

### Assignment #3

**Topic(s):** C, Makefiles, Writing modular code, Stack ADT, File I/O  
**Date assigned:** Monday, September 16, 2012  
**Date due:** Friday, September 28, 2012  
**Points:** 30

A palindrome is a word or phrase that is spelled the same backwards as it is forwards. A few examples include: mom, racecar, and rotator. You are to write a complete C program using a stack, an external data file, a Makefile, and well-defined modules that will read lines of characters, one line at a time, from a data file and determine if the line of characters read is a palindrome.

For example, consider the following data file:

```
mom
palindrome
racecar
rotator
computer science
Madam I'm Adam
```

The output results from your program must be the following:

```
PALINDROME CHECKER
-----
mom [palindrome]
palindrome [not a palindrome]
racecar [palindrome]
rotator [palindrome]
computer science [not a palindrome]
Madam I'm Adam [palindrome]

PROCESSING COMPLETE!
```

1. You are to break up your program into appropriate .h/.c files and on the day the assignment is due, turn in a colored hard copy of each .h/.c combination (fully documented). You might not have a .h file for every .c file.
2. Your code is to be written in C using Eclipse. Programs written in other environments will not be graded. Name your project **cs300\_stack\_PUNetID**.
3. Submit a file **cs300\_stack\_PUNetID.tar.gz** using the submit script on zeus by 9:15am on the day in which the assignment is due.
4. Use well thought out functions in solving this problem. Don't break code out later into a function
5. Code and test your modules one function at a time. This will lessen your level of frustration greatly.

6. Use V6.0 of the coding standards.
7. Write efficient/clean code.
8. Use the debugger to effectively develop a correct solution

### **Part I (Due Monday, September 24, 2012)**

For the first part of this assignment, you must completely code and test your **stack module**. A stack must be used correctly in your palindrome solution. We will discuss this in detail in class. The following stack header found in file **stk.h** is on zeus in the **CS300Public** folder. You are to use this exact header file without modification for this assignment, so copy the file from the CS300Public folder and use the exact file in your Eclipse.

You are to write a driver called **stackdriver.c** that thoroughly tests each function of your stack module. Be sure to test each error and boundary condition. Make sure your driver outputs results to the display screen so that I can see the results of your testing.

You are to submit a tarball called **cs300\_stack\_PUNetID.tar.gz** that contains your Eclipse project that solves the Part I description.

### **Part II (Due Friday, September 28, 2012 @ 11:59pm)**

You are to complete the remainder of the assignment. To do so,

1. Create a folder called **testfiles** in your project folder. Inside of this folder, create a data file called **palindromes.txt** that contains words or phrases, one per line. Use the following data file:

```
mom
palindrome
racecar
rotator
computer science
Madam I'm Adam
```

2. Write any remaining modules to solve the above problem. Minimally, you should have at least three modules: (a) a stack module, (b) a palindrome module, and (c) a driver module. Your Makefile is to produce a driver called **palindromeDetector**. This Makefile still needs to produce **stackDriver**.

**Note: The following are assumptions for this assignment:**

- 1) Print out each line from the data file exactly as the line appears in the data file followed by a single space and then either [palindrome] or [not palindrome] depending on whether the line is or is not a palindrome.

2) For this assignment, a palindrome is defined as a string of characters where all non-alphabetic characters are ignored. Further, upper-case and lower-case versions of the same letter are considered equal. Alphabetic characters are 'A' to 'Z' or 'a' to 'z'.

3) Make sure your output results look exactly as those shown on page 1 above.

```

/*****
File name:      stk.h
Author:        Computer Science, Pacific University
Date:         9/16/12
Class:        CS300
Assignment:    Determine Palindromes using a Stack
Purpose:      To define the header file for the stack module.
*****/

#ifndef STK_H_
#define STK_H_

#define MAX_STACK 1024

#define TRUE 1
#define FALSE 0

#define NO_ERROR 0
#define ERROR_STACK_EMPTY 1
#define ERROR_STACK_FULL 2
#define ERROR_NO_STACK_CREATE 3

// User-defined datatypes for easier reading

typedef short int BOOLEAN;
typedef short int ERRORCODE;
typedef char DATATYPE;

typedef struct Stack
{
    int top;
    DATATYPE data[MAX_STACK];
} Stack;

BOOLEAN stkCreate (Stack *);
BOOLEAN stkTerminate (Stack *);
BOOLEAN stkIsFull (Stack);
BOOLEAN stkIsEmpty (Stack);
BOOLEAN stkPush (Stack *, DATATYPE);
BOOLEAN stkPop (Stack *, DATATYPE *);
BOOLEAN stkPeek (Stack, DATATYPE *);

#endif /*STK_H_*/

```

Here is what my Makefile looks like. From this Makefile you can see that I have created two modules to solve this problem. If you find a need for more, feel free to create more. This Makefile will also be placed in CS300Public on zeus.

```
CC=gcc
CFLAGS=-g -Wall
TARGETS=palindromeDetector stackDriver

.PHONY: all clean dist

all: ${TARGETS}

stackDriver: bin/stackDriver.o bin/stk.o
    ${CC} ${CFLAGS} -o stackDriver bin/stackDriver.o bin/stk.o

bin/stackDriver.o: src/stackDriver.c include/stk.h
    ${CC} ${CFLAGS} -o bin/stackDriver.o -c src/stackDriver.c

bin/palindromeDetector: bin/palindromeDriver.o bin/stk.o
    ${CC} ${CFLAGS} -o palindromeDetector bin/palindromeDriver.o bin/stk.o

bin/palindromeDriver.o: src/palindromeDriver.c include/stk.h
    ${CC} ${CFLAGS} -o bin/palindromeDriver.o -c src/palindromeDriver.c

bin/stk.o: src/stk.c include/stk.h
    ${CC} ${CFLAGS} -o bin/stk.o -c src/stk.c

clean:
    rm ${TARGETS} bin/*.o

dist: clean tar czf ../cs300_stack_PUNetID.tar.gz ../cs300_stack_PUNetID
```