**LAB 5**

**Socket Programming**

|  |
| --- |
| Họ tên và MSSV: Lý Đại Phát  Nhóm học phần: CT293 Nhóm 02 |

* *Các sinh viên bị phát hiện sao chép bài của nhau sẽ nhận 0đ cho tất cả bài thực hành của môn này.*
* *Bài nộp phải ở dạng PDF, hình minh họa phải rõ ràng chi tiết.* Hình minh hoạ chỉ cần chụp ở nội dung thực hiện, không chụp toàn màn hình.

**Github repo:** https://github.com/xiaojiudev/CT293\_Lab5

- Chứa các code minh hoạ, hình ảnh kết quả thực thi code.

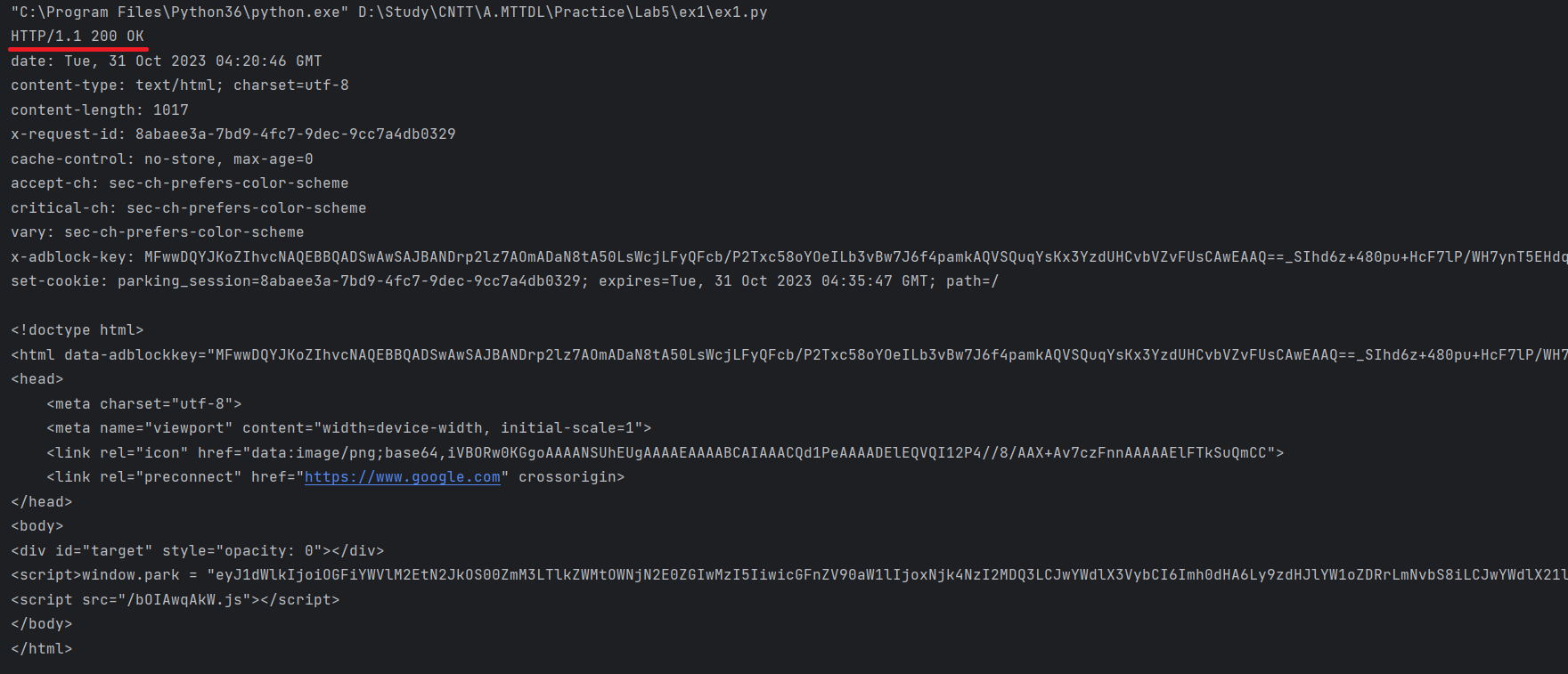
1. **Bài 1**

Viết chương trình nhận đối số là một URL (Ví dụ: www.cit.ctu.edu.vn). Sử dụng TCP socket nối kết đến web server trong URL để lấy file HTML về, và hiển thị nội dung file HTML đó ra màn hình.

(Chụp hình minh họa code bài làm và kết quả thực thi; đính kèm tập tin code khi nộp bài.)



Hình 1 Mã nguồn ex1.py



Hình 2 Kết quả thu được khi thực thi ex1.py (URL được dùng http://streamhd4k.com)

1. **Bài 2**

Viết chương trình theo mô hình Client-Server sử dụng TCP Socket.

Trong đó:

* Server lắng nghe ở cổng 8888, làm nhiệm vụ đọc một ký tự số từ '0' đến '9'.

(Ví dụ: nhận số 0, trả về "khong"; 1, trả về "mot" ; 9, trả về "chin"; nhận ký tự khác số thì trả về "Không phải số nguyên" ).

* Client sẽ nhập vào 1 ký tự, gửi qua Server, nhận kết quả trả về từ Server và thể hiện lên màn hình.

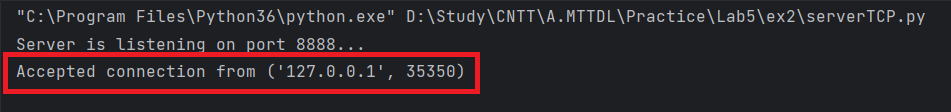
(Chụp hình minh họa code bài làm và kết quả thực thi; đính kèm tập tin code khi nộp bài.)



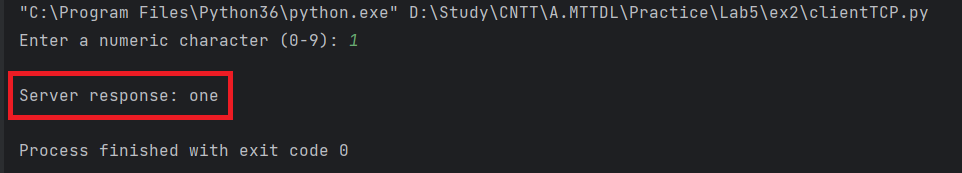
Hình 3 Mã nguồn serverTCP.py



Hình 4 Mã nguồn clientTCP.py



Hình 5 Kết quả thực thi serverTCP.py và clientTCP.py (1)



Hình 6 Kết quả thực thi serverTCP.py và clientTCP.py (2)

1. **Bài 3**

Viết chương trình theo mô hình Client-Server sử dụng UDP Socket.

Trong đó:

* Server hoạt động ở cổng 8888, sẽ nhận từ Client các yêu cầu là một chuỗi có khuôn dạng như sau:

*"OP Operant1 Operant2\n" Ví dụ: “+ 100 200”*

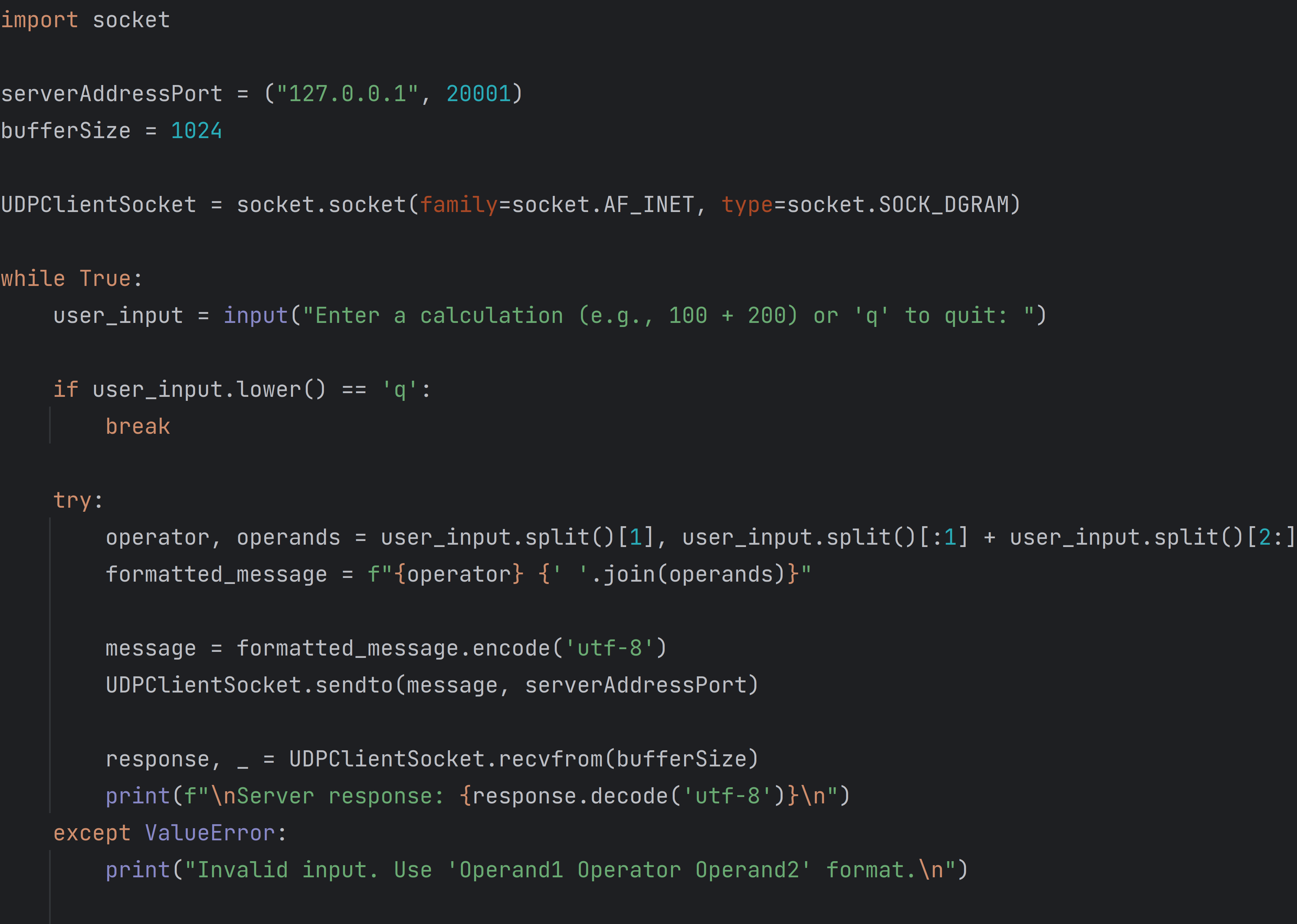
Trong đó:

* OP là một ký tự chỉ phép toán muốn thực hiện: '+' , '-' , '\*' , '/'
* Operant1, Operant2 là đối số của phép toán.
* Các thành phần trên cách nhau bởi 1 ký tự trắng ' '
* Mỗi khi server nhận được một thông điệp nó sẽ thực hiện phép toán “Operant1 OP Operant2” để cho ra kết quả sau đó đổi kết quá thành chuỗi và gửi về Client.
* Client cho phép người dùng nhập các phép toán muốn tính theo cách thức thông thường, ví dụ: 100 + 200. Client tạo ra thông điệp yêu cầu theo đúng dạng do Server quy định rồi gửi sang Server, chờ nhận kết quả trả về và in ra màn hình.
* Kết thúc yêu cầu bằng ký tự xuống dòng '\n'

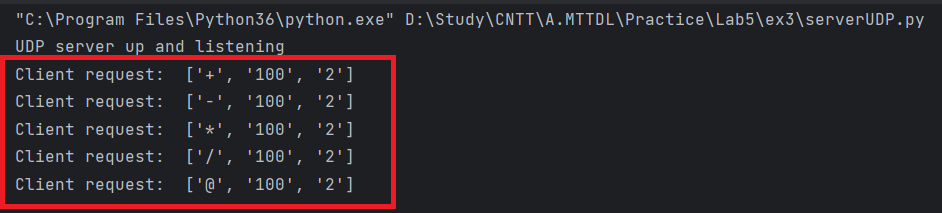
(Chụp hình minh họa code bài làm và kết quả thực thi; đính kèm tập tin code khi nộp bài.)



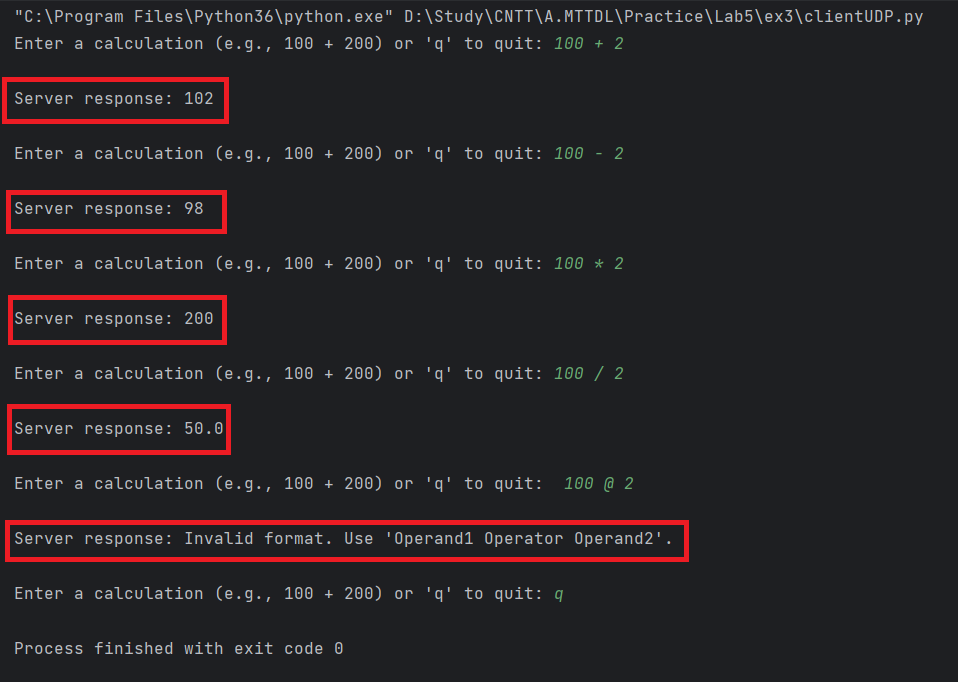
Hình 7 Mã nguồn serverUDP.py



Hình 8 Mã nguồn clientUDP.py



Hình 9 Kết quả thực thi serverUDP.py và clientUDP.py (1)



Hình 10 Kết quả thực thi serverUDP.py và clientUDP.py (2)

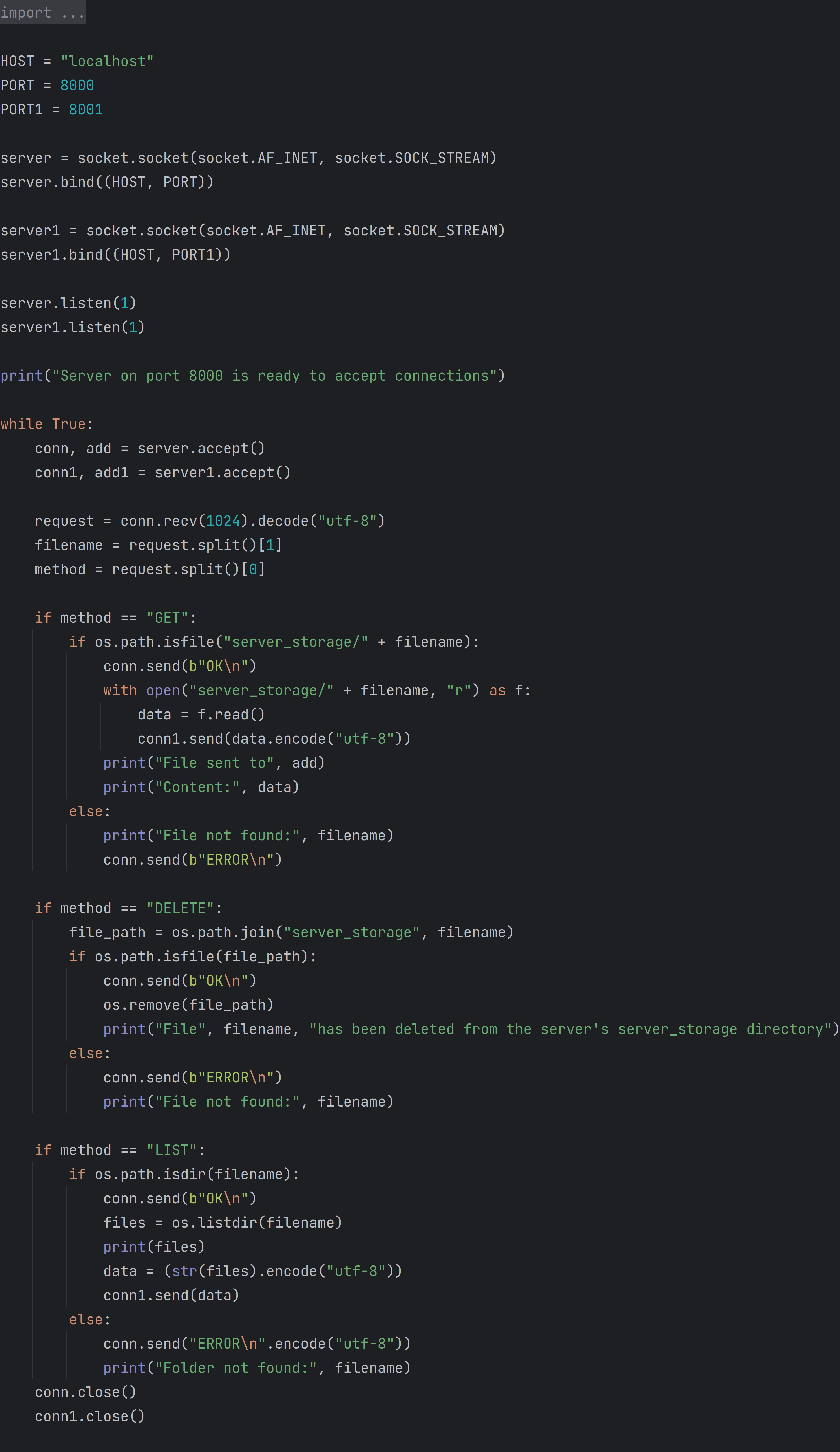
1. **Bài 4**

Viết chương trình theo mô hình Client-Server sử dụng TCP socket.

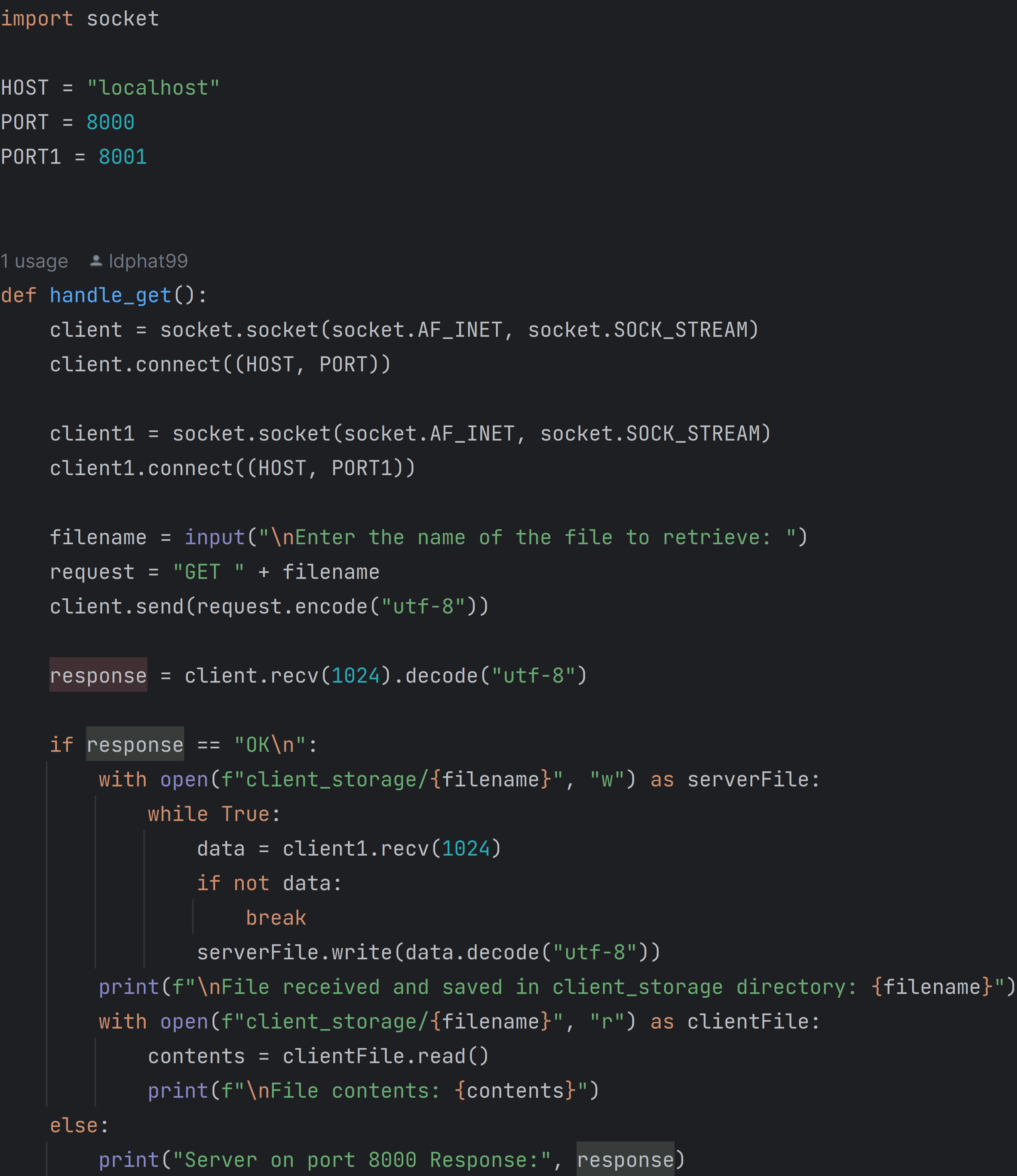
Trong đó,

* Server hoạt động ở 2 cổng (2 TCP socket) 8001 và 8002:
* Cổng 8000: sử dụng để nhận các lệnh từ Client và gửi phản hồi cho Client.
* Cổng 8001: sử dụng để gửi dữ liệu tập tin/thư mục tới Client.
* Ở cổng 8000, Server sẽ nhận từ Client các câu lệnh có dạng như sau:
* “GET <Tên file>”.
  + Nếu file yêu cầu tồn tại: Server sẽ gửi trả cho Client chuỗi “OK\n”. Sau đó là nội dung file sẽ được gửi tới Client qua cổng 8001.
* Nếu file không tồn tại: Server sẽ gửi trả cho Client chuỗi “ERROR\n”.
* Client nhận nội dung tập tin từ Server và lưu lại.
* “DELETE <Tên file>”.
  + Nếu file yêu cầu tồn tại: Server sẽ gửi trả cho Client chuỗi “OK\n”. Xóa file đó trên thư mục hiện hành của Server.
* Nếu file không tồn tại: Server sẽ gửi trả cho Client chuỗi “ERROR\n”.
* “LIST <Tên thư mục>”.
  + Nếu thư mục yêu cầu tồn tại: Server sẽ gửi trả cho Client chuỗi “OK\n”. Sau đó là nội dung thư mục sẽ được gửi tới Client qua cổng 8001.
* Nếu thư mục không tồn tại: Server sẽ gửi trả cho Client chuỗi “ERROR\n”.
* Client nhận nội dung thư mục từ Server và in ra màn hình.
* Client sẽ cho phép người dùng gửi các lệnh qua Server và nhận phản hồi ở cổng 8000.
* Kết thúc yêu cầu bằng ký tự xuống dòng '\n'

(Chụp hình minh họa code bài làm và kết quả thực thi; đính kèm tập tin code khi nộp bài.)



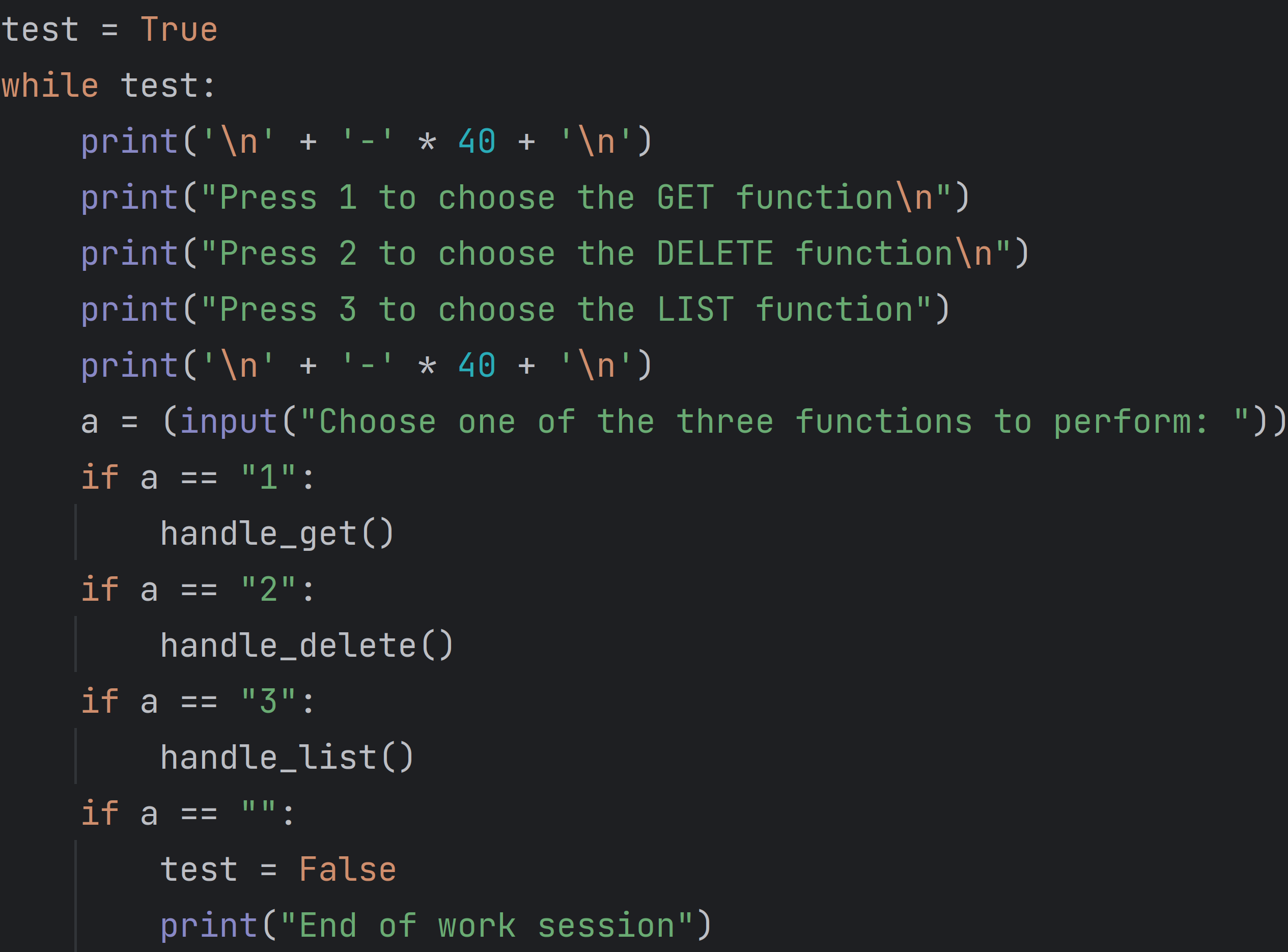
Hình 11 Mã nguồn server.py



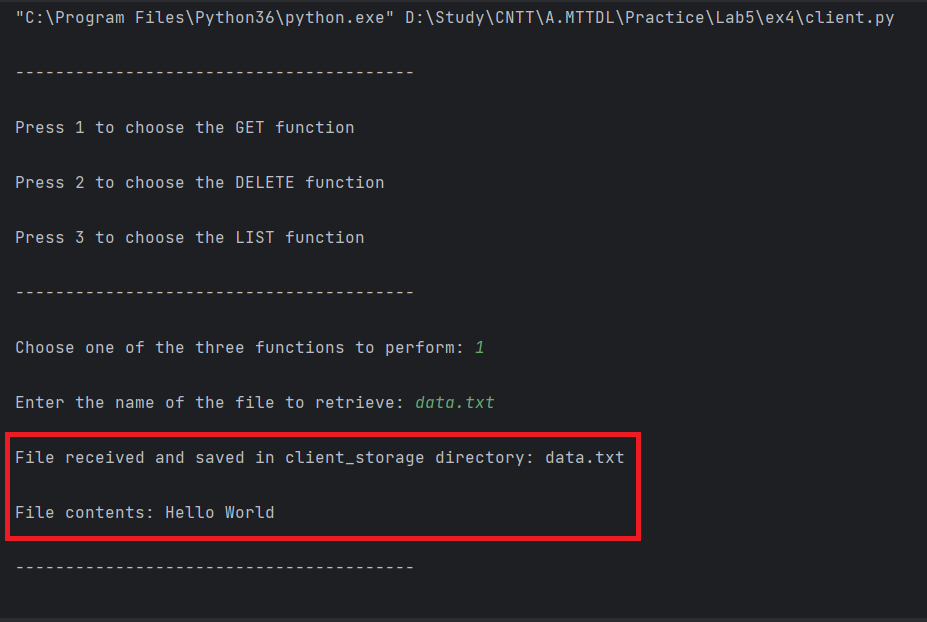
Hình 12 Mã nguồn client.py (1)



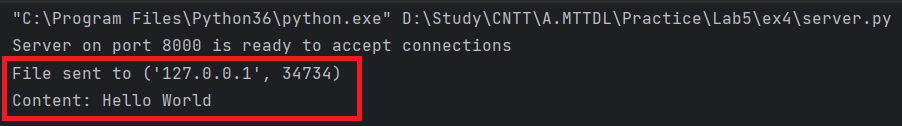
Hình 13 Mã nguồn client.py (2)



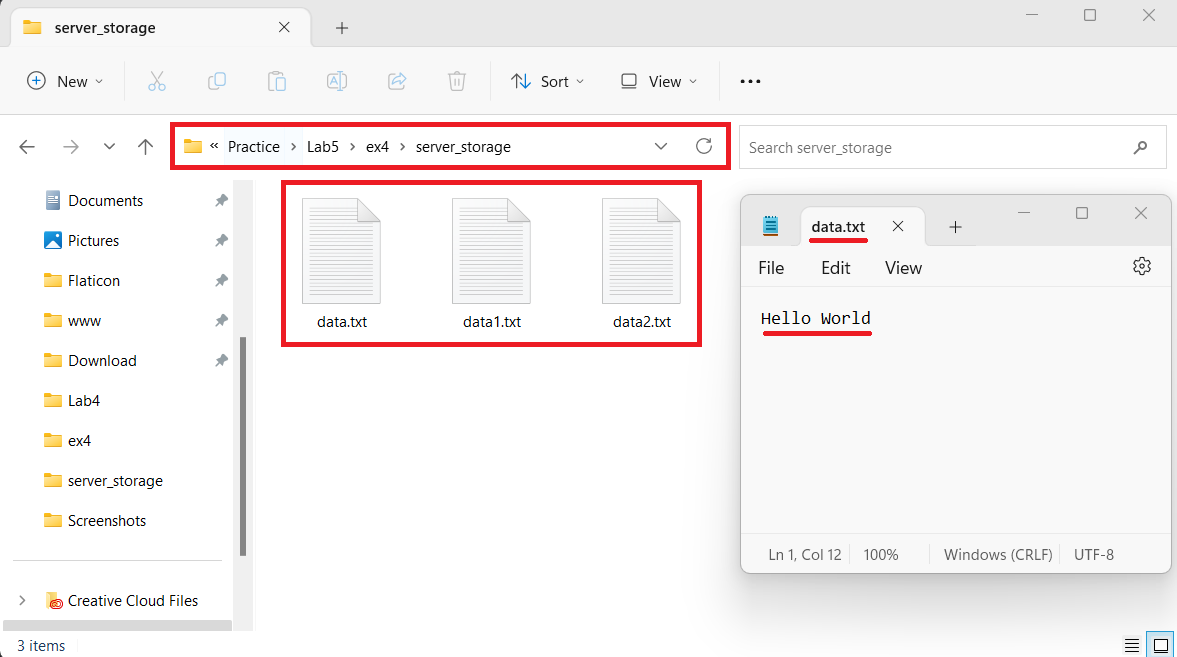
Hình 14 Mã nguồn client.py (3)



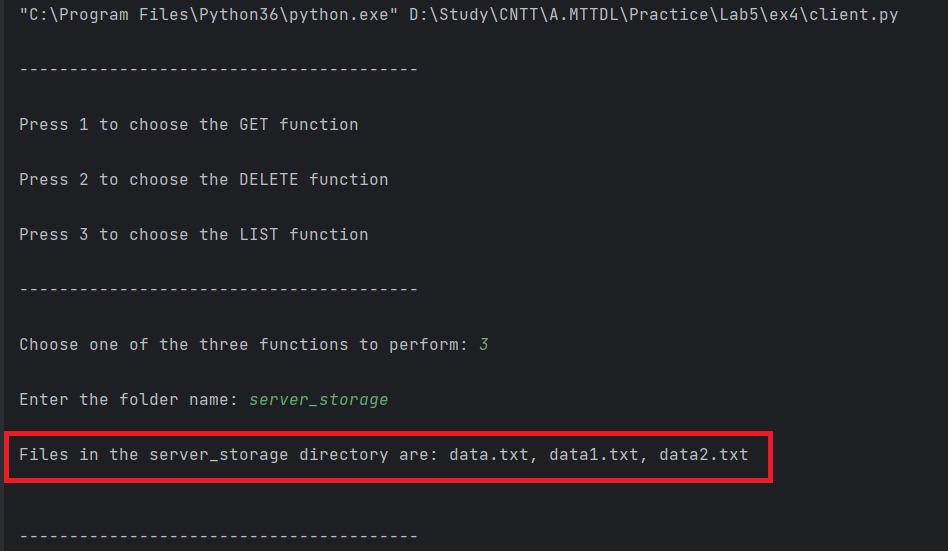
Hình 15 Kiểm tra chức năng GET (1)



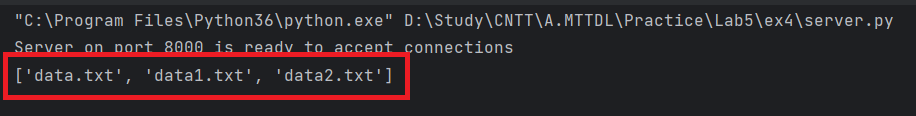
Hình 16 Kiểm tra chức năng GET (2)



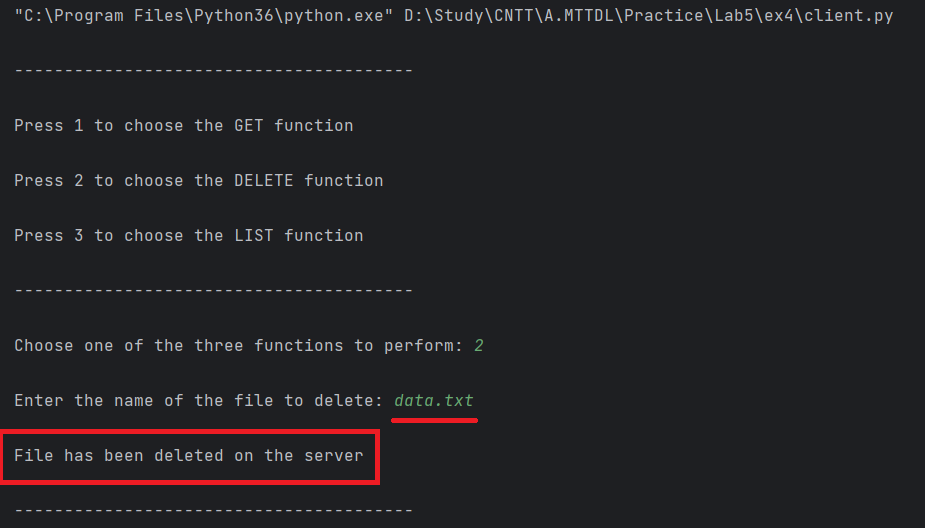
Hình 17 Kiểm tra chức năng GET (3)



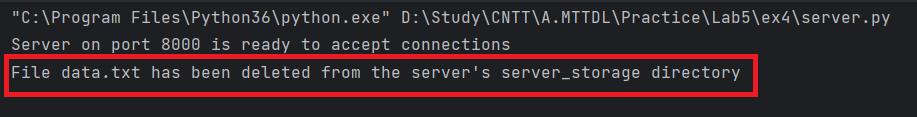
Hình 18 Kiểm tra chức năng LIST (1)



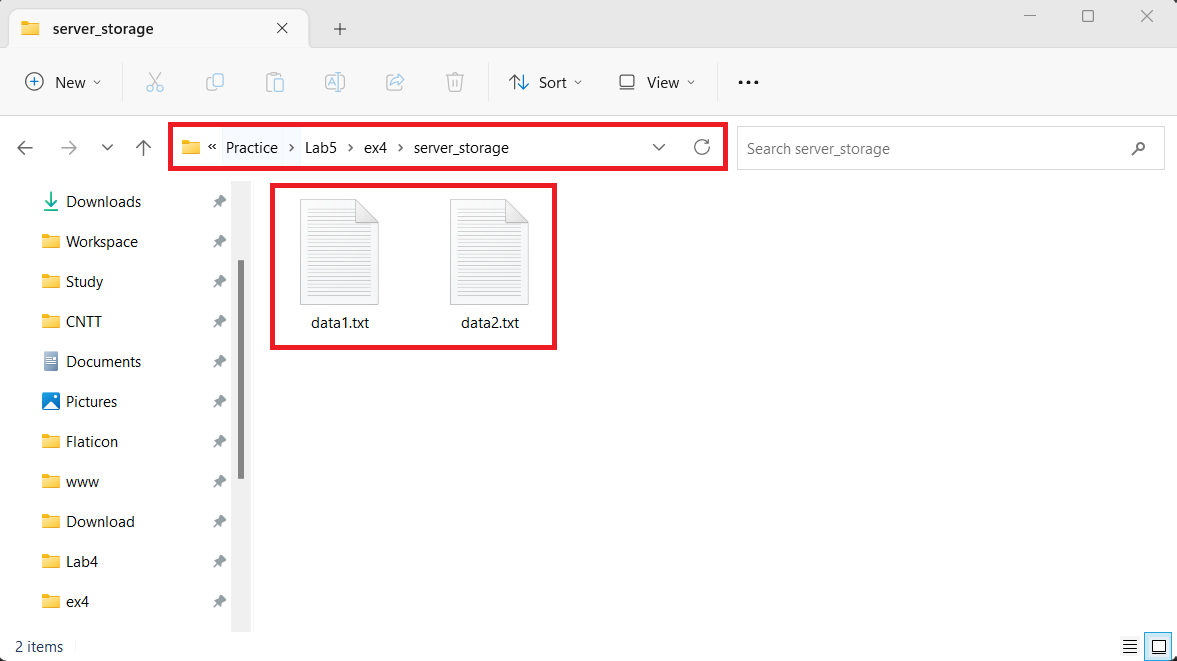
Hình 19 Kiểm tra chức năng LIST (2)



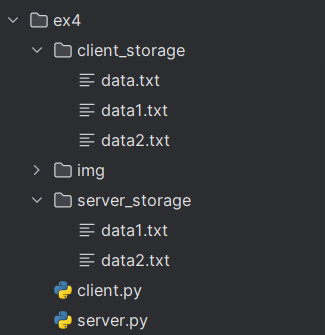
Hình 20 Kiểm tra chức năng DELETE (1)



Hình 21 Kiểm tra chức năng DELETE (2)



Hình 22 Kiểm tra chức năng DELETE (3)



Hình 23 Cấu trúc thư mục bài tập 4

--- Hết ---