CSC 225
Assignment #7

Objectives:

- Practice writing a recursive function in C
- Understand what function calls look like at the machine level

Documentation:

- Your textbook from 225 - Especially Chapter 14
- Lecture notes.

Part 1: C Recursion (6 points)

```
/**
 * This function counts backward from x to 1 by one.
 * For example, calling this function with a value of 5 should
 * result in the following output:
 *         5, 4, 3, 2, 1
 */
void countBackwardFrom(int x)

/**
 * This function counts from one to x by one.
 * For example, calling this method with a value of 5 should
 * result in the following output:
 *         1, 2, 3, 4, 5
 */
void countForwardTo(int x)
```

You must write the above functions and *demo* them to your instructor. To demo, write a main that prompts for an integer. Then use that integer to call both functions. You don't have to get the commas perfect, but it's fun to try!

- The functions must be implemented recursively.
- You **may not** use static or global variables.
- You **may not** use helper functions.
- Be prepared to show your code to your instructor.

Part 2: Recursive charCount – Orientation (2 points)

Copy the file charCount.c to your account:

```
cp ~jworkman/www/225/Assignments/Asgn7/charCount.c  .
```
(Or click here.)

Look at the `charCount()` function that calculates the number of occurrences of a character in a string using recursion. Compile and run this code. Make sure you understand how it works. Ask your instructor for clarification if necessary. You will be translating this code into LC-3 assembly code using the function calling format from Chapter 14!

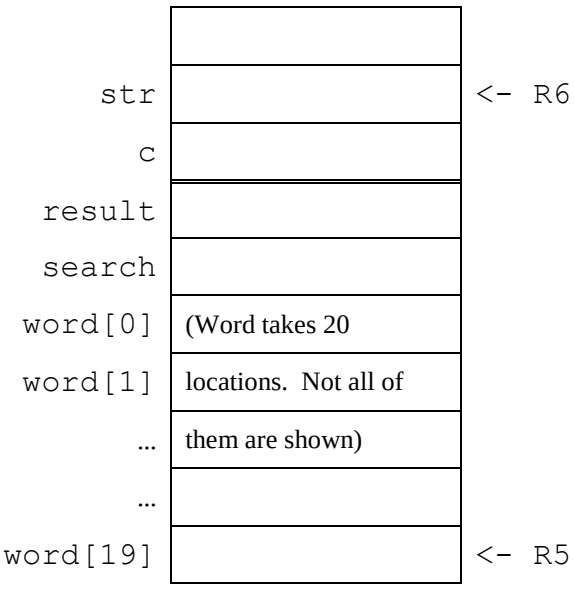
Now copy the file main.asm to your account:

```
cp ~ jworkman /www/225/Assignments/Asgn7/main.asm .
```
(Or click here.)

This is the LC-3 version of the `main()` function from charCount.c. Read this code and make sure you thoroughly understand it. Again, ask for any necessary clarifications.

Below is a partial picture of the stack as it looks right before the JSRR line is run to call the CHAR_COUNT subroutine. Everything below the double line is from main's activation record. Everything above the double line is for CHAR_COUNT. Draw this picture on your demo sheet and add to it what the stack will look like once the first call to CHAR_COUNT is **completely set up on the stack (the callee setup is complete).**

Have your instructor check your drawing (see Asgn7 demo sheet) before starting Part 3.



Part 3: Recursive charCount in LC-3! (12 points)

Finish the LC-3 implementation of charCount.c by writing the CHAR_COUNT subroutine in a file called charCount.asm. This file should begin at memory location x3300 and should contain the code for the CHAR_COUNT subroutine that is called from the main.asm file. Your CHAR_COUNT subroutine must work with the main.asm. You may not make any changes to main.asm.

Make sure you use the function call format presented in chapter 14. You will be pushing the parameters, bookkeeping, and local variables on the stack. *You will not get any credit for this part if you don't follow the format presented in chapter 14.* You must translate the charCount C code directly and exactly. Note that the charCount() function in the c code has a local variable. **You must use this local variable in your assembly code.**

- Note that if you use the stack for all your local variables you can trash any registers that you want. You should not have to save and restore registers.
- You may NOT preserve the count in a register between function calls. **You must get the returned value from the RT stack** and add to it.
- Your code must be recursive!
- There will not be more than 9 occurrences of the character in the string.
- Note that CHAR_COUNT will be both a callee (it's called by main, and by itself), and will be a caller (it calls a function – itself).

Debugging

Carefully look at the run time stack. You should see multiple activation records pushed onto the stack (one for each call to charCount). Based on the drawing that your instructor signed off, you should know what the first one looks like. Ask your instructor if you need help figuring out what successive activation records should look like. Try working with a very simple string like "Hi". You should see 3 activation records.

Deliverables

1. Demo sheet with signatures for Parts 1-3. Additionally you should have a drawing on your demo sheet for Part 2.
2. Handin your charCount.asm file by the end of lab on the day it is due. You may handin early and then simply re-handin if you need to make changes to your code. . **Handin will CLOSE at the end of lab.**

```
handin graderjw asgn7 charCount.asm
```