CS12 SI LAB6

If you did not finish lab 5 last week, go back and finish it first.

Problem 1: Write a stack class:
Write a class called “myStack” that will function as a stack in the typical LIFO manner and dynamically allocates its memory. The stack should change the size of its allocated memory during run time to both minimize the memory used and to ensure that there is no limit to the number of values pushed onto the stack. The required member functions for this class are:
void push(char c);
char pop();
It is important to also include a constructor and destructor to manage your dynamic memory. Make sure to write the declaration in a .h file and your implementation in a .cpp file

// Name your files for this problem *username_myStack.h* and *username_myStack.cpp*

Problem 2: Write a program that tests for correctly matching braces and parentheses:
You will now write a test program that will use your stack class, take a file as a command line argument and return whether the file has matching closing braces and parentheses.
Example:

(({dog}){})	would return true
((()}}cat)	would return false

Your program should output a message indicating the results.

// Name your .cpp file for this problem *username_main.cpp*

Problem 3: Makefile!!
Write a makefile named Makefile that compiles your program. It should first compile a .o for the stack class and then compile the main program linking in the -o file. Also add a clean target so that you don't get your workspace all cluttered.

Submission:
To turn in this lab first make sure that all of your .cpp files are in the same directory. Then run

```
tar -czvf username_lab5.tar username* Makefile
```

This should create a file called *username_lab4.tar* which you should upload to the given link.
https://drive.google.com/folderview?id=0B9ModvlYGFFEd29Qb195OFY1SGc&usp=sharing