If you areinterested inconverting postfix expressions to infix ones, below you can see an algorithm for doing it.

**Algorithm Postfix to Infix**

While (there are input symbols left) {

Read the next symbol from input.

If the symbol is an operand, push it onto the stack.

Otherwise (the symbol is an operator) {

If there are fewer than 2 values on the stack, show error

/\* input not enough values in the expression \*/

Else

a. Pop the top 2 values from the stack.  
             b. Put the operator, with the values as arguments and form a string.  
             c. Encapsulate the resulted string with parenthesis.  
             d. Push the resulted string back to stack.  
  
 }

} // end while

If there is only one value in the stack, that value is the desired infix string. Pop it.

Otherwise, show error

/\* The user input has too many values \*/

1. Create a class PostfixToInfix with the static method *convert*.
2. The simple translation of the provided algorithm inserts too many parentheses. Enhance the method to leave only needed ones.
3. Include the file InfixToPostfix,java in your package and in the test driver read the lines from a text file containing infix expressions; for each line you
4. Output it to the console;
5. Output the postfix form;
6. Output the conversion back to infix.
7. The text file would contain the following lines:

|  |
| --- |
| a + b - c  a + b \* c  a^b^d  (a - b)\*(c- d\*3)  2 - 6/a\*(b\*b+c-2)  1/(a \* b - c^d)  a^(b\*c)  a^(b^c)  (a^b)^c |

You can add more lines. Submit this file along with others.