```c
/*
 * Name: Joe Student
 * Course-Section:  CS-440
 * Assignment: Simplex server example
 * Date Due: 01/01/24
 * Collaborators: None
 * Resources: Textbook
 * Description: Example server to demonstrate how to use the BSD socket API
 */
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <netdb.h>

#include <err.h>
#include <stdio.h>
#include <string.h>
#include <unistd.h>

const int SERVER_PORT = 5432;
const int MAX_PENDING = 5;
const int MAX_LINE    = 256;

int
main(void)
{
    char buf[MAX_LINE];
    struct sockaddr_in sin;
    socklen_t new_len;
    int new_s, s;

    /*  build address data structure */
    memset((char*)&sin, 0, sizeof sin);
    sin.sin_family = AF_INET;
    sin.sin_addr.s_addr = INADDR_ANY;
    sin.sin_port = htons(SERVER_PORT);

    /* begin passive open */
    if ((s = socket(PF_INET, SOCK_STREAM, 0)) == -1)
        err(1, "unable to open socket");

    if ((bind(s, (struct sockaddr*)&sin, sizeof sin)) == -1)
        err(1, "unable to bind socket");

    if ((listen(s, MAX_PENDING)) == -1)
        err(1, "listen on socket failed");

    /* get next connection and print message */
    while (1) {
        new_len = sizeof sin;
        if ((new_s = accept(s, (struct sockaddr*)&sin, &new_len)) == -1) {
            close(s);
            err(1, "accept failed");
        }

        while (recv(new_s, buf, sizeof buf, 0) > 0)
            fputs(buf, stdout);

        close(new_s);
    }

    /* NOT REACHED */
    close(s);

    return 0;
}
```