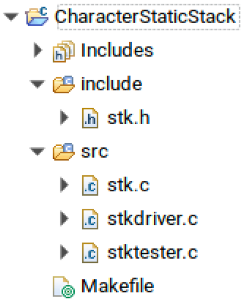
Assignment #1

Topic(s):	C, Makefiles, Writing modular code, Stack ADT	
Date assigned:	Wednesday, September 4, 2019	9:15 am
Date due:	Monday, September 16, 2019	9:15 am
Points:	30	

The purpose of this assignment is to have you implement a Stack ADT using a static array of characters

Specifically, for this assignment you will:

1) Download and untar /home/CS300Public/2019/01Files/CS300f19_Projects.tar.gz from zeus.
2) Point your Eclipse workspace at **CS300f19_Projects**
3) You are provided a completed stk.h and Makefile. Do not alter stk.h.
4) You must complete stk.c, stktester.c, and stkdriver.c
5) You will alter ONLY stk.c, stktester.c, and stkdriver.c. Only alter Makefile *to alter the tarball target!*



Attached is a simple driver that tests some of your stack functions.

To successfully complete this assignment:

1. Implement each of the functions for stk.h one at a time in a file called stk.c. After you write each function, write a snippet of code in stktester.c that will that the just written function. Part of your grade will be based on how well your driver tests each and every function in stk.h. Note: The driver that I supplied you is not a good example of extensively testing each of your functions. Your driver must have many well thought out assert statements. You will also need loops to write a proper driver.

2. Once you have implemented all functions, you are to write a stkdriver.c. This driver will read a word a character at a time from stdin (keyboard) and determine if the word is a palindrome. Example output for stktester.c and stkdriver.c is shown below. You may either read the word into a character array and process the array, or read the word a character at a time (**`getchar() or getc()`** see section 22.4 of your book for links explaining these functions.). You may use one or two stacks depending on the algorithm you choose.

Error Processing

1. In stk.c, check any parameter pointer for NULL and report an appropriate error message by calling **processError()** for a NULL pointer. For instance, if psStack is NULL in stkCreate, report STK_NO_CREATE_ERROR. Also, check for empty and full where appropriate. There are many other error message you must handle in stk.c.

Assignment Submission

The Makefile for the **CharacterStaticStack** project contains a target *tarball*. You must edit the Makefile and replace PUNETID with your own id. From the command line, you must run the tarball target to produce a tar file (similar to a compressed zip file) so you can submit your work. See the commands below.

You will need to copy the tar file to zeus, ssh to zeus to run commands on zeus, and then use the submit command to submit your file. Remember, we ran the submit command during Assignment Zero, Basic Linux Exercise. Review this exercise if you have trouble with the Linux command line.

The following commands, run from your Virtual Machine, will create, move, and submit the file.

You can only submit ONCE!

```
cd CS300f19_Projects/CharacterStaticStack
make tarball
cd ..
ls -al
scp cs300_1_PUNETID.tar.gz punetid@zeus.cs.pacificu.edu:
ssh punetid@zeus.cs.pacificu.edu
submit cs300f19 cs300_1_PUNETID.tar.gz
ls *.receipt
exit
```

Once you have successfully submitted a file, you will get a receipt which is a file that will end in **`.receipt`**. You can make sure your file was submitted correctly by typing the command:

```
checkReceipt submittedfile classname receiptfile
```

EXAMPLE

```
zeus:~> checkReceipt cs300_1_punetid.tar.gz cs300f19
cs300_1_punetid.tar.gz.cs300f19.receipt
```

will produce the result:

```
HASH>>>k�ו���������`U�y�h
HASH>>>k�ו���������`U�y�h�
SUCCESS! Your receipt is valid
```

DO NOT MOVE OR MODIFY THE SUBMITTED FILE **OR** THE RECEIPT FILE OR THE CHECK RECEIPT COMMAND WILL NOT BE SUCCESSFUL.

Test your project on Zeus before you submit!

After the SSH step and before the submit step above do the following:

```
tar xzf cs300_1_PUNETID.tar.gz
cd CharacterStaticStack
make clean
make
bin/stktester
bin/stkdriver
cd ..
# remove the directory from zeus in case you need to reupload and
# test again.
rm -rf CharacterStaticStack
```

If any of the above steps fail, you must fix your code before submitting!

Submit a color, double-sided, stapled packet of code by that deadline. The packet must be in the following order:

stk.c

stkdriver.c

stktester.c

You are to use the coding guidelines of the coding standards on the CS300 Web page.

Book sections:

4.13, 20, 22.4, 18

Example stkdriver.c output:

```
Enter a word: AA
AA is a Palindrome
Enter a word: aa
aa is a Palindrome
Enter a word: bb
bb is a Palindrome
Enter a word: bc
bc is NOT a Palindrome
Enter a word: abc
abc is NOT a Palindrome
Enter a word: abccba
abccba is a Palindrome
Enter a word: abcba
abcba is a Palindrome
Enter a word: QUIT
```

Example stktester.c output (your testing must be more comphrensive)

```
Program Start

SUCCESS TESTS:
SUCCESS: Loaded Error Messages
SUCCESS: Stack Created
SUCCESS: Stack size is 0
SUCCESS: Stack Terminated

Program End
```