String Reverser (Server)

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 send the reversed version back to the client.

Input

There is no input for this problem as the interaction happens over a network connection established between the server and the client. The server will listen on localhost (127.0.0.1) at port 12346, receive a specific message from the client, reverse the message, and send it back.

Output

The server should reverse the received message and send it back to the client. The complete output of the program is as follows (note that the reversed message is hardcoded and must match exactly):

```
Test handle_client_connection ...
Got a connection from ('127.0.0.1', 12345)
recv called with: call(1024)
send called with: call(b'.egassem siht esrever esaelP !revreS ,olleH')
close called with: call()
Test start_server ...
Listening on 127.0.0.1:12346 ...
Got a connection from ('127.0.0.1', 12345)
accept called with: call()
bind called with: call(('127.0.0.1', 12346))
listen called with: call(1)
```

Method

Your task is to implement and run both the server and client programs as provided. Ensure the server successfully binds to the specified

port, listens for incoming connections, accepts a client connection, receives the message, reverses it, sends it back to the client, and then closes the connection.

- Server Program: The server should bind to the specified localhost and port, listen for incoming connections, accept a client connection, receive the message from the client, reverse it, send it back, and then close the connection. It must handle a single client connection before shutting down for the purpose of this problem.

Evaluation

The submission will be evaluated based on the following criteria:

- The server must successfully start, receive the client's message, reverse it, send it back, and close the connection.
- The use of unit tests for the server to ensure the correct behavior of receiving and sending the reversed message.