Practice problems

[1]  
a. Write the number AB31 hex in binary: _______________________________.  
b. Write the number 73 decimal in hex: ______________ and in binary: ______________.  
c. In order to access all bytes in a 64MB memory, an address must have at least _____ bits.

[2]  
Write a C procedure, int foo(int x), that returns the index of the most significant bit of x whose value is 1. For example, if bit 15 of x is the most significant bit of x whose value is 1, then foo should return 15. If no bits are 1, then foo should return -1.

[3]  
Write an x86 assembly procedure bar that takes two parameters, an integer n and a pointer p to an integer array (i.e. p is the address of the start of the array), and adds up the first n integers of the array and returns the result.

[4]  
Write in C a recursive version of the bubble sort procedure (hint: it should take as parameters an array a[] and an integer size that gives the size of the portion of the array that is currently unsorted).

[5]  
Build, from AND, OR, and NOT gates, a circuit that represents the two-bit “less-than” function. That is, it has two two-bit inputs, A and B (i.e. A consists of 2 bits and B consists of 2 bits), and a single one-bit output, R, such that R is true (i.e. 1) when A< B.
[6] **True/False.** Circle the appropriate choice (there are no trick questions).

(a) T F In C, a pointer variable is used for storing an address.

(b) T F In x86 assembly, the eax register can be used to hold integers but not addresses.

(c) T F In x86 assembly, the instruction “shl eax,3” (or “shl $3,%eax” in AT&T syntax) multiplies the value in eax by 8.

(d) T F If, in C, an array is declared by “int a[10];”, then accessing a[11] during execution will generate an error message.

(e) T F A callee-saved register should be saved before it is written to in a procedure.

(f) T F C has no built-in boolean type, rather the value 0 is used to represent false and all other values represent true.

(g) T F ebp is a caller-saved register.

(h) T F In x86 assembly, local variables in a procedure are generally declared in the .data section.

(i) T F Compiling C into x86 assembly comprises translating each C statement into a single assembly instruction.

(j) T F In x86 assembly, a label corresponds to an address.

[7] Consider the following x86 code fragment for computing the sum of an array of 10 32-bit integers, pointed to by ecx.

```
    mov $0,%eax          #
    mov $0,%edx          #
    TOP:                 
      cmp $10,%edx       #
      jl OUT            #
      add (%ecx,%edx,),%eax  #
      inc %edx           #
      jmp TOP            #
    OUT:                 
      (a) There are two bugs in the code. What are they?

      (b) Put a comment following each "#" in the above code to describe what that instruction does (or is supposed to do, in the case of a bug).
```
(a) Define a C struct type CELL that contains the following fields: an integer x, a string y, and a next field that points to another structure of type CELL.

(b) Write in C a procedure corresponding to the declaration int list_length(CELL *head); that returns the length of a linked list whose first element is pointed to by head.

[9] **Circle the correct answer among the choices given.**

(A) Which language of the following is the most hardware independent?
   a. High Level language  
   b. Assembly language  
   c. Machine language  
   d. They are all hardware independent

(B) If we want to design a computer system, what is the correct order?
   a. develop the ISA first, then the control unit, then the datapath  
   b. develop the datapath, then the ISA, then the control unit  
   c. develop the control unit, then the ISA, then the datapath  
   d. develop the ISA first, then the datapath, then the control unit

(C) If we want to remove an instruction from the instruction set of a processor, which of the following needs to be updated?
   a. control unit  
   b. datapath  
   c. the compiler  
   d. all of them must be updated

(D) The main difference between MIPS ISA and x86 ISA is:
   a. MIPS ISA is more complicated  
   b. MIPS ISA is less complicated  
   c. x86 uses way more registers  
   d. x86 has less instructions
In C, suppose you have a binary tree whose node type is defined by the following:

```c
typedef struct node {
    int value;
    struct node *left;
    struct node *right;
} NODE;
```

(a) Below is a version of the procedure that creates a new node and returns a pointer to it, so that the new node can be inserted into the binary tree. This code compiles without errors, but doesn't execute correctly.

```c
NODE *new_node(int val)
{
    NODE n;
    n.left = NULL;
    n.right = NULL;
    n.value = val;
    return &n;
}
```

Explain what is wrong.

(b) Write the correct code for new node.

(c) Write a C procedure

```c
void insert_left(NODE *p);
```

that inserts the node pointed to by p as the leftmost leaf of the tree. Assume that the global variable root points to the root of the tree (but don't assume that root is not NULL).