UDP Client-Server Hash Exchange (Client)

Description

In this problem, you are asked to test the interaction between a server and a client through socket programming. The client will send a text message to the server, the server will calculate the MD5 hash of the received message, and then send this hash back to the client. The client should display the original message and its corresponding hash. Note that you need to use UDP for this problem.

Input

There is no input for this problem as the interaction happens over a network connection established between the server and the client. The client will connect to the server at localhost (127.0.0.1) at port 12345, send a specific message to the server, and receive a hash response.

Output

The client should output both the original message sent and the hash received from the server to standard output. The expected output format is (you do not need to write the unit test as it is provided in the skeleton, and please note that [hashed message] contains a string like '472a37a0e1a7bd2d989c93af9867210a'):

```
Original message: Hello, Server! Please hash this message.
Received hash: 1df1eb001708d60aa5b8c2f33d99f529
sendto called with: call(b'Hello, Server! Please hash this message.', ('127.0.0.1', 12345))
recvfrom called with: call(1024)
close called with: call()
```

Method

Your task is to implement and run both the server and client programs as provided. Ensure the client successfully connects to the server, sends the message, receives the hash, and prints the exact expected output.

- Server program: The server should start, bind to the specified localhost and port, listen for incoming connections, accept a client connection, receive the message from the client, calculate its MD5 hash, send the hash back to the client, and then close the connection. It must handle a single client connection before shutting down for the purpose of this problem.
- Client program: The client should connect to the server's specified host and port, send the message "Hello, Server! Please hash this message." to the server, receive the hash response, print both the original message and the hash in the specified format, and then close the connection.