

CISC105 Spring 2006 Lab10

- 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? Do you want to do well in this class? Do you even read this part of the lab? Demand for computer scientists is expected to exceed supply substantially in the next ten years. Consider a CS major or minor if you are enjoying writing lab programs.
- Name each program lab10.n.c, where n is the number in the list below. For example, the name of the file for the first will be lab10.1.c. Put the files in your lab10 directory.

Programs

1. Look up the function atoi (pronounced “a to eye”; why?) in your H&K text. Then write a program that uses argv and argc (see the last lab). Call it using “a.out 123” so that 123 goes as three chars into one row of argv (which row?). Print out 123 using a %s format specifier to prove it is a string, not a number. Then convert it to an integer using atoi and print the integer with %d.

(Think about: If you had to write a function that converted a string of numeric chars to a single integer, how would you do it? Would isdigit() help?)

2. Make a two dimensional array of char, where each row is 20 chars long and there are 10 rows. Fill it with the words from lab10.data using a file pointer. Then use the selection sort from class with strcmp (how? what will strcmp replace?) to sort the words alphabetically. Print all words from the array as they are first read in (unsorted) and then again after you complete the sort.

When it is time to swap two words in the array, the idea will be the same as swapping integers - that is, you need a temporary place to store one word while you perform the swap. Where can you store one word? What string function will you use to put it there?

3. Copy the program from 2 and modify it as follows. Use argc and argv (why both?) to get the file name from which you will read data. Depending on what the user types, either copy the new data file name into a char array to be used by fgets, or leave the char array alone. If the user types “a.out” by itself, the program will open and read the data from lab10.data. If the user types a file name, as in “a.out myfile.data” the program will get data from that file instead
4. Now merge the ideas from 2 and 3. Create a program that lets you type

```
./a.out datafilename 30
```

and then reads 30 words from datafilename.

5. Write a function that takes character array as an argument, along with maximum size, and writes the data from the line into a struct dog (see previous lab). An example of data might be: ``barfy 3 21 collie``

Can you guess the next step in this progression of programs?

You are responsible for adequately testing all programs.

You have a total of 5 programs named lab10.1.c to lab10.5.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.