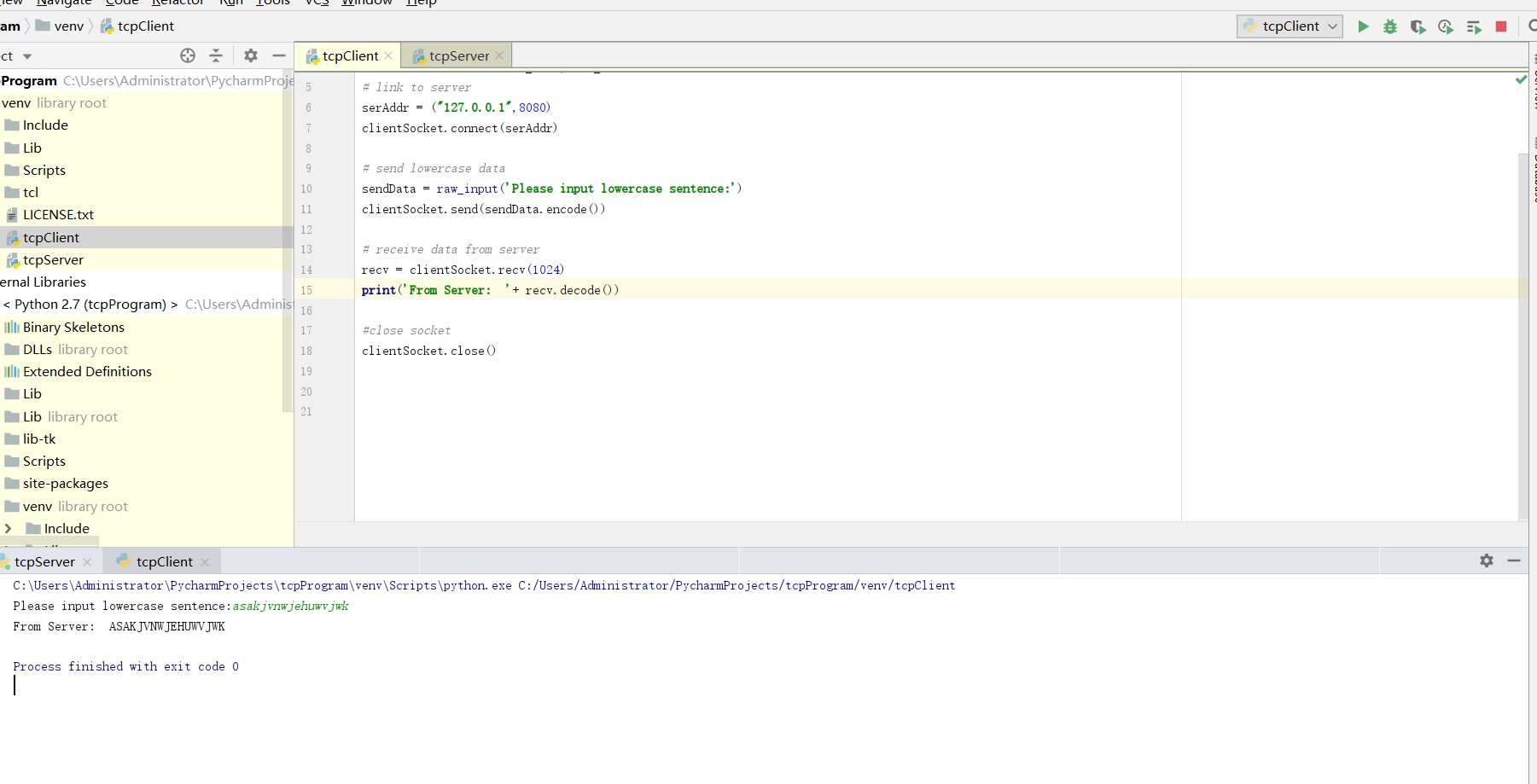
# Computer-Networking hw3

2016302580032 杨博东

1. TCP应用小程序，客户端发送小写字母序列，服务器将该序列转换为对应的大写字母序列，运行结果如下：



P5、答：UDP接收方对接收到的UDP报文段计算因特网检验和，并发现它与承载在检验和字段中的值相匹配，接收方并不能绝对确认没有出现过比特差错。因为检验和是三个16比特字的和的反码，验证时是验证三个16比特字的和与检验和相加是否能得到16位全为1的比特字。假如两个比特字中对应位在传输中都出现了比特差错，一个由1变为0，另一个由0变成1，则验证时并不能检验到差错，但实际上两个16比特字在传输中都出现了差错。

P7、答：考虑我们引入序号的初衷，发送分组需要序号是因为接收方需要根据发送分组序号判断该分组是否是已经确认接受的冗余分组。考虑ACK，发送方并不需要ACK序号去判断是否检测到冗余的ACK，冗余ACK对发送方来说是很显然的。当发送方接收到正确的ACK后它会发送下一个发送分组，一个冗余的ACK并不是发送方需要接收的，因而发送方会直接忽略冗余ACK。

P11 答：不能正常工作。发送方和接收方将进入死锁状态。考虑以下情形，发送方发送pkt0进入“等待ACK0状态”，等待接收方回复。接收方在“等待来自下层的0”状态，接收到来自发送方的pkt0，却不返回ACK0给接收方，然后进入“等待来自下层的1状态”。现在发送方在等待来自接收方的ACK0,接收方在等待来自发送方的发送分组1，因而进入了死锁状态。

P14 答:在只用NAK的协议之中，只有当分组X+1被接收到时，分组X的丢失才能被检测到。也就是说接收方接收X-1然后接收X+1，这时接收方意识到分组X丢失了。如果发送方只是偶尔发送数据，即分组X和分组X+1发送之间有很长时间间隔，则分组X将会在很长时间之后被检测到丢失，这时，该方案比使用ACK的方案更差。如果发送方发送大量分组，且端到端之间很少丢包，因而NAK只有很少被发送,ACK完全不被发送，因而比使用ACK的方案更好。

P44 答：（1） 6RTT

（2） 3MSS/RTT