

CISC105 Spring 2006 Lab09

- Write a program for each of the following problems. Be sure to save every separate program. All programs must be properly commented and indented (see Assignment Standards on the class website).
- Why use good style? Good style makes code easier to read and debug. If your program is easy to read and debug, it is more likely to work correctly. If it works correctly it is less likely that you'll be responsible for the collapse of a bridge, or a mistake in someone's insurance transaction. Always assume your program will end up doing something important! At least you'll get a good grade.
- Name each program lab09.n.c, where n is the number in the list below. For example, the name of the file for the first will be lab09.1.c. Put the files in your lab09 directory.

Programs

1. Make a two dimensional array of char, where each row is 20 chars long and there are 4 rows. Fill it with the four words from lab09.data using a file pointer. Then use strcmp to find the index of the first word, alphabetically. Print all words, then print the index and the word you found to be first.
2. Look at the program on page 676 of your text. Declare a main() with parameters just like those shown. The types of these parameters are specified in C, and the names are traditional but not mandatory (but if you change them, make sure your names are at least as descriptive). Create a print statement for an integer and 3 strings. Print the int **argc** and the strings **argv[0]**, **argv[1]**, and **argv[2]**. Now, in a shell, call your executable **program** on the arguments "programming" and "rocks". Do this the same way you can call the program "cc" on a file name.
3. Now merge the ideas from 1 and 2. Create a program that lets you type

```
./a.out dog able baker charlie
```

and then finds then first word of the four words that were put in argv[0] through argv[4]. Print which word, and the word you found. Test properly with other combinations.

4. Define a struct dog that stores number of legs, weight, name, and breed. Write a function that prints a single struct dog. Then declare and print the dogs 3-legged Sadie, a 55 lb poodle, and Rosie, a four-legged 123.6 lb Neopolitan Mastiff.
5. Declare an array of 10 struct dogs¹. Make up data in a data file and read all ten dogs into your array of structs. Find the lightest dog and print it with an appropriate message. Find the dog with the first alphabetical name and print it with an appropriate message.

You have a total of 5 programs named lab09.1.c to lab09.5.c. Make a single script file (see lab00 for the instructions) where you cat, compile, and run each one in its final form. Follow the instructions above for adding written information to your script.

Submit all 5 program files *and* your script on WebCT, and give the paper version of the complete script file **only** on paper to your TA at the **beginning** of your next lab (Wednesday labs submit Friday at the **beginning** of class). Note: Cat, compile, and run each program in order! Do *not* cat all programs, then compile, etc.

¹If you are bored you can make some type besides dogs, e.g. professors. Must have at least four data values of various types.