Question Number 1

a. SNR= 100
Bandwidth =400 KHz
C= B* $\log_2$ (1+SNR)
By putting the values:
C=400*10^3 * $\log_2$ (1+100) = 26.57 Mbps

- 0.5 marks if calculation mistake.
-0.5 marks if error in units.
-1 marks if only formula is correct.
-2 marks if formula used is incorrect.

b. Maximum data rate= 10Mbps
No. of frames sent in one sec=12000/60= 200 f/s
Size of each frame= 10,000 bits
Throughput= data sent in one second= 10,000*200 = 20000000bps =20 Mbps

-0.5 marks if calculation mistake.
-0.5 marks if error in units.
-1 marks if only formula is correct.
-2 marks if formula used is incorrect.

c. 00111110110001111011110001111101

-1 if only single stuffed bit is correct.
-2 marks if both the stuffed bits are incorrect.

d. i. 10 Kbps *5 + 2* 5Kbps = 60 kbps

-1 marks if data rate is incorrect.

ii. In synchronous TDM, each sender is assigned a fixed time slot in a TDM frame. The frame size is determined by the total number of senders and the time required by each sender to transmit their data.

Given that there are 7 senders and each sender wants to transmit 5 KB of data, the total amount of data to be transmitted in one frame is:
Total data = 7 x 5 KB = 35 KB = 280 Kb

In multiplexing, we are using 1 slot for each sender having data rate 10 Kbps and 2 senders with 5kbps are multiplexed into a single slot. **(2 marks)**

Since no information is given regarding the slot size so by default slot size is taken as 1 bit.

Data carried by each frame= 6 bits from each slot+1 synchronization frame
Total number of frames= =Total data/data bits of frame= 280 Kb/6 = 46.66 = 47K frames **(1 mark)**

This implies that the total number of framing bits will be equal to 47Kb (47K * 1b) (given in the question that synchronization bits are added for each frame)

Total data to be sent = 280 Kb+ 47Kb= 327 Kb **(1 mark)**

Efficiency = Data size/ Total data size = 280 Kb/ 327 Kb = 85.62% **(1 mark)**

OR

ii. In synchronous TDM, each sender is assigned a fixed time slot in a TDM frame. The frame size is determined by the total number of senders and the time required by each sender to transmit their data.

Given that there are 7 senders and each sender wants to transmit 5 KB of data, the total amount of data to be transmitted in one frame is:
Total data = 7 x 5 KB = 35 KB = 280 Kb

In multiplexing, we are 2 slots for each sender having data rate 10 Kbps and 1 slot for senders with 5kbps. **(2 mark)**

Since no information is given regarding the slot size so by default slot size is taken as 1 bit.

Data carried by each frame= 12 bits from each slot+1 synchronization frame= 13 bits
Total number of frames= =Total data/data bits of frame= 280 Kb/12 = 23.33 = 24K frames **(1 mark)**

This implies that the total number of framing bits will be equal to 24Kb (24K * 1b) (given in the question that synchronization bits are added for each frame)

Total data to be sent = 280 Kb+ 24Kb= 304 Kb **(1 mark)**

Efficiency = Data size/ Total data size = 280 Kb/ 304 Kb = 92.10% **(1 mark)**

Question Number 2

a. M = 1010001101
Divisor polynomial bit= 110101
Bits to be appended to message= (divisor polynomial bits – 1) = 5
Append 5 zeros to message bits, modified message: 101000110100000

```
110101 | 1 0 1 0 0 0 1 1 0 1 0 0 0 0 0
         1 1 0 1 0 1
         -----------
           1 1 1 0 1 1
           1 1 0 1 0 1
           -----------
               1 1 1 0 1 0
               1 1 0 1 0 1
               -----------
EXOR               1 1 1 1 1 0
                   1 1 0 1 0 1
                   -----------
                       1 0 1 1 0 0
                       1 1 0 1 0 1
                       -----------
                         1 1 0 0 1 0
                         1 1 0 1 0 1
                         -----------
                             0 1 1 1 0
```

$M' = 1\,0\,1\,0\,0\,0\,1\,1\,0\,1\,\boxed{0\,1\,1\,1\,0}$

-1 mark if final message is incorrect. (If there is error in the final message due to single incorrect EXOR operation).
-1 mark for each incorrect EXOR operation.
-2 marks if number of bits appended are incorrect.
- 6 marks if bits are not appended.

b. If two 16-bit words are swapped in the bit stream during transmission, the checksum will not detect this error.

The checksum algorithm is designed to detect errors that occur within a single 16-bit word. It works by adding up all the 16-bit words in the data, and then taking the one's complement of the sum. The result is then appended to the end of the data and sent along with it.

When the receiver receives the data, it performs the same checksum algorithm on the data (including the appended checksum), and compares the result with the checksum that was sent. If the two checksums match, the receiver assumes that the data was transmitted without errors. If the checksums don't match, the receiver knows that an error has occurred.

Now, let's consider the scenario where two 16-bit words are swapped during transmission. In this case, the data that was received by the receiver will be different from the data that was transmitted by the sender. However, the sum of the 16-bit words will remain the same, as only the order of the words has been changed. This means that the calculated checksum will also be the same as the checksum that was transmitted by the sender.

As a result, the receiver will not detect the error and will assume that the data is error-free. This is because the checksum algorithm is designed to detect errors that occur within a single 16-bit word, but not errors that occur due to the order of the words being swapped.

-2 marks if it is mentioned that checksum may or may not detect.
-2 marks if it is mentioned that checksum will not detect.
-1 mark if reason is incorrect

Question Number 3

a. Selective Repeat ARQ (Automatic Repeat Request) is a type of error control mechanism used in data communication systems. It allows the receiver to selectively request retransmission of only those packets that are lost or corrupted during transmission, rather than asking for the entire set of packets to be retransmitted.
In selective repeat ARQ, each packet is assigned a sequence number, which is used by the receiver to acknowledge the successful reception of the packet. The sender maintains a window of unacknowledged packets, which can be transmitted without waiting for an acknowledgement.

Now, let's consider a selective repeat ARQ scheme with 3-bit sequence numbers. This means that the sequence numbers range from 0 to 7 (2^3 = 8 possible values).
The sender maintains a window of unacknowledged packets, and the size of this window determines the maximum number of packets that can be transmitted without waiting for an acknowledgement. The window size is limited by the number of available sequence numbers.
In this case, we have 8 possible sequence numbers (0 to 7), but one of these sequence numbers (e.g., 0) is used for the initial transmission of packets and is not available for retransmission. This leaves us with only 7 available sequence numbers for retransmission. If the window size is larger than the number of available sequence numbers, there is a risk that the same sequence number may be used for two different

packets, which can result in errors in the received data. For example, suppose we have a window size of 5 and the following packets are sent:

Packet 0, 1, 2, 3, 4 are sent and are all received successfully.
Packets 0 and 1 are acknowledged, but the acknowledgement for packet 2 is lost.
The sender retransmits packet 2, but the acknowledgement for packet 2 is again lost.
The sender then transmits packets 5, 6 and 7, which are all received successfully.
The receiver now sends an acknowledgement for packet 3, which includes sequence numbers 0, 1, 2 and 4.
The sender incorrectly assumes that packets 0, 1 and 4 have been acknowledged and removes them from the window.
However, packet 2 has not been acknowledged and is still in the window.
If the sender transmits a new packet with sequence number 2, it will overwrite the unacknowledged packet 2, resulting in an error in the received data.

This problem can be avoided by limiting the window size to the number of available sequence numbers for retransmission, which in this case is 7-1=6. Therefore, the maximum window size for a selective repeat ARQ scheme with 3-bit sequence numbers is 6, which means that at most 6 unacknowledged packets can be sent at a time without waiting for an acknowledgement. A window size of 4 is also safe, as it is less than the maximum of 6.

****************************Example may vary*********************************

-2 marks if example is missing.
-2 marks if example is partially correct.
- 4 marks if it is not mentioned that frames are erroneously accepted.

b. For calculating efficiency: **5 marks**
Window size= 6 frames
Size of data frame = 64 bytes
Size of acknowledgement frame = 8 bytes
Transmission rate= 100Mbps
Length of the link= 10km
Propagation speed=$2*10^8$ m/sec **1 mark**
Propagation delay =$(10*1000)/(2*10^8)$ m/sec= $50 *10^{-6}$ sec
Round Trip Time (RTT)= 2* Propagation delay= $100*10^{-6}$ sec
Transmission Time of sender= Length of Frame/ Bandwidth = $64*8/ 100 * 10^6$ = $5.12 * 10^{-6}$ sec **1 mark**
Transmission Time of receiver= Length of acknowledgement/ Bandwidth = $8*8/ 100 * 10^6 = 0.64 * 10^{-6}$ sec

Total time= RTT+ Transmission Time of sender + Transmission Time of receiver = $105.76 * 10^{-6}$ sec **1 mark**

Efficiency =((6* Transmission Time of sender)/ Total time) *100= 30.72/105.76 29.04% **2 marks (-1 marks if calculation mistake)**

For maximum efficiency: **3 marks**
(N* Transmission Time of sender)/ Total time >=1 **2 marks**
N =105.76/5.12 =20.65
N=21 **1 mark**

OR

Window size= 6 frames
Size of data frame = 64 bytes
Size of acknowledgement frame = 8 bytes
Transmission rate= 100Mbps
Length of the link= 10km
Propagation speed=$2*10^8$ m/sec
Propagation delay =$(10*1000)/(2*10^8)$ m/sec= $50 *10^{-6}$ sec **1 mark**
Round Trip Time (RTT)= 2* Propagation delay= $100*10^{-6}$ sec
Transmission Time of sender= Length of Frame/ Bandwidth = $64*8/ 100 * 10^6$= $5.12 * 10^{-6}$ sec **1 mark**

Efficiency =N/ 1+2a **1 mark**
a= 9.76 **1 mark**
Efficiency= (6/ 1+2*9.76)*100 =29.22% **1 mark**

For efficiency=100% **3 marks**
N/ (1+2a) =1 **2 marks**
N= 1+2a
N =20.52
N=21 **1 mark**