**简单**

背景：使用QT实现Tcp协议的服务器和客户端

1. 查询QHostAdress和QAbstractSocket(QTcpSocket的基类)的帮助文档，实现当客户端连入上服务器后，在服务器端打印连上客户端的ip地址。

2. 实现当服务器关闭后，所有未断开连接的客户端也随之关闭。

1. 实现每当客户端断开连接后，服务器就打印“disconnected”。

**中等**

背景：使用QT实现Tcp协议的服务器和客户端

1. 当客户端连接上服务器后，服务器发送字样“welcome”至客户端，客户端把接受到的内容显示在界面上。

2. 当客户端连接上服务器后，服务器每隔3s发送“hello”信息至客户端，客户端把接受到的内容显示在界面上。

3. 有结构体data\_t的声明如下：

typedef struct data

｛

int a;

int b;

int ret;

｝data\_t;

要求客户端发送创建一个data\_t类型的结构体，设置a和b的值，并将该结构体发送给服务器，服务器接受后将ret的值设置未a和b的和，最后将修改过的结构体对象发送给客户端，客户端在界面显示服务器修改过后的ret的值。

**困难**

背景：使用QT实现Tcp协议的服务器和客户端

1. 客户端可以输入用户名和密码打包发送给服务器进行注册，注册的方式是服务器将用户的信息存入数据库，要求用户名不能重复。

2. 在第一题的基础上实现用户登陆的功能。

3. 在第二题的基础上实现相同用户名的用户无法同时登陆。