Assessment 1

This assessment will help us determine some of the programming exercises in the first few weeks of the course. Students enter CSE 422S with a wide range of programming background. Inability to answer some of the questions may indicate the need to cover some topics but does not necessarily mean that you should not be taking this course.

Problem 1 (Number Representation)

Suppose that a variable x is declared to be an int with a value of 107 (i.e., 'int x = 107'). What is the value in hexadecimal?

Problem 2 (Arithmetic Operators, While-Loop)

If a = 567 below, what is the value returned from the function foo?

```c
int foo (int a) {
    int b, c;
    c = 0;
    while (a) {
        b = a%10;
        c = c*10 + b;
        a = a/10;
    }
    return c;
}
```

Problem 3 (Array, For-Loop)

Suppose that n[ ] is defined to be:

```c
int n[10] = { 5, 3, 7, 6, 15, 8, 10};
```

and nitems = 7, the number of items in n[ ]. What is the value returned from the function foo below?

```c
int foo ( int n[], int nitems) {
    int j, k;
    j = n[nitems-1];
    for ( k = nitems-2; k >= 0; k-- ) {
        if (j > n[k])  j = n[k];
    }
    return j;
}
```
Problem 4 (Integer Pointers)

Consider the following declarations:

```c
int x = 10;
int y = 20;
int z;
int *px = &x;
int *py = &y;
```

Which of the following expressions are valid? If valid, give the value of z.

a) $z = (*px)++$;

b) $z = (*py) + y$;

c) $z = (*x) + (*y)$;

Problem 5 (String Pointers)

What is printed by the following code?

```c
char s[] = "abcd";
char *p = s;
p++;
printf("%s ", p);
(*p)++;
printf("%s ", s);
```
Problem 6 (Array of String Pointers)
Consider the following declaration:

```c
char *A[] = { strdup("I"), strdup("am"), strdup("in"), strdup("CSE 422S") };
```

where `strdup(s)` returns the address of a copy of the C string `s`. In each part below, assume that you are starting with the above declaration.

a) Write code to change the C string "CSE 422S" to "422S" by *overwriting* the string located at `strdup("CSE 422S")` above.

b) Write code to change the C string "CSE 422S" to "422S" by overwriting only the appropriate characters located at `strdup("CSE 422S")` above but leaving all other characters intact.

c) Write code that will print out in one line the C strings contained in the A array.

Problem 7 (The Make Utility)
The following is a simple Makefile for the 'make' utility. What will happen if I enter the command 'make a'?

```
a:   b  c.txt

b:
   rm -f *.o

c.txt:  foo.txt
   sort foo.txt > c.txt
```
Problem 8 (Interfaces, Abstract Data Type)

We consider a symbol table that maps string symbols to integer values. Assume that an implementation of the symbol table has been provided with one of the interfaces given below (You can choose either the C++ or the C version).

Write code that creates a symbol table with room for \( N \) symbols and then sets the values of the symbols 'a' and 'b' to 2 and 3 respectively. Assume that the value of \( N \), an integer, is not known until run-time, but you do not need to provide code showing how \( N \) is initialized. The code should also print the values of 'a' and 'b' that are in the symbol table before and after the values of 'a' and 'b' are set.

C++ Interface:

- `SymTbl(int n)`: Create a new symbol table with room for 'n' symbols
- `int value(const char *name)`: Return the value of symbol 'name'
- `int set(const char *name, int value)`: Set the value of the symbol 'name' to 'value'
- `int count(void)`: Return the number of items in the symbol table

C Interface:

- `typedef struct SymTblDesc * SymTbl_t`: Symbol table descriptor type declaration
- `SymTbl_t p = SymTbl(int n)`: Create a new symbol table with room for 'n' symbols and return a pointer to the descriptor
- `int value(SymTbl_t p, const char *name)`: Return the value of symbol 'name'
- `int set(SymTbl_t p, const char *name, int value)`: Set the value of the symbol 'name' to 'value'
- `int count(SymTbl_t p)`: Return the number of items in the symbol table