Q1: Receiver Initiated Selective Repeat protocol:
Ans: Sample pseudocode and a flow chart given. Other correct answers are accepted.

Assumptions:
- 1 sender and 1 receiver
- Session between sender and receiver is initialized and permanent
- Sequence number is 1, 2, 3…..
- Sender and receiver can buffer w packets
- Sender re-transmits a packet when it times out without an ACK or when it receives a NACK

The steps are as follows:
1. Receiver REQuest packet (with next sequence number) from the sender; and starts a timer;
2. Sender send out packet labeled 1, 2, ..., upon receiving REQ with the same label from the receiver;
3. Receiver passes copy of packet correctly received to the network layer and REQ for the next packet
4. Receiver re-transmit REQ for same packet if timer times out or packet has errors.

Sender side Algorithm:
1. Initialize packet number (PN) to 0;
2. Wait until packet is received from network layer; assign PN to packet;
3. If REQ with NP received,
   if (NP > PN )
     PN ++;
     go to step 2;
   else
     transmit packet with label PN;
     go to step 3;

Receiver side Algorithm:
1. Initialize next packet (NP) to 0
2. Send REQ with label NP; start timer;
3. If error-free packet with label NP received within timeout
   pass it to network layer, increase NP by 1;
   go to step 2;
Otherwise, go to step 2;
At Receiver side:

Start

Polling the sender

Can new pkt fill in win?

Before T.O. pkt arrive?

NACKnum > Threshold? For that pkt

N

Send error message

N

NACK to sender

Is pkt correct?

Y

ACK to sender, release win. space, and send pkt to net

N

IS pkt Num Left of win?

Y

Win shift to right

N

Is data end?

Y

End
At Sender Side:

Start

- Can win accept new polling? (Y/N)
  - N: Is pkt ready? (Y/N)
    - Y: Send pkt
    - N: Are other pkt need to send? (Y/N)
      - Y: Before T.O. ACK arrives? (Y/N)
        - Y: Is it NACK? (Y/N)
          - Y: NACKnum > Threshold? for that pkt (Y/N)
            - Y: Send error message
            - N: Release win. space, and packet buffer
          - N: Win shift to right
        - N: Is it left of win? (Y/N)
  - Y: If new polling in buffer? (Y/N)
    - Y: Read polling
    - N: NACKnum > Threshold? for that pkt (Y/N)
      - Y: Send error message
      - N: Release win. space, and packet buffer

- Is it data end? (Y/N)
  - Y: End
  - N: Is it data end? (Y/N)

End
Q2: Efficiency of Stop & wait ARQ protocol and Selective Repeat ARQ protocol:

Ans:

Stop & Wait ARQ (SWP):

\[ S = P + A + 2(\tau + Q) \]

In successful transmission \( P_s = P_d \times P_a \)

Let \( X \) be the time needed to successfully send a pkt

\( T \) is timeout

So average transmission time for SWP:

\[ E\{X\} = P_sS + (1-P_s)(T+E\{X\}) \]

Solving \( E\{X\} \) we get

\[ E\{X\} = S + (1-P_s)T/P_s \]

Efficiency = \( \eta = \frac{\text{Packet_time}}{E\{X\}} = \frac{P_s}{S + (1-P_s)T} \)

Selective Repeat ARQ (SRP):

\[ E\{Y\} = P_s + (1-P_s)(1+E\{Y\}) \]

Thus, \( E\{Y\} = (P_s + (1 - P_s))/P_s = 1/P_s \)

\[ E\{X\} = P_sE\{Y\} \]

Efficiency = \( \eta = \frac{\text{Packet_time}}{E\{X\}} = P_s = P_dP_a \)
Q3: Selective Repeat with NACKs and Lazy Retransmit:
Ans:

\[ S = P + A + 2(\tau + Q) \]
Probability of success = \( P_s \) [Success of a packet] = \( P_s \) [Packet is correctly received; atleast one of \( W_T \) ACKs is correctly received] = \( P_d \ast [1 - (1 - P_a)^{W_T}] \)

![Diagram](image_url)

\[ 1, P_s \ast [1 - (1 - P_a)^{W_T}] \]

\[ 1, 1 - P_s \ast [1 - (1 - P_a)^{W_T}] \]

\[ E(Y) = P_s + (1 - P_s) (1 + E(Y)) \]
\[ E(Y) = \{P_d \ast [1 - (1 - P_a)^{W_T}] \}^{-1} \]
\[ E(X) = P \ast E(Y) \]

Efficiency = \( \eta = \) Packet_time / \( E(X) = P_s \)

Q 4: SWP sequence space is equal to set \{0,1\}
Ans:
Assume the sender has buffer \( W = 1 \) and the receiver has no buffer.
At time \( T \), Packet with label 1 is sent to the network after receiving an ACK from the receiver or the previous packet, which was labeled 0 since the sender buffer can hold only 1 packet. Thus, next time packet with label 0 can be sent if an ACK for packet 1 is received.

At the receiver side, the ACK for packet label 0 is already sent to the sender at time \( T \). When the receiver gets the packet with label 1 from sender he knows the next packet is 0.

Q 5: Bit stuffing strategy for flag 10011001
Ans:
To avoid an occurrence of the pattern 10011001 in the payload:
The sender inserts a “0” in the middle of the flag “10011001” so that it become “100101001”. The receiver delete the “0” if “100101001” occurs in the payload so that it becomes the correct bit sequence “10011001”