

## Cp Sc 1110 Program 5

**Due: Thursday Nov. 20th, 11:59 pm**

---

The purpose of this assignment is to give you practice in using pointers, command line arguments, and files. Sample input and output is the same as program 4.

### Problem Statement

1. Download Step
  - 1.1. Download my starting program 5 using one of two methods:
    - 1.1.1. To save on your local machine - Go to the samples page (Blackboard: Samples > Samples), right click on prog5Start.c, and save as into your desired location.
    - 1.1.2. To save on a SOC machine – Go to your desired location and enter  

```
cp /home/psterli/public_html/1110/samples/prog5Start.c ./
```
  - 1.2. From your desired location:  

```
cp prog5Start.c prog5.c
```
  - 1.3. Change prog5.c's author and date.
  - 1.4. If necessary, copy your sample4.txt to this location.
  - 1.5. Compile prog5.c and then test: 

```
./a.out < sample4.txt
```
2. Pointer Conversion Step
  - 2.1. Convert all prog5.c non-main functions to use pointers instead of arrays.
  - 2.2. From now on, all non-main functions should not have any brackets ([])'s).
  - 2.3. Compile and test: 

```
./a.out < sample4.txt
```
  - 2.4. Make a backup: 

```
cp prog5.c prog5-2.c
```
3. File I/O Step
  - 3.1. The program will now be changed to accept the file name of the file to read from the command line rather than using input redirection. Change prog5.c to accept command line parameters.
  - 3.2. At the beginning of main's executable instructions, verify that the file name has been specified on the command line.
  - 3.3. Move the instructions for reading pct to loadBulbs before the while loop.
  - 3.4. Modify the loadBulbs headers (prototype and definition) to make it a void function and add three parameters:
    - 3.4.1. A char pointer to the file name string.
    - 3.4.2. A float pointer to the variable receive the percentage.
    - 3.4.3. An int pointer to the variable receive the number of readings placed in the array.
  - 3.5. Modify the body of loadBulbs to
    - 3.5.1. Open a FILE for input using the file name string.
    - 3.5.2. Verify the file opened correctly. If not display a message indicating that the file could not be opened which includes the file name and then exit(2).
    - 3.5.3. Convert the input statements to read from the FILE rather than stdin.
    - 3.5.4. Close the file when finished.
    - 3.5.5. Through the parameters, return the percentage and the number of readings placed in the array as needed.
  - 3.6. Add the additional parameters to the call to loadBulbs in main.
  - 3.7. Compile and test without redirection: 

```
./a.out sample4.txt
```
  - 3.8. Make a backup: 

```
cp prog5.c prog5-3.c
```

4. Dynamic Allocation Step
  - 4.1. In prog5.c, make a copy of loadBulbs's prototype and definition (header and body).
  - 4.2. Change the name of one function to countBulbs (header and definition).
  - 4.3. Change both countBulbs headers to remove parameters for array pointer and max readings.
  - 4.4. Change both loadBulbs headers to remove parameters for the percentage and number loaded.
  - 4.5. Change the body of countBulbs so that it does not place values into the array, only counts and then returns the percentage and the count.
  - 4.6. Change loadBulbs so that it does not return the two values in the parameters.
  - 4.7. Remove the constant and the array from main.
  - 4.8. In main, change the call to loadBulbs to
    - 4.8.1. Call countBulbs and get the percentage and the number of readings.
    - 4.8.2. Dynamically allocate an array to hold the readings and verify the allocation.
    - 4.8.3. Call loadBulbs to load the readings into the dynamic memory.
  - 4.9. Verify that there are no brackets in non-main functions.
  - 4.10. Compile and test: `./a.out sample4.txt`

## Requirements

1. Your program must adhere to this assignment's problem statement requirements and the general programming assignment requirements. Violations will lead to deductions. General requirements: <http://www.cs.clemson.edu/course/cpsc111/lect/requirements.pdf>
2. Submit prog5.c electronically using Webhandin:
  - 2.1. If your program is stored on the SOC system and you are submitting from a non-SOC machine: You must first transfer your C program to your local machine. If your machine is a Windows machine you will use SSH Secure Shell File Transfer Client, MobaXterm Sftp download, or a comparable product. On a Mac OS or Linux machine, you will use sftp. See the appropriate documentation for instructions.
  - 2.2. Go to <http://handin.cs.clemson.edu> to open Webhandin.
  - 2.3. If requested sign in with your username and password.
  - 2.4. On the My Courses page, click on the link containing CPSC 1110.
  - 2.5. On the Assignment page select prog5.
  - 2.6. The place where you put your files is referred to as a "bucket". When browsing your bucket you'll be shown some information about the assignment and a list of files which you have submitted. From this page, you are able to submit a file or delete a previously submitted one. To submit a file, click the button toward the bottom of the page. To delete one, click the delete button next to the file. If you've submitted a file, there will be a time noted at the top of the page which marks when Webhandin received your submission.
  - 2.7. Submit prog5.c. After submitting, click on the filename link. Your program should be displayed.