English Name: ____________________

Quiz Preparation: Converting Values

1. Write the appropriate vocabulary word next to its definition.

a)	nibble	a set of 4 binary digits
b)	byte	a set of 8 binary digits
c)	binary	a name for the base-2 number system
d)	hexadecimal	a name for the base-16 number system
e)	complement	an operation that inverts (or "flips") the bits of a binary number (0 ↔ 1)
f)	convert	to change from one form to another, for example from binary to decimal

2. Given the binary number 1110_1001, make the conversions specified below. It is easy to make small mistakes, so show your work for partial marks.

a) Convert the value to decimal, assuming the byte represents an ***unsigned number***.

bit position		7	6	5	4	3	2	1	0
place value	exponential	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
	decimal	128	64	32	16	8	4	2	1
binary digits (bits)		1	1	1	0	1	0	0	1
sum of decimal bit values		128 + 64 + 32 + 8 + 1 = **233**							

b) Convert the value to decimal, assuming the byte represents a ***sign and magnitude***.

bit position		- / +	6	5	4	3	2	1	0
place value	exponential		2^6	2^5	2^4	2^3	2^2	2^1	2^0
	decimal		64	32	16	8	4	2	1
binary digits (bits)		1	1	1	0	1	0	0	1
sum of decimal bit values		- (64 + 32 + 8 + 1) = **-105**							

c) Convert the value to decimal, assuming the byte represents a ***two's complement*** number.

place value in decimal	128	64	32	16	8	4	2	1
original two's complement	1	1	1	0	1	0	0	1
complement the bits (flip)	0	0	0	1	0	1	1	0
add one (+1)	0	0	0	1	0	1	1	1
sum of decimal bit values	16 + 4 + 2 + 1 = 23; negated = **-23**							

d) Convert the value to *hexadecimal*.

1	1	1	0	1	0	0	1
E				9			

English Name: ____________________

Quiz Preparation: Converting Values

3. Again working with the same binary number, 1110_1001, complete the tables below. It is easy to make small mistakes, so show your work for partial marks.

a) Perform a ***left shift*** by <u>two bit positions</u>, then convert the result to decimal, assuming the byte represents an ***unsigned number***.

place value in decimal	128	64	32	16	8	4	2	1
original unsigned number	1	1	1	0	1	0	0	1
logically shifted unsigned number	1	0	1	0	0	1	0	0
sum of decimal bit values	128 + 32 + 4 = **164**							

b) A ***left shift*** by <u>two bit positions</u> should be equivalent to multiplication by four (×4); however the value calculated in part (3a) is obviously not even close to four times the value calculated in part (2a). What word describes what happened.

overflow

c) Perform a ***logical right shift*** by <u>two bit positions</u>, then convert the result to decimal, assuming the byte represents an ***unsigned number***.

place value in decimal	128	64	32	16	8	4	2	1
original unsigned number	1	1	1	0	1	0	0	1
logically shifted unsigned number	0	0	1	1	1	0	1	0
sum of decimal bit values	32 + 16 + 8 + 2 = **58**							

d) *Perform a* ***logical right shift*** by <u>two bit positions</u>, as was done in part (c), then convert the result to ***hexadecimal***.

0	0	1	1	1	0	1	0
3				A			

e) What mathematical operation (operator and operand) is equivalent to a ***logical right shift*** by <u>two bit positions</u>, assuming the byte represents an ***unsigned number***.

operand (binary)	operator	operand (decimal)
1110_1001	÷	4 (or: 2^2)

Note: the reason this mathematical expression isn't exactly true is due to the ***loss of precision*** as bits are shifted out of the number, and lost.

f) Perform an ***arithmetic right shift*** by <u>two bit positions</u>, then convert the result to decimal, assuming the byte represents a ***two's complement*** number.

place value in decimal	128	64	32	16	8	4	2	1
original unsigned number	1	1	1	0	1	0	0	1
arithmetically shifted signed number	1	1	1	1	1	0	1	0
one's complement	0	0	0	0	0	1	0	1
two's complement	0	0	0	0	0	1	1	0
sum of decimal bit values	4 + 2 = 6; negated = **-6**							