ID:_____   Name:_____

1. Review: Logical connectives (or operators) of two propositions p and q.

Negation of p	$\neg p$	not p
Conjunction of p and q	$p \wedge q$	p and q
Disjunction of p and q	$p \vee q$	p OR q
Exclusive Or of p and q	$p \oplus q$	p XOR q
Implication of p and q	$p \rightarrow q$	if p, then q
Biconditional of p and q	$p \leftrightarrow q$	p if and only if q

2. Truth table of Compound Propositions:

Construct the truth table of the compound proposition $(p \vee \neg q) \rightarrow (p \wedge q)$

p	q	$\neg q$	$p \vee \neg q$	$p \wedge q$	$(p \vee \neg q) \rightarrow (p \wedge q)$
T	T	F	T	T	T
T	F	T	T	F	F
F	T	F	F	F	T
F	F	T	T	F	F

3. Precedence of Logical Operators

Operation	Precedence
$\neg$	1
$\wedge$	2
$\vee$	3
$\rightarrow$	4
$\leftrightarrow$	5

Adding parentheses to specify the order of operators in each compound proposition.

(a) "$\neg p \wedge q$" means ___$(\neg p) \wedge q$___ , ___NOT $\neg (p \wedge q)$___.

(b) "$p \vee q \wedge r$" means ___$p \vee (q \wedge r)$___.

(c) "$p \wedge q \vee r$" means ___$(p \wedge q) \vee r$___.

(d) "$p \rightarrow q \vee r$" means ___$p \rightarrow (q \vee r)$___.

(e) "$p \vee q \rightarrow r$" means ___$(p \vee q) \rightarrow r$___.

4. Truth Value and **Bit**:

Computers represent information using bit which has two values: __1__ and __0__ . A bit can

be used to represent a truth value because there are two truth value: _true_ and _false_.

Truth Value	Bit
True	1
false	0

5. **Boolean Variable**, bit string and bit operations

A variable is called a __Boolean__ variable if its value is either true or false.

A __bit string__ is a sequence of zero or more bits. The __length__ of this string is

the number in the string.

The table for Bit Operators OR, AND, XOR:

XOR

x	y	$x \vee y$	$x \wedge y$	$x \oplus y$
1	1	1	1	0
1	0	1	0	1
0	1	1	0	1
0	0	0	0	0

6. Bitwise OR, Bitwise AND, Bitwise XOR:

We define __bitwise OR__ , __bitwise AND__ , __bitwise XOR__ of two strings of the

same length to be the strings as their bits do OR, AND, XOR of the corresponding bits in the

two strings, respectively.

7. Find bitwise OR, bitwise AND, bitwise XOR of the two given bit strings.

01 1011 0110 and 11 0001 1101

bitwise OR
```
01 1011 0110
11 0001 1101
_____
11 1011 1111
```

bitwise AND
```
01 1011 0110
11 0001 1101
_____
01 0001 0100
```

bitwise XOR
```
01 1011 0110
11 0001 1101
_____
10 1010 1011
```