P1:

a) The segments sent from A to S may have the source port number as 467 and destination port number as 23(Telnet).

• 23 is the port number of the Server S.

• 467 is the port number of the Client A.

b) The segments sent from B to S may have the source port number as 513 and destination port number as 23(Telnet). 23 is the port number of the Server S.

• 23 is the port number of the Server S.

• 513 is the port number of the Client B.

c) The segments sent from S to A may have the source port number as 23(Telnet) and destination port number as 467.

• 23 is the port number of the Server S.

• 467 is the port number of the Client A.

d) The segments sent from S to B may have the source port number as 23 and destination port number as 513.

• 23 is the port number of the Server S.

• 513 is the port number of the Client B.

e) Yes, it is possible that the source port number in the segments from A to S is the same as that from B to S even when A and B are different hosts. This is because IP addresses are also included in the segments to recognise the correct host.

f) No, the source ports would need to be different. It is not possible that the source port number in the segments from A to S is the same as that from B to S when they are the same hosts. The server to distinguish the hosts uses IP addresses.

P2:

Assume the IP addresses of the hosts A, B, and C are a, b, c, respectively. (Note that a, b, c are distinct.)

To host A: Source port =80, source IP address = b, destination port = 26145, destination IP address = a

To host C, left process: Source port =80, source IP address = b, dest port = 7532, destination IP address = c

To host C, right process: Source port =80, source IP address = b, dest port = 26145, destination IP address = c

P3:

One's complement = 1 1 0 1 0 0 0 1.

To detect errors, the receiver adds the four words (the three original words and the checksum). If the sum contains a zero, the receiver knows there has been an error. All one-bit errors will be detected, but two-bit errors can be undetected (e.g., if the last digit of the first word is converted to a 0 and the last digit of the second word is converted to a 1).

P7:

The protocol rtd3.0 is used to transfer data from sender to receiver.

If a sender transfer the packet to the receiver, then receiver will receive and send ACK(Acknowledgement) to the sender for conformation.

If sender received ACK then go to the next level.

In this process, needs sequence number to the sender for finding duplicate packets data or ACK data.

If the sender find any duplicate ACK, then ignore it. In this process ACK packets does not require sequence number.

So, ACK packets does not require sequence numbers.

P8:

To best answer this question, consider why needed sequence numbers in the first place. And saw that the sender needs sequence numbers so that the receiver can tell if a data packet is a duplicate of an already received data packet. In the case of ACKs, the sender does not need this info (i.e., a sequence number on an ACK) to tell detect a duplicate ACK. A duplicate ACK is obvious to the rdt3.0 receiver, since when it has received the original ACK it transitioned to the next state. The duplicate ACK is not the ACK that the sender needs and hence is ignored by the rdt3.0 sender.

P14:

If the sender sends the data infrequently (occasionally), then the NAK-only protocol is not preferred. It is good to use the protocol that uses ACKs.

The main disadvantage of NAK-only protocol is that it can detect that the packet has been lost only when the next packet is received by the receiver.

The NAK-only protocol realize the loss of packet after a long time when it receives the data packet with wrong sequence number as the sender sends the data packets occasionally.

When the receiver realize the packet loss, it sends NAK to the sender and the sender has to retransmit both the lost and next packet.

If the sender sends the data frequently, then the NAK-only protocol is preferred. The protocol that uses ACKs is not preferred as it require to send more number of acknowledgements.

When the sender sends data frequently and the data loss rate is very less, then a NAK-only protocol is preferable to a protocol that uses ACKs.

The receiver sends the NAK as it realizes the packet loss quickly as the data transmitted frequently.

Since the data loss is less, the number of NAK in NAK-only protocol is less when compared with protocol that uses ACKs.