Stacks

Date assigned: Friday, September 17, 2010
Date due: Monday, September 27, 2010
Points: 30

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

For example, consider the following datafile:

```
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!
```

Part I (Due Tuesday, September 21, 2010 by 5pm)

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 will be placed 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 project 03punetid.

```
/*****************************************************************************
 File name:      stk.h
 Author:         Computer Science, Pacific University
 Date:           9/17/10
 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 EMPTY_STACK -1
#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_*/
```

You are to write a driver called stackdriver.c that tests each function of your stack module. Make sure your driver outputs results to the display screen so that I can see the results of your testing.

By 5pm on the date listed in Part I above, you are to submit a tarball called 03punetid.tar.gz that solves the Part I description. Further, you MUST use subversion for this part of the assignment and have at least five commits total with meaningful commit messages ranging over at least two different days. You must show me your commit history before 5pm of the Part I due date. Any time you have at least five commits over two different days, you can show me your commit history. You don’t have to wait until the last minute to do so. I doubt that I will be in my office at 5pm when Part I is due.

Part II (Due Monday, September 27, 2010 by 9:15am)

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 datafile called palindromes.txt that contains words or phrases, one per line. Use the following datafile:

```
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 palindromedriver.

Note: The following are assumptions for this assignment:

1) Print out each line from the datafile exactly as the line appears in the datafile 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.

As always,

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).
2. Your code is to be written in C using Eclipse 3.6. Programs written in other environments will not be graded. Submit a project folder called 03punetid.tar.gz

using the submit script on zeus. Make sure to completely test your solution before submitting your final solution.

3. Test your modules one function at a time. This will lessen your level of frustration greatly.
4. You are to use the coding guidelines from V6.0 of the coding standards.
5. You must use your own makefile for this assignment so make sure that you uncheck the Automatically generate makefiles option when you create the project.

Goals for this assignment:

1. Break your program up into well thought out modules.
2. Use well thought out functions in solving this problem. Don't break code out later into a function.
3. Code and test your program one function at a time.
4. Implement the stack data structure efficiently in solving this problem
5. Write efficient/clean code
6. Use the debugger to effectively develop a correct solution