```c
#include <stdio.h>              // standard input/output functions, printf() function
#include <unistd.h>             // close(socket) function
#include <stdlib.h>             // exit(0) function
#include <sys/socket.h>         // sockaddr structure
#include <arpa/inet.h>          // htons(), htonl() functions, inet_addr structure
#include <string.h>             // memset() function
#define BUFFER_SIZE 32

int main(){
/*..............................................................................................................
    1. Create a client socket using the socket(domain, type, protocol) function call
        int domain = AF_INET       ==> IPv4 communication domain
        int type = SOCK_STREAM     ==> sequenced, reliable, two-way, connection-based byte streams
        int protocol = IPPROTO_TCP ==> TCP
................................................................................................................
    int client_socket = socket(PF_INET, SOCK_STREAM, IPPROTO_TCP);
    if(client_socket < 0){
        printf("An error occurred while creating the client socket\n");
        exit(0);
    }
    printf("The client socket was successfully created.\n");
/*..............................................................................................................
    2. Address format:
        struct sockaddr_in{
            sa_family_t sin_family;   ==> address family: AF_INET
            in_port_t sin_port;       ==> port in network byte order
            struct in_addr sin_addr;  ==> internet address
        };
    Internet address:
        struct in_addr{
            uint32_t s_addr;          ==> address in network byte order
        };
................................................................................................................
    struct sockaddr_in serverAddress;                       //Note that this is the *server* address.
    memset(&serverAddress, 0, sizeof(serverAddress));
    serverAddress.sin_family = AF_INET;
    serverAddress.sin_port = htons(12345);
    serverAddress.sin_addr.s_addr = inet_addr("127.0.0.1");
    printf("Address assigned.\n");
/*..............................................................................................................
    3. Initiate a connection on a server socket using the connect() function call
        int connect(int sockfd, const struct sockaddr *addr, socklen_t addrlen);
        sockfd  ==> socket file descriptor
        addr    ==> pointer to the address structure created above
        addrlen ==> size of the addr structure
................................................................................................................
    int connection = connect(client_socket, (struct sockaddr *)&serverAddress, sizeof(serverAddress));
    if(connection < 0){
        printf("An error occurred while connecting client to server.\n");
        exit(0);
    }
    printf("Connection established successfully.\n");

    char message[BUFFER_SIZE];
    printf("Enter a real number... (max length = %d characters)\n", BUFFER_SIZE);
    fgets(message, BUFFER_SIZE, stdin);

    int bytes_sent = send(client_socket, message, strlen(message), 0);
    if(bytes_sent != strlen(message)){
        printf("An error occurred while sending the message to the server.\n");
        exit(0);
    }
    printf("Message sent successfully.\n");

    char receive_buffer[BUFFER_SIZE];
    int bytes_recd = recv(client_socket, receive_buffer, BUFFER_SIZE - 1, 0);
    if(bytes_recd < 0){
        printf("An error occurred while receiving the message from the server.\n");
        exit(0);
    }
    receive_buffer[bytes_recd] = '\0';
    printf("%s", receive_buffer);

    close(client_socket);
}
```