**Half-Problem Set 2: Stacks**

For this assignment, you will be designing and implementing two Python programs that use the Stack structure to solve the given problem. You are to use the lliststack.py module from the textbook.

**Postfix Expressions.** Write a program (postfix.py) that extracts a postfix expression from the user and evaluates it as described in class. Your program should extract the expression from the user via standard input. The expression should be formatted such that there is at least one blank space between each operator and operand. For example,

Enter a postfix expression: 5 6 \* 13 2 \* +

Assume all operators and operands are valid. The operands should be integer values while the valid operators include

+ - / \* %

After extracting the expression, use the split() method on the string to split it into individual tokens that are stored in a Python list. Note that a multi-digit operand (i.e. 13) is considered a single token.

The postfix expression should be printed to standard output, one token at a time, as it is being evaluated. In addition, if the expression is valid, the result should be printed on the next line following the expression as shown below

5 6 \* 13 2 \* +

= 56

If the expression is invalid, stop evaluating the expression once it is determined there is an error and print 3 dollar signs. then print an error message on the next line to indicate the error. For example, if there are too many operators for the given operands encountered, your program should produce the results shown below

Enter a postfix expression: 17 6 / + 8 \*  
17 6 / + $$$  
Error: insufficient number of operands.

When there are too few operators for the given operands, your program should produce the results shown below

Enter a postfix expression: 17 6 / 25 5 +  
17 6 / 25 5 + $$$  
Error: insufficient number of operators.

**Palindromes.** Write a program (palindrome.py) that extracts a string from the user and uses one or more stacks to determine if the string forms a palindrome. A palindrome is a string that reads the same from either front to back or back to front. For example, the following strings are all palindromes

aa  
abcdcba  
xy zz yx

After extracting the string and determining if it is a palindrome, your program should output the original string followed by a message on the next line that indicates if the string is a palindrome.

For example, if the user enters

abcdcba

your program should print

abcdcba  
palindrome

On the other hand, if the user enters

xyzxyz

your program should print

xyzxyz  
not a palindrome

Your solution must also meet the following requirements:

* You may not create or use any sequences other than the string returned by the input() function.
* You cannot use the len() function to determine the length of the string nor can you use subscripts to access the individual characters within the string.

**Requirements.** Your implementations should be efficient and the Python code should well-structured and commented appropriately.