CS 483 – Operating Systems
Spring 2017

PEX 0: C Proficiency - 12 Points

Due: 9 Jan @ 2300 (night before Lesson 3)

Help Policy

AUTHORIZED RESOURCES: CS483 textbook resources and the C99 standard **ONLY**. You MUST document any online resource (if you referenced the C99 standard online) you used to complete this assignment. **Note: Sites like stackoverflow are NOT authorized for this PEX!** This assignment is designed to assess your basic skills in the C programming language – you will be offered the opportunity to use other online sources for the remaining PEX's – but not this one!

NOTE:

- Never copy another person's work and submit it as your own. Here are a few blatant examples of copying:
  - Making an electronic copy of another cadet's solution and then modifying it slightly to make it appear as your own work.
  - Reading a printout or other source of another cadet's work as you implement your solution.
  - Completing your entire solution by following explicit instructions from another cadet, while he/she refers to his/her own solution
- Do not jointly implement a solution (outside designated pair programming partners).
- Help your classmates maintain their integrity by never placing them in a compromising position. Do not give your solution to another cadet in any form (hard copy, soft copy, or verbal).
- **DFCS will recommend a grade of F for any cadet who egregiously violates this Help Policy or contributes to a violation by others. Allowing another cadet to see your assignment to help them will result in a zero on this assignment.**

Documentation Policy

- You must document all help received from sources other than your instructor or instructor-provided course materials (including your textbook).
- The documentation statement must explicitly describe WHAT assistance was provided, WHERE on the assignment the assistance was provided, and WHO provided the assistance.
- If no help was received on this assignment, the documentation statement must state "NONE."
- Vague documentation statements must be corrected before the assignment will be graded and will result in a grade deduction equal to 5% (ceiling) of the total possible points.

OBJECTIVES

1. Demonstrate proficiency solving problems using the C programming language.
2. Be able to manage a C programming project with `CLion/CMake`.

OVERVIEW

This short PEX focuses on implementing various basic aspects of the C programming language using the CLion Integrated Development Environment. This will ensure you have a functional IDE, as well as providing feedback to your instructor about your C proficiency.

You must write your program on your Ubuntu virtual machine using CLion/CMake. Do NOT code in Windows or any other IDE.

IMPLEMENTATION

Implement the following. Y**our outputs must be in the correct order for full points.**

a. Create a standard C program called pex0.c with the ability to pass command line arguments into main().
b. Print to standard out (i.e. the console) the first command line parameter passed into main (e.g., argv[1]).
c. Using a **for** loop, print to standard out all numbers from 1 to 20 that are divisible by 3.
d. Create a function called **func1**.
- **func1** shall accept a single parameter, an integer array
- **func1** shall return the first element of the integer array parameter multiplied by 2 (note: this should be an integer)

e. Add an integer array variable to your program and set the first element to 5. Pass that array to **func1** and print the returned value to standard out.
f. Create a C header file called pex0.h
- define a struct called **`node`** with the following members:
  - a char array called **`name`** of size 64
  - a pointer called **`next`** that can point to another **`node`** struct. We can use this to set up a linked list of **nodes**.

g. Add a variable of type **node** to your program and call it node1. Using strncpy, set the member variable **name** of node1 to the string “Node 1”.
h. Add a variable of type **node** to your program and call it node2. Using strncpy, set the member variable **name** of node2 to the string “Node 2”.
i. Set the member variable **next** of node1 to point to node2.
j. Set the member variable **next** of node2 to NULL.
h. Print to standard out the member variable **name** of node1.
j. Print to standard out the member variable **name** of node2. You must not access the node2 variable directly, you must reference it using the member variable **next** of node1.
k. Add a new variable to your program: a pointer to a **node** struct and call it node3. Create a dynamically allocated **node** variable using malloc() and use the node3 pointer to store its reference.
l. Using strncpy, set the member variable name of node3 to the string “Node 3”.
m. Set the member variable next of node3 to NULL.
n. Print to standard out the member variable **name** of node3.

SUBMISSION INSTRUCTIONS

Submit your PEX via the PEX0 page on Moodle. You will compress (zip) your project folder and upload the zip archive. Be sure your zip includes your pex0.c and pex0.h files and your `CMakelists` file (if you correctly zip your project folder you should be all set).

Moodle documentation has been disabled for this assignment. All of your documentation must be in the file containing your `main()`function as part of the header comment (the template header has a place for you to put your documentation). You will not have Moodle to enforce the quality of documentation, so be sure you are thorough in your documentation of WHAT assistance was provided, WHERE on the assignment the assistance was provided, and WHO provided the assistance.

Score Sheet

Requirements	
a. Design/Implementation/Coding Standards (i.e. is your code readable)	**+1**
b. Proper command line argument referencing	**+1**
c. Correct usage of control structures (if, for, etc.)	**+1**
d. Correct use of built-in C functions (printf, strncpy, etc.)	**+1**
e. Proper function creation and use	**+1**
f. Correct use of variables	**+1**
g. Correct struct definition and manipulation/use	**+2**
h. Correct usage of pointers	**+2**
i. Correct usage of dynamic memory	**+2**
Penalties	
c. Vague/Missing Documentation (5%)	**-0/1**
d. Late Submission (25% cap/day)	
Total	**12**