1. The following data fragment occurs in the middle of a data stream for which the byte-stuffing algorithm described in the text is used:

   A B ESC C ESC FLAG FLAG D

   What is the output after stuffing?

   Answer. The output is: A B ESC ESC C ESC ESC ESC FLAG ESC FLAG D

2. You receive the following data fragment:

   0110 0111 1100 1111 0111 1101.

   The protocol uses bit stuffing. Show the data after destuffing.

   Answer. The data after destuffing is: 0110 0111 110 1111 0111 111

3. An 8-bit byte with binary value 10101111 is to be encoded using an even-parity Hamming code. What is the binary value after encoding?

   Elaborate the detailed calculation process.

   - 解.

     3.a. Determine the number of parity bits (r):

     The data length $m = 8$. To satisfy the inequality $2^r \geq m + r + 1$.
     - If $r = 3$, $2^3 = 8 < 8 + 3 + 1 = 12$ (Not enough).
     - If $r = 4$, $2^4 = 16 \geq 8 + 4 + 1 = 13$ (Enough).

     So, 4 parity bits are needed. The total length of the codeword is 12 bits.

     3.b. Determine bit positions:

     Parity bits are located at positions that are powers of 2 ($1, 2, 4, 8$). Data bits fill the remaining positions ($3, 5, 6, 7, 9, 10, 11, 12$).

     The data is `10101111`. Mapping them to positions:

Pos	1	2	3	4	5	6	7	8	9	10	11	12
Type	P_1	P_2	D_1	P_4	D_2	D_3	D_4	P_8	D_5	D_6	D_7	D_8
Val	?	?	1	?	0	1	0	?	1	1	1	1

     3.c. Calculate Parity Bits (Even Parity):
     - P_1 checks positions 1, 3, 5, 7, 9, 11.

     Data bits: $1, 0, 0, 1, 1$. Sum of 1s is 3 (Odd): $P_1 = 1$.
     - P_2 checks positions 2, 3, 6, 7, 10, 11.

     Data bits: $1, 1, 0, 1, 1$. Sum of 1s is 4 (Even). $P_2 = 0$.
     - P_4 checks positions 4, 5, 6, 7, 12.

     Data bits: $0, 1, 0, 1$. Sum of 1s is 2 (Even). $P_4 = 0$.
     - P_8 checks positions 8, 9, 10, 11, 12.

     Data bits: $1, 1, 1, 1$. Sum of 1s is 4 (Even). $P_8 = 0$.

     3.d. Final Codeword: Combining the parity bits and data bits: 101001001111

4. A 12-bit odd-parity Hamming code whose hexadecimal value is 0xB4D arrives at a receiver. What was the original value in hexadecimal? Assume that not more than 1 bit is in error.

解. 0xB4D in binary is 1011 0100 1101.

Pos	1	2	3	4	5	6	7	8	9	10	11	12
Type	P_1	P_2	D_1	P_4	D_2	D_3	D_4	P_8	D_5	D_6	D_7	D_8
Val	1	0	1	1	0	1	0	0	1	1	0	1

- Check for P_1:

  Positions 1, 3, 5, 7, 9, 11: 1, 1, 0, 0, 1, 0. Sum of 1s = 3 (Odd). P_1 is correct.
- Check for P_2:

  Positions 2, 3, 6, 7, 10, 11: 0, 1, 1, 0, 1, 0. Sum of 1s = 3 (Odd). P_2 is correct.
- Check for P_4:

  Positions 4, 5, 6, 7, 12: 0, 1, 0, 1, 0. Sum of 1s = 2 (Odd). P_4 is correct.
- Check for P_8:

  Positions 8, 9, 10, 11, 12: 1, 1, 1, 1, 0. Sum of 1s = 4 (Odd). P_8 is correct.

Hence there is no error detected. Just extract the data bits: $D_1, D_2, D_3, D_4, D_5, D_6, D_7, D_8 = 10101101$

Convert 10101101 to hexadecimal: 0xAD.

5. Suppose that a message 1001 1100 1010 0011 is transmitted using the Internet Checksum (4-bit word). What’s the value of the checksum?

`Elaborate the detailed calculation process.`

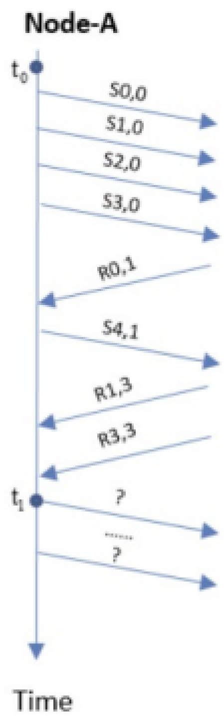
解.
- $1001 + 1100 = 10101 \Rightarrow 0101 + 1 = 0110$
- $0110 + 1010 = 10000 \Rightarrow 0000 + 1 = 0001$
- $0001 + 0011 = 0100$
- One’s complement of 0100 is 1011, which is the checksum.

6. A stop-and-wait protocol achieves 25% bandwidth efficiency using 900-bit frames over a channel with a one-way propagation delay of 50 msec. What is the bandwidth of this channel in bits per second?

解 . $25\% = \frac{\text{Transmission delay}}{\text{Propagation delay} *2\text{+ Transmission delay}} \Rightarrow \text{Transmission Delay} = \frac{1}{30}s = \frac{900 \text{ bits}}{\text{bandwidth}} \Rightarrow \text{bandwidth} = 27k \text{ bps}$

7. Both Node A and B use the Go-Back-N protocol for continuous two-way data transmission, both parties use piggyback ACK, and the frame length is 2000 bits. Sx,y and Rx,y respectively denote the data frames sent by A and B, where x is the sequence number for the outgoing frame, and y is the acknowledgment number which is the number for the next frame to receive. The field length of SEQ numbers and ACK numbers of data frames is 4 bits. The data transmission rate of the channel is 100 Mbps and RTT is 0.96 ms. The figure shows one scenario in which the Node-

A sends and receives data frames, at the initial time t0 both sequence number and acknowledgment sequence number of Node-A is 0, and at t1 Node-A has enough data to be transmitted.



7.a. In the figure, from t0 to t1, Node-A can confirm that how many frames Node-B has received correctly?

Answer: 3

Which ones are the frames received correctly? (Denote them as Sx,y)

	x	y
First Frame: S	0	0
Last Frame: S	2	0

7.b. In the figure, from t1, if no timeout occurred and no more data frame is received from Node-B, how many data frames can Node A send?

Answer: 13

What are the first frame and the last frame (Denote them as Sx,y)?

	x	y
First Frame: S	5	2
Last Frame: S	1	2