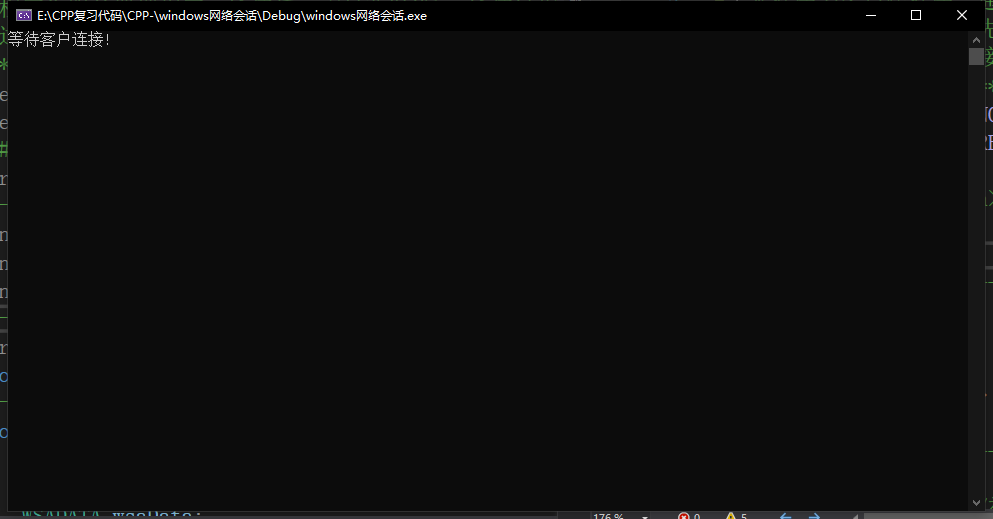
1. **运行、测试网络会话程序，记录测试结果，分析遇到的问题与解决的办法**

**1.运行、测试网络会话程序**

先让服务器端运行起来，等待客户端连接。



再运行客户端程序，输入服务端的ip地址(本机环回地址)和端口号port



至此为止，连接成功，开始进行通信！

客户端发送 “你好，吃了吗”，服务端成功接收到消息，并打印到控制台上。等待服务端回复。



服务端回复消息，等待客户端回复消息。



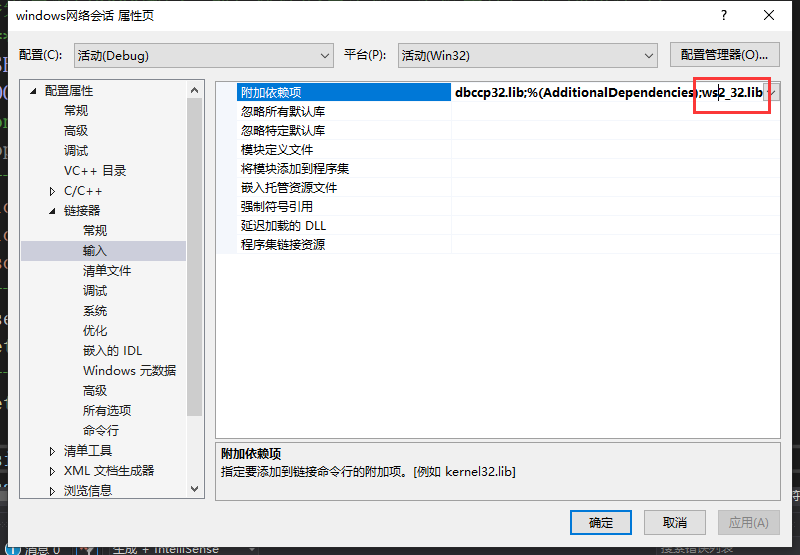
测试输入bye正常结束程序



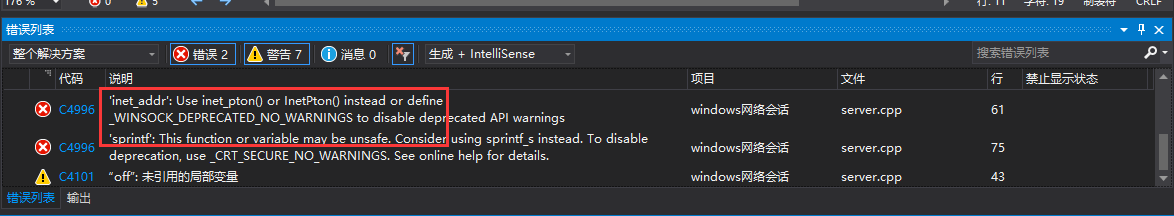
1. **分析遇到的问题与解决的办法**

(1)使用vs运行此程序时，需要先在项目中的链接器设置中，加上，附加依赖项ws2\_32.lib。

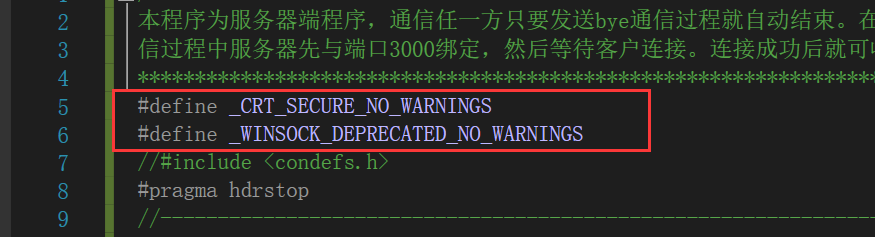
这样保证使用ws2\_32.lib，调用bind，accept等函数时能正确链接到库文件Windows Sockets 库文件。



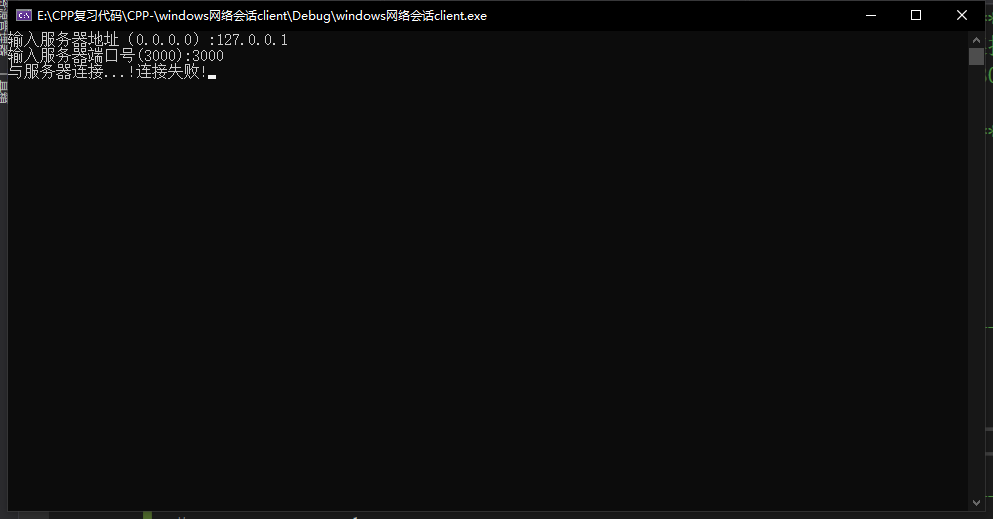
(2)在vs中使用inet\_addr和sprintf不安全。报错和解决方法如下。



前方加上定义即可。



1. 启动程序时，需让服务端先运行，等待客户端来连接。



1. **总结 TCP/IP 网络套接字编程的一般编程步骤。**

**对于windows服务端来说：**

1. 初始化 Windows Sockets 库：在程序开始时，调用 WSAStartup 函数初始化 Windows Sockets 动态连接库。

**2. socket创建套接字：使用socket函数创建一个套接字，指定协议族和套接字类型，填充sockaddr结构体。**

**3. bind绑定地址和端口：对于服务器端，使用 bind 函数将套接字与特定的 IP 地址和端口号进行绑定，以便监听客户端的连接请求。**

**4. listen服务器端监听连接：对于服务器端，使用listen函数开始监听连接请求，指定等待队列的长度，即允许同时等待连接的客户端数量。**

**5. accept接受连接请求：对于服务器端，使用accept函数接受客户端的连接请求，生成一个新的套接字(sersocket)来表示该连接，并返回连接的客户端地址信息(addr)。**

6. 与客户端通信：服务器端和客户端通过各自的套接字进行通信。使用 send`函数向对方发送数据，使用 recv`函数接收对方发送的数据。

7. 关闭套接字：通信结束后，使用closesocket函数关闭套接字。

8. 清理资源：在程序结束时，使用 WSACleanup函数卸载 Windows Sockets 库。

**对于客户端来说：**

1. 初始化 Windows Sockets 库：和服务端一样

2.Socket创建套接字：使用 socket函数创建一个套接字，**填充sockaddr结构体，客户端需要输入服务器的ip地址和端口号。**

3. connect连接服务器：使用 connect 函数连接到服务器，指定服务器的 IP 地址和端口号。

4. 与服务器通信：服务器端和客户端通过各自的套接字进行通信。使用 send函数向对方发送数据，使用 recv函数接收对方发送的数据。

5. 关闭套接字：通信结束后，使用 closesocket函数关闭套接字。

6. 清理资源：在程序结束时，使用 WSACleanup 函数卸载 Windows Sockets 库。

**总结：**

客户端和服务端在开始建立**tcp连接之前都需要**1.进行初始化 Windows Sockets 库，在结束通信之后需要1.关闭套接字：**通信结束后，**使用 closesocket函数关闭套接字。2.清理资源：在程序结束时，使用 WSACleanup 函数卸载 Windows Sockets 库。

服务端建立连接是调用函数顺序是：socket，bind，listen，accept。

客户端建立连接调用函数顺序是：socket，connect