CISC105 Spring 2006 Lab08

- 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).
- Write a little program every day. Programs can make a thoughtful or romantic gift ("I coded this just for you. Look, no bugs!").
- Name each program lab08.n.c, where n is the number in the list below. For example, the name of the file for the first will be lab08.1.c. Put the files in your lab08 directory.

Programs

- 1. Use a for loop to print each character from a to z, and another for loop to print from A to Z. As you print each letter as a char, also print the same value as an integer to see the ASCII code for that char.
- 2. **strcmp** is a string function in C (so include string.h). Even though there is no type "string" in C, there are functions that rely on the structure we call a string (what is a string?). Print the following expressions as **integers**:

```
strcmp("boy", "apple")
strcmp("apple", "boy")
strcmp("cat", "cbt")
strcmp("bat", "bas")
strcmp("bat", "bar")
strcmp("bat", "baq")
strcmp("bark", "bark")
```

Explain briefly the return values from strcmp in terms of your results from 8.1.

- 3. **strlen** is another C string function, this time for finding the length of a string. Print the integer result of calling strlen on a char array containing the word "spud". Then, in the same program, write your own function **myStrLen** containing a while loop that will determine the length of a string argument and return the length. Demonstrate by calling the function on "spud" and "orangutan".
- 4. Look up the function **fgets** using **man** in your shell.

Write a program that opens a file pointer called **input** to read a data file called lab08.data, which you will download from the lab directory (test your pointer!). Declare a char array of size 80. Loop three times on a function call to fgets as follows:

fgets(nameOfYourArray, 80, input);

After each call, and still inside the loop, print your array as a string (what format specifier will you use?).

5. Write a program with a function that will look at an array of char and pull out the third word and put it in an array. For example, if you have an array of char containing the line

''a text file online, which do''

your function will extract the word "file", store it in a separate char array, and then print it. Remember, it is easier to write and debug a function if you start in a small, separate program and code incrementally.

If there is no third word in the line (how will you know where the line ends in the array?) your function should only store a null character.

Hint: one way to approach this would be to write a helper function similar to the one we wrote in class. Have it look at a certain index in the array and tell you the index of the next alphabetical char (i.e. the next thing that isn't white space).

- 6. Combine the program from 4 and the function from 5 so that you read in lines from a file and print the third word of each line (if it exists). Read all lines in the data file by comparing the return value of fgets to the constant NULL in a while loop, as shown in class.
- 7. Make a two dimensional array of char, where each row is 20 chars long and there are 10 rows. Read ten words, with no whitespace, into the ten rows using fgets. When this works, create a function that takes a 2-d array, row size, array length, and an open file pointer as parameters and fills the array from the file. The function will return 1 if it succesfully fills the array, or -1 if it cannot finish filling the array.

You have a total of 7 programs named lab08.1.c to lab08.7.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 7 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.