**Lab2 - Stacks due February 3**

Purpose:  Use a stack   
Background:  There are 3 main techniques for evaluating mathematical expressions:

\* prefix   
\* infix   
\* postfix

Most math classes teach equation evaluation using the in-fix technique.  Some calculators use post-fix evaluations, which internally use a stack.  Here is an example:  
 Infix:   (3+4)\*5    
 Postfix: 34+5\*  

This labs purpose is to use a stack to evaluate an expression.  The technique:  
1. Read in a string that has an expression using post-fix notation   
2. For each character in the string:    
3. If it is a number, push it on the number stack    
4. If it is an operator (like +, -, \*, /),

a. pop two numbers off the stack     
b. perform the operation     
c. push the result on the stack

Implement a link-list based stack. Read in the Posfix.csv in the documents folder for this week for a list of post-fix expressions to evaluate.  If there is an error in the expressions, your program should note the error and then move to the next expression.

Extra credit: convert infix to postfix, postfix to infix