commands: -----

Execution of the client is done using the following command: `./numbers_client.py [hostname [port]]` hostname and port number are optional parameters.

- "hostname" refers to the server's hostname or IP address. If not provided, the program will default to localhost.
  - "port" is the port number on which the server listens for connections. If not provided, the program will default to 1337.
1. User Authentication: When a client connects to the server, the server sends him a message: Welcome!, Please log in. Client sends the username and password. If successful, the server responds with the message: Hi , good to see you. Otherwise, responds with the message: Failed to login. If the login attempt fails, the client has the option to retry by entering a valid username and password.
  2. Calculation Request: Client sends a command in the format: calculate: X operator Z (e.g., calculate: 5 + 3), when X and Z are integers and operator is one of the following: +, -, x, /. If the input received from the client is in the correct format: Server calculates the result (result = X op Z) and responds with the message: response : result. Else, the server closes the connection with the client.

3. Palindrome Check Request: Client sends a command in the format: is\_palindrome: number (e.g., is\_palindrome: 12321). If the input received from the client is in the correct format: Server responds with the message: If number is palindrom: response: Yes else: response: No Else, the server closes the connection with the client.
4. Prime Number Check Request: Client sends a command in the format: is\_primary: number (e.g., is\_primary: 12). If the input received from the client is in the correct format: Server responds with the message: If number is primary: response: Yes else: response: No Else, the server closes the connection with the client.
5. Quit: Client can send : "quit" message to disconnect from the server (if the client has already authenticated). Then, the client will close the connection.

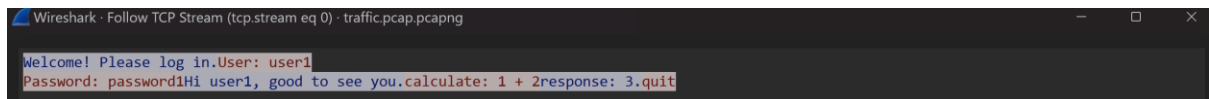
בחרנו להראות תהליך רגיל של יוזר אל מול השרת, בו היוזר מתחבר, מבצע פעולה ומתנתק.

השרת בכחול, הלקוח בכתום.

ראשית בהתחברות, השרת שולח ללקוח את הודעת הכניסה.

היוזר מבצע כניסה כדרוש, ולאחריה מופיעה הודעת ההתחברות המוצלחת.

לאחר מכן, היוזר מבקש מהשרת לבצע פעולת חישוב, השרת משיב את התשובה כראוי ולאחר מכן היוזר בוחר להתנתק ולסיים את הסשן אל מול השרת.



Wireshark · Follow TCP Stream (tcp.stream eq 0) · traffic.pcap.pcapng

```
Welcome! Please log in.User: user1
Password: password1Hi user1, good to see you.calculate: 1 + 2response: 3.quit
```