Infix to Postfix Conversion Algorithm

1. Add `)` to the end of the infix expression.
2. Push `(` onto the stack.
3. Repeat until each character in the infix notation is scanned:
    - **IF** a `(` is encountered, push it on the stack.
    - **IF** an operand (whether a digit or a character) is encountered, add it to the postfix expression.
    - **IF** a `)` is encountered, then:
        - a. Repeatedly pop from the stack and add to the postfix expression until a `(` is encountered.
        - b. Discard the `(`. Remove it from the stack and do not add it to the postfix expression.
    - **IF** an operator `O` is encountered, then:
        - a. Repeatedly pop from the stack and add each operator (popped from the stack) to the postfix expression which has 'equal or higher' precedence than `O`.
        - b. Push the operator `O` to the stack.
4. Repeatedly pop from the stack and add it to the postfix expression until the stack is empty.
5. EXIT.

Infix to Prefix Conversion Algorithm

1. Reverse the infix string. Note that while reversing the string, you must interchange left and right parentheses.
2. Obtain the postfix expression of the reversed expression from Step 1 using the postfix algorithm above, with a slight change in Step 3.4:
    - **IF** an operator `O` is encountered, then:
        - a. Repeatedly pop from the stack and add each operator (popped from the stack) to the postfix expression which has 'higher' precedence than `O`.
3. Reverse the postfix expression to get the prefix expression.