COS126: General Computer Science **Spring 1996**

Exercise # 5

Handed out:April. 12, 1996

These exercises are meant to guide class discussion. *Do not* turn them in.

1. What is the running time of this function? Express your answer in $O()$ notation.

```
int f(int a, int b)
{
 int i, j=0;
 if(b==0)
    return 100;
 else
    return 100+ f(a, b-1);
}
```

2. Suppose that `struct tree` is defined as in the lecture notes. Now answer the following questions.

   (a) What does the following function do?

   ```
   void mystery(struct tree *t)
   {
    t->value = -5;
    if (mystery-> left !=NULL)
       mystery(t->left);
    if (mystery->right != NULL)
       mystery(t->right);
   }
   ```

   (b) According to our convention for binary search trees, the value at every node is greater than every value in its left subtree, and less than every value in its right subtree. Suppose we wish to reverse this convention, so that now the value at the node has to be *less* than every value in its left subtree.

   Write a function `old-to-new` that converts a binary search tree to fit the new convention. The function should have the following declaration, where `t` is a pointer to a tree that fits the old convention.

   `void old-to-new(struct tree *t);`

3. Which of the following are syntax errors and which are not?

```
char c="a";
char *c="a";
char c[10]="a";
char c[]="a";
```

4. Show the "parse tree" generated for this expression: `(((a+b)*c)+(d+c))`.

5. Below are two routines in TOY machine code, each starting at address 00. The first is a subroutine and the second is the main routine. What is the result of linking these together and loading them starting at address 11? What is the initial PC value?

```
00:  B000    R0 <- 0
01:  2312    R3 <- R1-R2
02:  6306    if R3 < 0 jump to 06
03:  1320    R3 <- R2 + R0 = R2
04:  1210    R2 <- R1 + R0 = R1
05:  1130    R1 <- R3 + R0 = R3
06:  7400    jump to address in R4

00:  B505    R5 <- 05, an address
01:  9150    R1 <- M[R5 + 0]
02:  9251    R2 <- M[R5 + 1]
03:  84??    jump to subroutine; return address in R4
04:  0000    halt
```

6. Which of the following must be true about virtual memory:

   (a) A virtual address must have at least as many bits as a physical address.

   (b) A physical address must have at least as many bits as a virtual address.

   (c) Both of the above.

   (d) None of the above.

7. Suppose a paged virtual memory system uses 2^{12}-byte pages (also known as 4K byte pages, a typical size). The address is 32 bits long and there are 2^{28} bytes (256Megabytes = 256 MB) of physical memory. Suppose the following is a fragment of the page table (addresses in hex):

page no.	in memory	physical location
FFFF1	yes	CD14
FFFF2	yes	53B1
FFFF3	no	disk location
FFFF4	yes	FFF4
FFFF5	no	disk location
FFFF6	yes	CD15

   a Given the following sequence of memory accesses, which is the first to cause a page fault? (A page fault is what was referred to as a "miss" in lecture – the accessing of an address that is not in physical memory.)

```
FFFF4010
FFFF2FEC
FFFF6118
FFFF5304
FFFF1000
```

b For the memory references above before the first page fault, what are the physical addresses used?

c For the first page fault above, suppose page FFF1 is replaced to bring the new page from disk. What does the fragment of the page table look like after this is done.

d After the page replacement just described in (c), are there any more page faults for the sequence of memory accesses given above?