```c
//BROADCAST CLIENT

#include <stdio.h>
#include <string.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <stdlib.h>
#include <unistd.h>

#define DEFAULT_PORT 12345
#define SERVER_IP "127.0.0.1" // Replace with actual server IP address if needed

int main(int argc, char *argv[]) {
    int client_socket;
    struct sockaddr_in server_address;
    char buf[1024];
    int port = DEFAULT_PORT;
    socklen_t server_address_len;
    int recv_size;

    if (argc > 2) {
        port = atoi(argv[2]);
    }

    // Create client socket
    client_socket = socket(AF_INET, SOCK_DGRAM, 0);
    if (client_socket == -1) {
        perror("Error: client socket not created");
        return 1;
    }

    // Fill in server's sockaddr_in
    server_address.sin_family = AF_INET;
    server_address.sin_port = htons(port);
    server_address.sin_addr.s_addr = inet_addr(SERVER_IP);

    // Send message to the server
    printf("Enter message to send: ");
    fgets(buf, sizeof(buf), stdin);
    sendto(client_socket, buf, strlen(buf), 0, (struct sockaddr *)&server_address,
sizeof(server_address));

    // Receive broadcasted messages from the server
    while (1) {
        server_address_len = sizeof(server_address);
        recv_size = recvfrom(client_socket, buf, sizeof(buf), 0, (struct sockaddr
*)&server_address, &server_address_len);
        if (recv_size == -1) {
            perror("Error: recvfrom call failed");
            return 1;
        }
        // Null-terminate the received data
        buf[recv_size] = '\0';

        // Print broadcasted message from the server
        printf("Broadcasted message from server: %s\n", buf);
    }

    // Close client socket
    close(client_socket);

    return 0;
}
```