CMP 210 – Data Structures and Algorithms
Spring 2021
Assignment - 01

Issue Date: **Monday – March 22nd, 2021**
Submission Deadline: **Thursday – March 25th, 2021 (Till 12:00 am)**

Instructions!

1. You are required to do this assignment on your own. Absolutely **NO** collaboration is allowed, if you face any difficulty feel free to discuss with me.
2. Cheating will result in a **ZERO** for the assignment. (Finding solutions online is cheating, copying someone else's solution is cheating). Also, do not hand your work over to another student to read/copy. If you allow anyone to copy your work, in part or in whole, you are liable as well.
3. Hard **DEADLINE** of this assignment is **Thursday, March 25th, 2021**. No late submissions will be accepted after due date and time so manage emergencies beforehand.

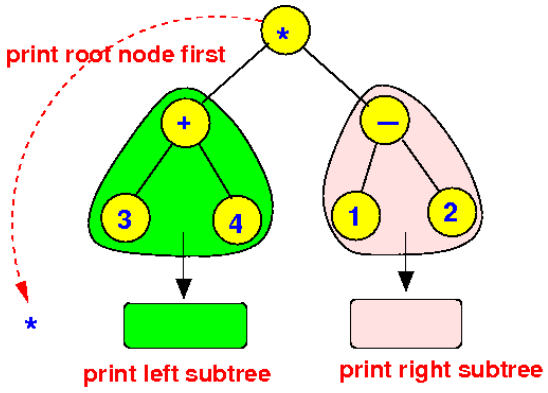
Expression Evaluation: [30 Marks]

A mathematical expression is made up of operands, operators and delimiters. We have several ways of writing mathematical expressions. Infix, postfix and prefix notations are three different but equivalent ways of writing expressions.

Infix notation is the usual way we write expressions, in which operators are written in between the operands.

$$A \times (B + C)/D$$

Infix notation needs extra information to make the order of evaluation of the operators clear, for example rules built into the language about operator precedence and associativity, and brackets ( ) to allow users to override these rules. For example, the usual rules for associativity say that we perform operations from left to right, so the multiplication by A is assumed to come before the division by D. Similarly, the usual rules for precedence say that we perform multiplication and division before we perform addition and subtraction.



Postfix notation (also known as "Reverse Polish Notation") is the way of writing expressions in which operators are written after their operands. The order of evaluation of operators is always left-to-right, and brackets cannot be used to change this order. The infix expression given above is equivalent to

$$ABC + \times D/$$

We have already discussed Infix to postfix conversion in the class. Now, you are required to implement a convertor to convert a given expression from infix to postfix. After converting the expressions also evaluate the expression and return result.

Implement following functions:

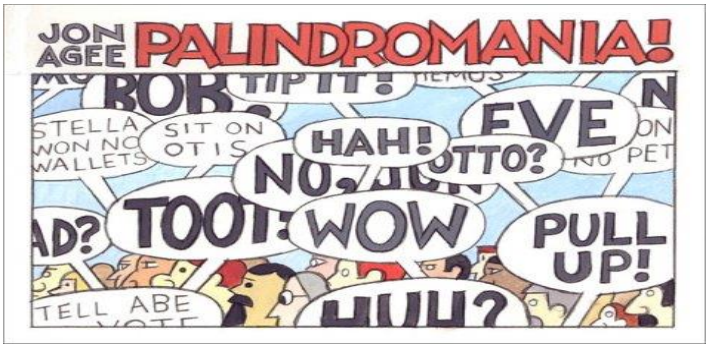
1. **isBalanced (expression):** This function should check whether the given infix expression is correct and balanced or not. We will consider an expression correct if the parenthesis are correctly added and are balanced. For example: $(a \times \{b + c\} - 4/x + [e - 5])$ is a correct expressions. While, $(5 + \{6 \times) - 2\}$ and $\{5 + z)$ are examples of incorrect expressions.
2. **infixToPostfix (expression):** This function should take an infix expression as input and returns equivalent postfix expression.
3. **Evaluate (expression, key):** This function should take either a prefix expression or a postfix expression as first argument and takes a Boolean key as second argument (0 for prefix expression, 1 for postfix expression) and returns the value of the expression after evaluation.

Also write a proper main program providing menu to make it easy to test your functions. No marks shall be given without this driver program.

Palindromania! [10 Marks]

A palindrome is a word, phrase, number or other sequence of units that has the property of reading the same in either direction. Write a program that could determine whether the given string is a palindrome.

Examples:



Palindrome Words:

- Dad
- 1221
- Racecar
- Rotator
- Level
- Civic

Palindrome phrases:

- Too bad--I hid a boot.
- Do geese see God?
- "Go Hang a Salami! I'm a Lasagna Hog!"
  (title of a book on palindromes by Jon Agee, 1991)
  Note: Handle spaces and punctuation marks carefully.

Good Luck!