CISC105 Spring 2007 Lab11

- This lab is designed to help with your project. You'll write small programs that demonstrate things you are supposed to do in the project. In general, writing small test programs that demonstrate the concepts you want to employ is a very good way to approach a large assignment.
- Summer time is coding time! Take a laptop to the pool and start talking about null-terminating characters. You'll meet lots of interesting people.
- Remember to code a line or two, then stop and test. Test ! = compile. You have to run your program and print results before you know if something works, and running it once doesn't test it much.
- Some programs below are associated with a question. Answer the questions using C comments below your code in the program file unless the instructions say otherwise.

For each numbered problem below you will write a small program. Name each program lab11.n.c, where n is the number in the list below. For example, the name of the file for the first will be lab11.1.c

Programs

1. Define a struct type called Bug (why is this capitalized?). A bug has an integer number of legs, a string "call" (for sounds like "bzzz" or "sssss" or "click"), and a "terminal" sound that it makes when stepped on, like "crunch" or "splat".

Write a function that nicely prints a Bug.

Declare a Bug with six legs that says "ch-ch-chit" and goes "squish", and a Bug that says "hey!" and goes "ow". Print both Bugs using your function.

- 2. Make an array of four Bugs. Do not initialize the array with constants. Use assignment and strcpy to put values in the Bugs. Use a loop to print your array using your print function.
- 3. Sort an array of Bugs based on the number of legs. Make sure your data demonstrate your program well. What will have to change in the sorting function we wrote in class?
- 4. Make a 2-d array of char, 4 rows and 8 columns. Put four words in the array. Sort the array using the sorting function from class. Inside the sorting function, you won't be able to use comparison and assignment operators as we did with ints. Instead, you'll need a function to compare two strings, and another function to copy strings from one place to another (for the swap part).

You should have a total of 4 programs named lab11.01.c to lab11.4.c. Make a single script file (see lab00 for the scripting instructions) where you cat, compile, and run each one in its final form (if it didn't compile, don't run it in the script - mark the place in the printed script file with a colored marker so it stands out).

On the first page of every printed copy for this course, your name, section, and TA's name must appear.

Submit all program and script files on MyCourses before midnight Thursday of next week, and give the paper version to your TA at the beginning of your Friday lab (or in lecture Friday if you have a Wednesday lab). Note: cat, compile, and run each program in order! Do *not* cat all programs, then compile, etc.