## CISC105 Spring 2007 Lab09

- Your labs are designed to teach you principles of programming. We go over those principles in lecture, but be sure that for each lab problem you know what you are supposed to be learning from it.
- 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 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

1. Locate the string library functions streat, strepy, and stremp in your textbook. Use the Unix "man" command to get information on them and see their prototypes (do you remember what "const" means?). Start getting familiar with the way things are described in man pages.

## **Programs**

2. Write a program that demonstrates the three functions from 1. Start by demonstrating the functions on string literals (things in double quotes). One of the functions won't work on just literals; when you find which one, comment out that portion, figure out why it won't work and write a brief explanation in your comments.

In the second part of the same program, call the three functions on initialized character arrays.

Your program should print messages saying what is happening at each step.

- 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 "dik-dik". 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 "goony bird" and "bandicoot". Note: myStrLen must do its own counting, not call strlen.
- 4. Declare a single array of 80 char and initialize it to "Bite the wax tadpole". Use a while loop to traverse the array and find the index of the start of the fourth word (word size shouldn't matter). Print the index and the letter found there. You may assume one space between words.

Add to your program so that a user can enter any four word phrase and the program will find and print the index and letter at the start of the fourth word. To read in multiple words to a single array, you won't be able to use scanf (why not?) so use the statement

fgets(nameOfYourArray, sizeOfYourArray, stdin);

5. Copy the program from 9.4. Modify it so that a user enters an integer n followed by a phrase (which will be 79 characters or fewer). The program will print the index and first letter of the nth word in the phrase. For example:

```
% a.out
Please enter a word number and a phrase, or -1 to stop:
2 Georgia on my
at index 9 letter is o
Please enter a word number and a phrase, or -1 to stop:
3 all dogs lie quietly
at index 10 letter is l
Please enter a word number and a phrase, or -1 to stop:
1 alas poor
at index 1 letter is a
Please enter a word number and a phrase, or -1 to stop:
6 put the lime in the coconut
at index 21 letter is c
Please enter a word number and a phrase, or -1 to stop:
-1
Farewell!
```

6. Declare a two dimensional array of ints:

int table[5][10];

Place the multiples of three into this array with a nested for loop. Then print the array with another pair of for loops, as follows:

> a.out									
0	3	6	9	12	15	18	21	24	27
30	33	36	39	42	45	48	51	54	57
60	63	66	69	72	75	78	81	84	87
90	93	96	99	102	105	108	111	114	117
L20	123	126	129	132	135	138	141	144	147

You should have a total of 5 programs named lab09.01.c to lab09.5.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.