KING SAUD UNIVERSITY COLLEGE OF COMPUTER AND INFORMATION SCIENCES COMPUTER SCIENCE DEPARTMENT		
CSC 329: Computer Network	**Tutorial 5 and 6**	**2nd Semester 1441**
Name:	**Student ID:**	

Part1: Multiple-Choice Questions

1) The Hamming code is a method of _____

a. Error detection
b. Error correction
c. Error encapsulation
d. (a)and(b)

2) A receiver receives the code 11001100111. When it uses the Hamming encoding algorithm, the result is 0101. Which bit is in error?

a. 1
b. 3
c. 5
d. none of the above

3) Which error detection method involves polynomials?

a. Simple parity check
b. Two-dimensional parity check
c. CRC
d. Checksum

4) In CRC, if the data unit is 111111 and the divisor 1010, what is the dividend at the transmitter?

a.111111000
b.1111110000
c.111111
d.1111111010

5) In CRC there is no error if the remainder at the receiver is _____

a. Equal to the remainder at the sender
b. Zero
c. Nonzero
d. The quotient at the sender

6) Which error detection method can detect a burst error?

a. simple parity check
b. Two-dimensional parity check
c. CRC
d. (b) and (c)

7) Which error detection method uses ones complement arithmetic?

a. Simple parity check
b. Two-dimensional parity check
c. CRC
d. Checksum

8) Flow control is needed to prevent _____ .

a. Bit errors
b. Overflow of the sender buffer
c. Overflow of the receiver buffer
d. Collision between sender and receiver

9) A timer is set when _____ is (are) sent out.

a. A data frame
b. An ACK
c. A NAK
d. All the above

Part2: Exercises

1) Hamming code is a technique that is used to achieve forward error control. This allows a receiver to correct any single error, if any, in the received message. If the transmitted character is **01001010**, generate the Hamming codeword.

Implementing the Hamming codeword

We have the following redundant bits: **r_1,r_2,r_4,r_8**

12	11	10	9	8	7	6	5	4	3	2	1
0	1	0	0	r_8	1	0	1	r_4	0	r_2	r_1

Each one of them is responsible for the following bits:

r_1: 1, 3, 5, 7, 9, 11
r_2: 2, 3, 6, 7, 10, 11
r_4: 4, 5, 6, 7,12
r_8: 8, 9, 10, 11,12

0	1	0	0	r_8	1	0	1	r_4	0	r_2	r_1
0	1	0	0	r_8	1	0	1	r_4	0	r_2	**1**
0	1	0	0	r_8	1	0	1	r_4	0	**0**	**1**
0	1	0	0	r_8	1	0	1	**0**	0	**0**	**1**
0	1	0	0	**1**	1	0	1	**0**	0	**0**	**1**

2) Assume that the **received** Hamming codeword is:

11	10	9	8	7	6	5	4	3	2	1
1	0	1	0	0	1	1	0	0	1	0

Deduce the original data word from the above codeword after correction if any error.

To check the received data we will calculate the party bit for each combination
r1 : 1,3,5,7,9,11 (3) → 1
r2 : 2,3,6,7,10,11 (3) →1
r4: 4,5,6,7 (2)→0
r8: 8,9,10,11 (2) →0

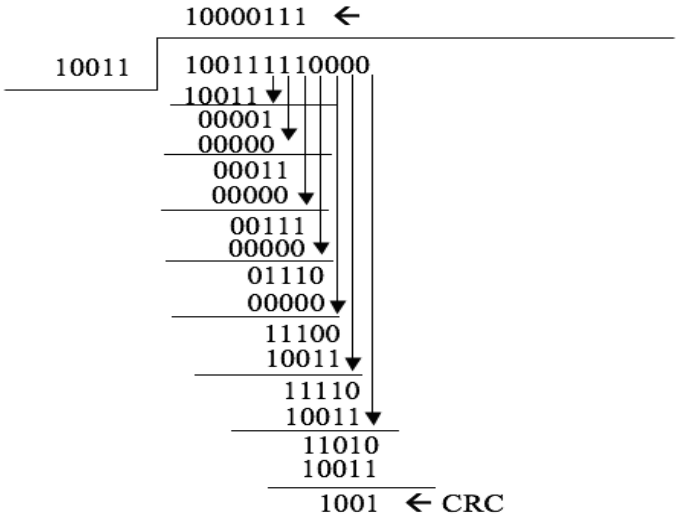
r8 r4 r2 r1
0 0 1 1 → error in bit in position 3 should be 1

The Correct code word is

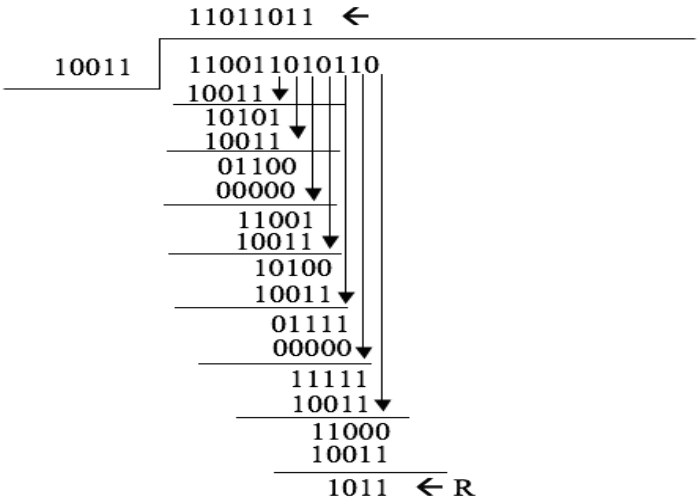
11	10	9	8	7	6	5	4	3	2	1
1	0	1	0	0	1	1	0	**1**	1	0

Data word is 1010111

3) Given the sequence **1 0 0 1 1 1 1 1** and a divisor of **10011**, find the CRC.



4) Assume that the following frame **(1 1 0 0 1 1 0 1 0 1 1 0)** is received across a data link using CRC for error detection. And the divisor is represented by the equation of $x^4 + x + 1$. Check whether the received frame is correct or incorrect. If the frame is correct, then deduce the original message.



5) Find the polynomial equivalent of **011000111001.**

$x^{10}+ x^9+x^5+x^4+x^3+1$

6) Find the checksum for the following bit sequence. Assume a 16-bit segment size.

0001111010110001
1001001001000111

		1	1	1	1							1	1	1		
0	0	0	1	1	1	1	0	1	0	1	1	0	0	0	1	
1	0	0	1	0	0	1	0	0	1	0	0	0	1	1	1	+
1	0	1	1	0	0	0	0	1	1	1	1	1	0	0	0	SUM
0	1	0	0	1	1	1	1	0	0	0	0	0	1	1	1	CheckSum

7) Suppose the receiver receives
1001001110010011 1001100001001101 0100111100000111
Check if the data received has error or not by using Checksum.

1001**0011**1001**0011**
1001**1000**0100**1101**
0100111100000111
_____
1 0111101011100111 partial sum
1 + carry
_____
0111101011101000 sum

Complement 1000010100010111 means that the pattern is corrupted.

8) Draw the sender and receiver flow diagram for a system using Stop-and-Wait ARQ. Given the following actions:

a. **Frame 0** is sent; frame 0 is acknowledged.
b. **Frames 1** is lost.
c. **Frame 1** is acknowledged.
d. **Frames 0** is sent; frame 0 is acknowledged but the acknowledgment is lost.
e. **Frame 0** is acknowledged.