String Reverser (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 reverse the message, and then send this reversed message back to the client. The client should display the original message and the reversed message.

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 12346, send a specific message to the server, and receive a reversed message as a response.

Output

The client should output both the original message sent and the reversed message received from the server to standard output. The expected output format is (please note that the reversed message is hardcoded and must match exactly):

Original message: Hello, Server! Please reverse this message.

Received reversed message: .egassem siht esrever esaelP !revreS ,olleH

connect called with: call(('127.0.0.1', 12346))

send called with: call(b'Hello, Server! Please reverse this message.')

recv 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 reversed message, and prints the exact expected output.

- Client Program: The client should connect to the server at the specified host and port, send the message "Hello, Server! Please reverse this message.", receive the reversed message from the

server, print both the original and the reversed message, and close the connection.

Evaluation

The submission will be evaluated based on the following criteria:

- The client must successfully connect to the server, send the message, receive the reversed message, and print the output in the specified format.
- The use of unit tests for the client to ensure the correct behavior of sending and receiving the data.