CISC105 Fall 2006 Lab07

- Write a program or webpage 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).

- Practice your Unix commands when working on the lab. Try moving, copying, etc just to keep that information fresh in your mind.

- 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 lab07.n.c, where n is the number in the list below. For example, the name of the file for the first will be lab07.1.c. Put the files in your lab07 directory.

Programs

1. Declare an array of type char. A character is represented by putting a symbol inside single quotes, so `'a'` is of type char. Use characters in quotes the same way you would use an integer value, i.e. you can say

   ```
   char c = 's';
   ```

   Initialize the array to contain 8 characters as follows: `'h'`, `'a'`, `'t'`, `'\n'`, `'\0'`, `'r'`, `'a'`, `'t'`. After initializing the array to contain these characters, use a loop to print the whole array out, one element at a time, using the format specifier `%c`.

2. Once you have done 7.1 and it is working correctly, write a new program and print the entire array at once without a loop, using **only** the name of the array and the format specifier `%s`. Explain what is printed.

3. Copy the previous program. Try using assignment to change the third element of the array , `'t'`, to match the fifth element without using a quoted char constant (how?). Describe (writing on your script by hand) what happens when you print now.

4. Copy the previous program. Change the array using assignment so that all characters in the array will print except the last.

5. 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.

6. **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 "loris". 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 "loon" and "eland".

7. 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 letter 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);
```

8. Copy the program from 7.7. 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!
```

You should have a total of 8 programs named lab07.1.c to lab07.8.c. Make a single script file (see lab00 for the instructions) where you cat, compile, and run each one in its final form.

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