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

//CLIENT

int main() {
    char* input;
    const char* server_name = "localhost";
    const int server_port = 8877;

    // build address data structure
    struct sockaddr_in server_address;
    memset(&server_address, 0, sizeof(server_address));
    server_address.sin_family = AF_INET;

    // convert address from string to binary form
    inet_pton(AF_INET, server_name, &server_address.sin_addr);

    // convert host byte order to network byte order
    server_address.sin_port = htons(server_port);

    // open a socket, print error if socket is not created correctly
    int sock;
    if ((sock = socket(PF_INET, SOCK_STREAM, 0)) < 0) {
        printf("could not create socket\n");
        return 1;
    }

    // establish connection, print error if connection is not made
    if (connect(sock, (struct sockaddr*)&server_address,
                sizeof(server_address)) < 0) {
        printf("could not connect to server\n");
        return 1;
    }

    int i = 0;
    int len = 0;
    int max = 100;
    char buff[max];
    char* pbuff = buff;

    // while still connected to the server
    while(1) {
```

```
        // type input to send to the server
        printf("Type something: ");
        input = readline("");
        send(sock, input, strlen(input), 0);

        i = recv(sock, pbuff, max, 0);
        pbuff += i;
        max -= i;
        len += i;

        buff[len] = '\0';
        printf("received: '%s'\n", buff);
    break;
    }

    // close the socket
    shutdown(sock,0);
    return 0;
}
```