<u>Stuffing :-</u> Flag byte is added in the beginning and ending of frame to separate one frame from another.

Byte Stuffing

Bit Stuffing

Byte Stuffing:

As we know that flag byte is added in the beginning and ending of frame but if the data part contains the flag then the receiver wrongly predict that it is the end of frame .

To avoid this **in *Byte stuffing one extra*** **(ESC)** ***byte is added whenever there is a flag in the text.***

If ESC is there in text itself then in that case also we have to add one extra **(ESC)** ***byte.***

Altogether one extra **(ESC)** ***byte is added whenever there is a flag or ESC in the text.***

Bit stuffing:

Fix Flag is used:- 01111110

As we know that flag byte is added in the beginning and ending of frame but if the data part contains the flag then the receiver wrongly predict that it is the end of frame .

To avoid this ***In bit stuffing, if a 0 and five consecutive 1 bits are encountered, an extra 0 is added.*** The extra bit is added after one 0 followed by five 1s regardless of the value of the next bit.

This means that if the flag like pattern **01111110** appears in the data, it will change to 011111010 (stuffed) and is not mistaken as a flag by the receiver.

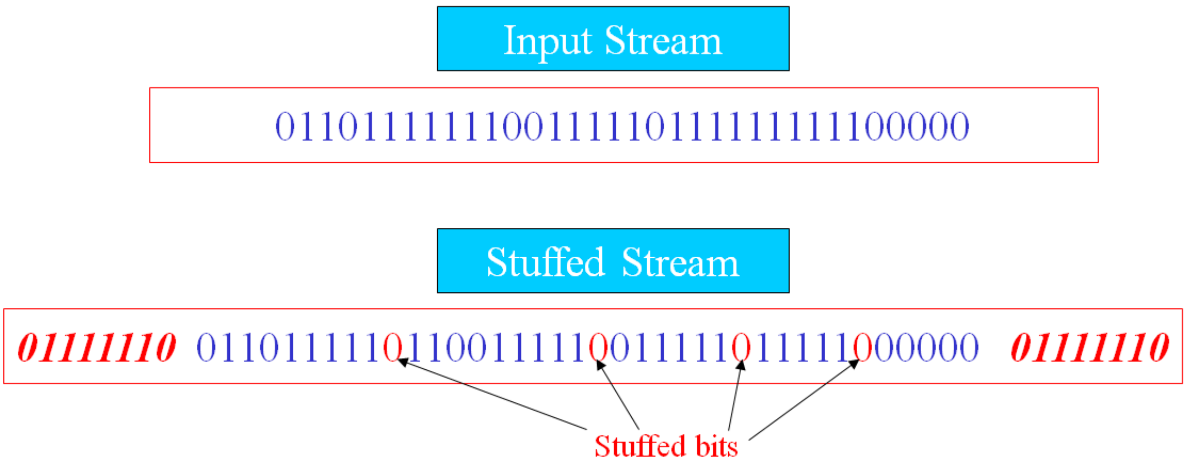
Q. The following data fragment occurs in the middle of a data stream for which the **byte-stuffing** algorithm described in the text is used: A B ESC C ESC FLAG FLAG D . What is the output after stuffing?

Answer: **FLAG** A B ESC ESC C ESC ESC ESC FLAG ESC FLAG D **FLAG**

Q. A bit string, , needs to be transmitted at the data link layer. What is the string 0111101111101111110actually transmitted after **bit stuffing**?

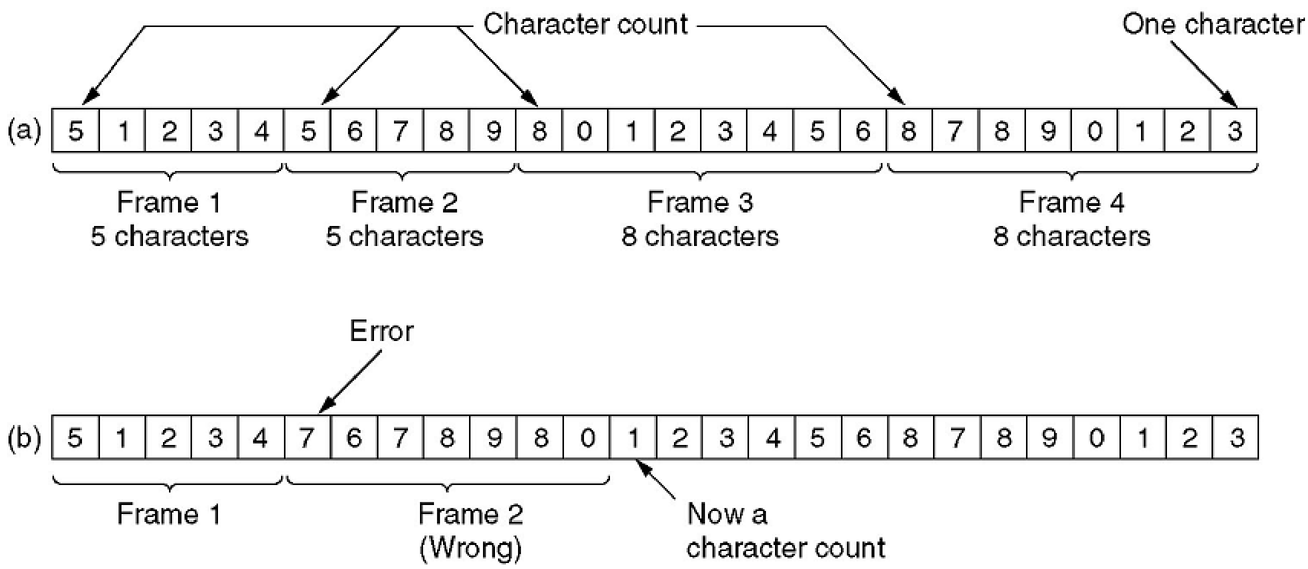
Answer: **01111110** 011110111110011111010 **01111110**

Q. Apply bit stuffing



Framing By Chracter Count:

Count is added in the beginning of every frame .



Hamming Code (Single error correction)

$2^r - 1 >= r + k$

r=redundant bits(check bits)
k=data bits

Thus, eight data bits require four check bits..

The data and check bits are arranged into a12-bit word The bit positions are numbered from 1 to 12.Those bit positions whose position numbers are powers of 2 are designated as check bits. Each check bit operates on every data bit which contains1in the same position as the position of 1 on check bit.

$C1 = D1 \oplus D2 \oplus D4 \oplus D5 \oplus D7$

$C2 = D1 \oplus D3 \oplus D4 \oplus D6 \oplus D7$

$C4 = D2 \oplus D3 \oplus D4 \oplus D8$

$C8 = D5 \oplus D6 \oplus D7 \oplus D8$

Bit position	12	11	10	9	8	7	6	5	4	3	2	1
Position number	1100	1011	1010	1001	1000	0111	0110	0101	0100	0011	0010	0001
Data bit	D8	D7	D6	D5		D4	D3	D2		D1		
Check bit					C8				C4		C2	C1

Let the data to be send is : 00111001

$C1 = 1 \oplus 0 \oplus 1 \oplus 1 \oplus 0 = 1$

$C2 = 1 \oplus 0 \oplus 1 \oplus 1 \oplus 0 = 1$

$C4 = 0 \oplus 0 \oplus 1 \oplus 0 = 1$

$C8 = 1 \oplus 1 \oplus 0 \oplus 0 = 0$

Suppose now that data bit 3 sustains an error and is changed from 0 to 1. When the check bits are recalculated, we have

$$C1 = 1 \oplus 0 \oplus 1 \oplus 1 \oplus 0 = 1$$

$$C2 = 1 \oplus 1 \oplus 1 \oplus 1 \oplus 0 = 0$$

$$C4 = 0 \oplus 1 \oplus 1 \oplus 0 = 0$$

$$C8 = 1 \oplus 1 \oplus 0 \oplus 0 = 0$$

When the new check bits are compared with the old check bits, the syndrome word is formed:

$$\begin{array}{ccccc} & C8 & C4 & C2 & C1 \\ & 0 & 1 & 1 & 1 \\ \oplus & 0 & 0 & 0 & 1 \\ \hline & 0 & 1 & 1 & 0 \end{array}$$

The result is 0110, indicating that bit position 6, which contains data bit 3, is in error.