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

If \( a = 567 \) below, what is the value returned from the function \( \text{foo} \)?

```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 2

a) Suppose that \( n[] \) is defined to be:

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

and \( \text{nitems} = 7 \), the number of items in \( n[] \). What is the value returned from the function \( \text{foo} \) 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;
}
```

b) What is returned by the function \( \text{foo} \)?
Problem 3
Consider the following declarations:

```c
int x = 10;
int y = 20;
ext = 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 4
What is printed by the following code?

```c
char s[] = "abcd";
char *p = s;
p++; printf("%s ", p);
(*p)++; printf("%s ", s);
```

Problem 5
Consider the following declaration:

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

a) Write code to change the string "CSE 422S" to "422S" by overwriting the string.

b) Write code to change the string "CSE 422S" to "422S" by overwriting the appropriate characters but leaving all other characters intact.

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

Problem 6
The following is a simple Makefile for the 'make' utility. What will happen if I enter the command 'make a'?

```makefile
a: b c.txt

b:
rm *.o

c.txt: foo.txt
    sort foo.txt > c.txt
```
Problem 7

Suppose that the current directory contains the three files foo.c, foo.o and foo. foo.c is a simple, self-contained C++ program. foo.o is the object module produced from foo.c, and foo is the executable. Write a simple Makefile such that if and only if foo.c has a more recent modification date, the command 'make' will cause the g++ compiler to produce an up-to-date version of foo.o and foo. Also, 'make clean' will remove the file 'foo' and all .o files from the current directory.

Problem 8

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 dynamically creates a symbol table with room for 10 symbols and then sets the values of the symbols 'a' and 'b' to 2 and 3 respectively. The code should also print the values of 'a' and 'b' 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(char *name): Return the value of symbol 'name'
- int set(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: Symbol table descriptor type declaration
- SymTbl p = SymTbl(int n): Create a new symbol table with room for 'n' symbols and return a pointer to the descriptor
- int value(SymTbl p, char *name): Return the value of symbol 'name'
- int set(SymTbl p, char *name, int value): Set the value of the symbol 'name' to 'value'
- int count(SymTbl p): Return the number of items in the symbol table