

Name: \_\_\_\_\_

Email: \_\_\_\_\_

1. Assume char is **5-bits** and short is **10-bits**, convert the following into formats described by each column.

	8051 assembler	big endian two's compl.	little endian two's compl.	big endian one's compl.	big endian signed magnitude
unsigned char range:					
signed short range:					
signed char x=-7;					
signed short x='7';					
unsigned char x=7;					
unsigned char x=023;					
signed short x='-7'					
unsigned short x=0x3ce;					

2. Disassemble the following 8-bit string into assembler and decimal:

	8051 instruction	big endian two's compl.	little endian two's compl.	big endian one's compl.	big endian signed magnitude
11000011b					
0x3a					

3. Using C++ operator precedence, **add** the correct parenthesis:

$w = a - b \& d \ll f ;$
$w = a * b - 5 \& 0xff / z ;$

4. Using C++ convert the following using the following values:

**register unsigned char u=7, a=0xFF, b=0x01, c=0x72;**

**register signed char s=3, w=0x45, x=0x80, z=-2;**

State if **overflow or carry** has occurred after execution. Assign the 8051 registers as follows: u=R0, c=R1, b=R2, d=R3, s=R4, w=R5, x=R6; z=R7;

	two's complement big endian	8051 instructions	OV	C
u = ~ b;				
u = b & c;				
u = -c;				
u = c   d ^ c;				
u = d >> 2;				
d++;				
s = -w;				
s += w;				
s &= w;				
s = u = x += z;				

5. Using the 8051 instruction, assemble the instruction into hex, and then execute it showing the clock time in machine cycles:

Mem. Addr.	Machine instructions	Assembly	Clock Time	PC	OV	CY	AC	Reg. A	Reg. R2
0x0		MOV A,#0x42	0	0x0	0	0	0	0xff	0xff
		MOV R2,#0xce							
		CLR C							
		CPL C							
		ADDC A,R2							
		Total Time=							

6. Convert the 24-bit number 0x204298 to mime base64: \_\_\_\_\_
  
  
  
  
  
  
  
  
  
  
7. Assuming standard C++ data types, write a "single" C++ code statement of setting both bit  $d_5$  and  $d_4$  leaving all other bits unchanged in the variable *char x*.
  
  
  
  
  
  
  
  
  
  
8. Write the 8051 instructions for problem 7 and assume  $x$  is register R7.
  
  
  
  
  
  
  
  
  
  
9. Assuming standard C++ data types, write a "single" C++ code statement of setting both bit  $d_5$  and  $d_4$ , clearing bit  $d_2$  and leaving all other bits unchange in the variable *char h* ;
  
  
  
  
  
  
  
  
  
  
10. Write the 8051 instructions for problem 9 and assume  $h$  is register R2.
  
  
  
  
  
  
  
  
  
  
11. Write a C++ function to return 1 if an character has three 1's: unsigned char three(char in) ... . For example: three(0x51) returns 1; three(0x50) returns 0;