

Assignment #3

Topic(s): C, Makefiles, Writing modular code, Stack ADT, Dynamic Memory
Date assigned: Wednesday, September 23, 2015
Date due: Friday, October 2, 2015
Points: 25

We have discussed the Stack ADT and implemented the Stack ADT statically. The purpose of this assignment is to have you implement the Stack ADT dynamically. Specifically, create a project called DynamicStack with an include file, stk.h, and source files stk.c and stkdriver.c. A copy of stk.h is shown below and also exists on zeus in /home/CS300Public/2015/03Files. You are to copy stk.h from zeus and implement each function prototype in stk.h in a file called stk.c. Do not modify stk.h in any way.

```
1 /*****
2 File name:  stk.h
3 Author:    Computer Science, Pacific University
4 Date:     9/22/2015
5 Class:    CS300
6 Assignment: Dynamic Stack
7 Purpose:   Header for a dynamic stack
8 *****/
9
10 #ifndef STK_H_
11 #define STK_H_
12
13 #include <stdbool.h>
14
15 //*****
16 // Constants
17 //*****
18 #define NO_ERROR 0
19 #define ERROR_STACK_EMPTY 1
20 #define ERROR_STACK_FULL 2
21 #define ERROR_NO_STACK_CREATE 3
22 #define ERROR_NO_STACK_TERMINATE 4
23 #define ERROR_NO_STACK_MEMORY 5
24
25 //*****
26 // User-defined types
27 //*****
28 typedef short int ERRORCODE;
29 typedef char DATATYPE;
30
31 typedef struct StackElement *StackElementPtr;
32 typedef struct StackElement
33 {
34     DATATYPE data;
35     StackElementPtr psNext;
36 } StackElement;
37
38 typedef struct Stack *StackPtr;
39 typedef struct Stack
40 {
41     int size;
42     StackElementPtr psTop;
43 } Stack;
44
45 //*****
46 // Function prototypes
47 //*****
48 extern ERRORCODE stkCreate (StackPtr psStack);
49 extern ERRORCODE stkTerminate (StackPtr psStack);
50 extern ERRORCODE stkIsFull (const StackPtr psStack, bool *pbIsFull);
51 extern ERRORCODE stkIsEmpty (const StackPtr psStack, bool *pbIsEmpty);
52 extern ERRORCODE stkPush (StackPtr psStack, DATATYPE value);
53 extern ERRORCODE stkPop (StackPtr psStack, DATATYPE *pValue);
54 extern ERRORCODE stkPeek (const StackPtr psStack, DATATYPE *pValue);
55 extern ERRORCODE stkSize (const StackPtr psStack, int *pSize);
56
57 #endif /* STK_H_ */
58
```

If you've done this assignment correctly, you should be able to take your static stack driver and use it as a driver for the dynamic stack. No other changes should be necessary. You should also be able to make 1 change to the #include of palindromeChecker.c and change the Makefile of palindrome checker to refer to DynamicStack instead of StaticStack. No other changes are necessary. Now how cool is that!!!! ☺

To successfully complete this assignment:

1. Create a new project called DynamicStack. Make sure that you create a new C project->Makefile project->Empty Project->Linux GCC. Add the directories (src, include, bin).
2. Implement each of the functions for stk.h one at a time in a file called stk.c. Note that stkIsFull will always return false unless NULL pointers are passed into it.
3. Create a makefile for the project DynamicStack. Watch the Makefile videos and look at the Makefile for StaticStack to help you create the project.
4. Once you have implemented each function, you are to write a driver (stkdriver.c) that extensively tests all of the functions in your program. Part of your grade will be based on how well your driver tests each and every function listed above. Note: The driver that I supplied you in the previous assignment is not good for testing your project.
5. You are to submit a tarball called **cs300_3_PUNetID.tar.gz** that contains your Eclipse projects DynamicStack, StaticStack, and PalindromeChecker after extensive testing on zeus.
6. List how many hours you worked on the assignment in the header comments.

If you find any mistakes or you think there are discrepancies, please email me ASAP. I will check into your issue, fix as necessary, and email the entire class if changes are made.