ÇANKAYA UNIVERSITY

SENG 201 Data and Game Structures

Lab Assignment 5

Prefix, infix, and postfix are three different notations used to represent mathematical operations, such as addition, subtraction, multiplication, and division. They describe how the operators and operands (numbers) are arranged in an expression. Each notation has its advantages and is used in various contexts.

1. **Infix Notation:**

   Infix notation is the most common and familiar way of writing mathematical expressions. In this notation, operators are placed between the operands. For example:

   - `( 2 + 3 ) * ( 10 – 5 )`

2. **Prefix Notation (also known as Polish Notation):**

   In prefix notation, the operator comes before the operands. This notation eliminates the need for parentheses to specify the order of operations because it's unambiguous. This notation is efficient for computers and is used in some programming languages, like Lisp and Scheme. For example:

   - `* + 2 3 – 10 5`

3. **Postfix Notation:**

   In postfix notation, the operator comes after the operands. Like prefix notation, postfix notation eliminates the need for parentheses and is unambiguous. For example:

   - `2 3 + 10 5 - *`

ÇANKAYA UNIVERSITY

PART 1

Write a method "prefixToInfix" that takes a formula written in prefix form and prints the infix form of the result. Assume that there will be only 4 operations (+,-,*,/) and single digit numbers in the input. Here are some examples:

- prefixToInfix("+*342") → prints ((3*4)+2)
- prefixToInfix("/+*3422") → prints (((3*4)+2)/2)
- prefixToInfix("+-*+-*+-*1234567891")
  → prints (((((((((1*2)-3)+4)*5)-6)+7)*8)-9)+1)

Use the following template to start writing your method:

```
public static String prefixToInfix ( String formula ) {
      // write your code here
      // hint: make use of stack!
}
```

PART 2

Write a method "postfixToInfix" that takes a formula written in postfix form and prints the infix form of the result. Assume that there will be only 4 operations (+,-,*,/) and single digit numbers in the input. Here are some examples:

- postfixToInfix ("34*2+") → prints ((3*4)+2)
- postfixToInfix ("34+22*/") → prints ((3+4)/(2*2))
- postfixToInfix (1234567891+-*+-*+-*")
  → prints (1*(2-(3+(4*(5-(6+(7*(8-(9+1)))))))))

Use the following template to start writing your method:

```
public static String postfixToInfix ( String formula ) {
      // write your code here
      // hint: make use of stack!
}
```

Your code should handle inputs with **any length**!

You should submit one zip file name as **"YourNameSurname_Lab5.zip"** and it should contain the java files you created!