CPE 112 Introduction to Data Structure
Laboratory Exercise 3

Objective

This lab gives you practice creating a linked stack and its application.

Instructions

Create a program for converting an infix into a postfix and prefix expression. This program applies a linked stack for storing the operators in the infix expression. Then, these operators will be popped to represent a postfix expression. Furthermore, the algorithm to convert an infix notation to postfix notation can be applied for converting prefix notation.

1. Implement the linked stack. You need to define a stack structure (the linked list node). The **push** and **pop** function must be created.
   - The **push** function will add the new node next to the header of the linked list.
   - The **pop** function will remove the first node from the linked list.
   - The **peek** function can be implemented for retrieving the data in the first node of the linked list. It is an optional function. If your pop function returns the data of popped node, you may not create this function.

2. Declare the string variables for infix, postfix, prefix expression. Received the infix expression from the keyboard.

3. Create the function for converting infix notation to postfix notation. The result of conversion must be stored in the postfix variable.

4. Create the reverse function. This function will reverse the infix string. Moreover, it must interchange left and right parentheses.

5. Apply the reverse and postfix conversion for converting infix notation to prefix notation. The result of conversion must be stored in the prefix variable.

6. Show the results of conversions. The first line is postfix notation and the second line is prefix notation.

7. Upload your C source file to the **Grader** and **LEB2** for evaluating your score.

Important Note

1. Any global variables are not allowed. (If global variables are used, your score will be reduced by 20%).

2. Stack must be implemented by using a linked list. (If you implement stack by using an array, your score will be reduced by 50%)

Example

Input	Output
A-(B/C+(D%E*F)/G)*H	ABC/DE%F*G/+H*- -A*+/BC/%D*EFGH