Last Name:_____

First Name:_____

Student ID:_____

Score: /79

I do swear that all work done on this Midterm is my own. I swear that I did not copy any answers from anyone or attempt to cheat an any way. I swear this by the honor of my name and my true and trustworthy word .

Signature: _____

Name of the Person Sitting to your LEFT: _____

Name of the Person Sitting to your RIGHT: _____

1. (3 points)If you have 43 people at a party, what is the minimum number of bits you would need to represent them all?

   6 bits

2. (2 points) Everything in the computer is represented using what?

   bits

3. (5 points) Convert 10 10 11 01 11 11 from binary to hex.

   0xADF

4. (5 points)Convert 0xA7C1 to binary

   1010 0111 1100 0001

5. (10 points) Convert 264_7 to its its base 5 representation.

   $264_7 = 2*7^2 + 6*7^1 + 4*7^0 = 144_{10}$

   144 / 5 = 28 R 4
   28 / 5 = 5 R 3
   5 / 5 = 1 R 0
   1 / 5 = 0 R 1

   1034 base 5

6. (2 points per box) For each of the following **8 bit numbers** write its value in **base 10** if it is represents an unsigned number, a signed using signed magnitude, and and a signed number using 2's complement

Binary Representation	Unsigned	Signed Magnitude	2's Complement
1100 0110	198	-70	-58
0000 0011	3	3	3
1001 0000	144	-16	-112

7. (20 points) Write the IEEE floating point representation of -31.25. For replicated bit values you may write the value of the bit * the number of repetitions. Ie if you wanted 10 0's in a row you would write '0' * 10. Please **Clearly Indicate** which bits belong to which fields and the order of the fields.

Sign	Exponent	Mantissa
1	131 = 10000011	111101 + '0' *17

8. (5 points) Given that we have the following value in memory

Address	4	5	6	7
Value	0xFF	0x7A	0x12	0x06

What would its value in hex be if the machine was little endian? Its value if the machine is big endian?

Little Endian: **0x06127AFF**

Big Endian: **0xFF7A1206**

9. (3 Points each) For this problem bits are numbered from 0 to 31. Given a variable called **`Val`** declared as as an **`unsigned int`,** write valid C/C++ code to
   1. Set bits 6, 9, and 20 of **`Val`** to 1

   Val |= (1 << 6) | (1 << 9) | (1 <<20) or 0x100240 or 1049152

   2. Set bits 2, 5, and 8 to 0.

   Val &= ~((1 << 2) | (1 << 5) | (1 << 8))

   3. Examine the value of bit 17.

   Val & (1 << 17) or 0x20000 or 131072

   4. Display bits 5 - 11 as an unsigned int

   cout << (val >> 5) & 0x7F

10. (5 points) Assuming a word size of 1 byte, does the following bit pattern represent the number 66?
    0100 0010

    Maybe. It depends on how we interpret the bits.

11. (4) What are the 4 major steps of the CPU cycle?

    Fetch, Decode, Execute, Write

12. (5 points) Given we have an array in **C** called Array declared as
    ```
    int Array[50][5][6][20];
    ```
    Write an expression that would access Array[I][J][K][L] that does not use brackets. Hint: cast Array into an int*.

    ***((int*)Array + I * 5 * 6 * 20 + J * 6 * 20 + K * 20 + L)**

13. (5 points) Given we have an array in **C** called Array declared as
    ```
    int**** Array;
    ```
    Write an expression that would access Array[I][J][K][L] that does not use brackets.
    ***(*(*(*(Array + I) + J) + K) + L)**

Extra Credit. (5 points) Given a **column major** matrix that has N dimensions with the size of each dimension being $d_1, d_2, d_3, d_4, \ldots, d_N$ and indices $i_1, i_2, i_3, i_4, \ldots, i_N$ write the generalized formula to calculate the 1 Dimensional equivalent index.

See Wikipedia for a nice formula.

11. (5 points per blank) The following C code implements the cumulative sum of an array x with length x_len

```
for(int I = 1; I < x_xlen; I++)
   x[i] += x[i-1];
```

The following assembly code seeks to emulate the same behavior. Assume that x is label in the data section.

```
     .text
     .equ x_len, length of x
     .equ wordsize, 4
init: #initialize all values before the for loop will be run here

     movl_____________________________#blank 1


     movl_____________________________#blank 2

     ret
end_init:

_start:
#esi will hold the pointer to x
#ecx will be the counter
#eax will hold temporary values
call init

_____________________________________#blank 3


     cmpl $x_len, %ecx


     _________________ end_for_loop #blank 4

     decl %ecx


     movl ____________________, %eax #blank 5


     incl ____________________ #blank 6

     addl %eax, (%esi, %ecx, wordsize)
     incl %ecx
     jmp for_loop
end_for_loop:
done: movl %eax, %eax
```