ISE1 C2 Test

Q.1 → 0011111111010100011111001 - Given message

→ 01111110 - Flog

→ To differentiate message from flag, we are searchin 0 followed by consecutive 1 fine times. If we find then we are inserting / stuffing extra '0'.

→ By applying above rule the message becomes

001111101110101.000111110001 ← message after stuf

→ message to.be sent (codeword) to be sent

$\underbrace{01111110}_{\text{flag}}$ 001111101110101000111110010 $\underbrace{1111110}_{\text{flag}}$

After destuffing shows original message

Q.2 → welcometobeautifulcityofmaharashtra - Given message

→ After tokenization we get

welcome	→ 7	length of each frame
to	→ 2	
beautiful	→ 9	
city	→ 4	
of	→ 2	
maharashtra	→ 11	

→ After applying character count technique; message become

↓c								↓cc			↓cc															↓cc			↓cc		
8	w	e	l	c	o	m	e	3	t	o	10	b	e	a	u	t	i	f	u	l	s	c	i	t	y	3	o	f	12	m	a

r	a	s	h	t	r	a

← message to be sent

→ At receiver the receiver easily identify the frames of va size using character count associated with frame & extract the original data

→ welcome to beautiful city of maharashtra.

→ 0110 - Given Dataword

→ Number of redundant (extra) bits needs to be added using formula $2^r >= d+r+1$ where d=4 (message or

→ Here r = 3 satisfies the formula

ie the size of codeword is = d+r = 4+3 = 7

→ Let us identify position of parity/redundant bits using formula 2^m where m = 0,1,2,3 .....

→

D4	D3	D2	P4	D1	P2	P1
0	1	1	P4	0	P2	P1
7	6	5	4	3	2	1

→ Letus identify the parity bits

P1 = 1 3 5 7 = 0010 ⇒ 1

P2 = 2 3 6 7 = 0010 ⇒ 1

P3 = 4 5 6 7 = 0110 ⇒ 0

→ Final codeword after adding parity bits

0	1	1	0	0	1	1
7	6	5	4	3	2	1

→ during transmission no error is introduced as per problem statement

→ Now let us verify that no error is detected

→ By recalculating parity bits

P1 = 1 3 5 7 = Φ010 ⇒ 0

P2 = 2 3 6 7 = 1010 ⇒ 0

P3 = 4 5 6 7 = 0110 ⇒ 0

→ Decimal equivalent of $(000)_{10}$ is not pointing to any positions in codeword. It means no error is detec

→ Given message or dataword 11010110 11

→ x^4+x+1 - Given generator polynomial

→ Let us extract the divisor from $G(x) = x^4+x+1$

→ Expanded form of polynomial is
$1 \cdot x^4 + 0 \cdot x^3 + 0 \cdot x^2 + 1 \cdot x^1 + 1 \cdot x^0$ which is encoded 10011

→ No. of extra bits needs to be added with dataword is equal to highest degree of $G(x)$ or size of divisor minus 1

→ So, dividend is 11010110110000

→ Let us divide

```
10011 | 1 1 0 1 0 1 1 0 1 1 0 0 0 0
        1 0 0 1 1
        0 1 0 0 1 1
          1 0 0 1 1
          0 0 0 0 0 1 0 1 1 0
                    1 0 0 1 1
                    0 0 1 0 1 0 0
                        1 0 0 1 1
                        0 0 1 1 1 0
```

→ Final Codeword is 1,1 0 1 0 1 1 0 1 1,1 1 1 0

→ As per the problem statement 3rd bit is changed during transmission

→ codeword received is 11010110111010

gain perform division using G(x)

```
10011 | 11010110111010
        10011
        010011
         10011
         00000 10111
               10011
               0010001
                 10011
                 000100.
```

As remainder is not zero, error is introduced during transmission

Q.5 Let us divide the data into 4 segments of size 8

10011001 11100010 00100100 10000100

```
S1        10011001
S2        11100010
        ①01111011
                11
          01111100
S3        00100100
          10100000
S4        10000100
        ①00100100
                 1
Sum       00100101
Checksum  11011010
```

```
  10011001        S1
  11100010        S2
①01111011
        11
  01111100
  00100100        S3
  10100000
  10000100        S4
  00100100
         1
  00100101        sum
  11011010        check
  11111111
```

After Inversion 00000000

No error is introduced during transmission