```java
import components.queue.Queue;
import components.simplewriter.SimpleWriter;
import components.simplewriter.SimpleWriter1L;

/**
 * HW29
 *
 * @author Sam Espanioly
 */
public final class HW29 {

    /**
     * Default constructor--private to prevent instantiation.
     */
    private HW29() {
        // no code needed here
    }

    /**
     * Evaluates a Boolean expression and returns its value.
     *
     * @param tokens
     *            the {@code Queue<String>} that starts with a bool-expr string
     * @return value of the expression
     * @updates tokens
     * @requires [a bool-expr string is a prefix of tokens]
     * @ensures <pre>
     * valueOfBoolExpr =
     *   [value of longest bool-expr string at start of #tokens]  and
     * #tokens = [longest bool-expr string at start of #tokens] * tokens
     * </pre>
     */
    public static boolean valueOfBoolExpr(Queue<String> tokens) {
        boolean check = false;
        if (tokens.length() > 0
                && !tokens.front().equals("### END OF INPUT ###")) {
            String token = tokens.dequeue();
            if (token.equals("T")) {
                check = true;
            } else if (token.equals("F")) {
                check = false;
            } else if (token.equals("NOT")) {
                tokens.dequeue(); // for (
                check = valueOfBoolExpr(tokens);
                tokens.dequeue(); // for )
            }
            if (token.equals("(")) {
                check = valueOfBoolExpr(tokens);
                token = tokens.dequeue();
                if (token.equals("AND")) {
                    check = check && valueOfBoolExpr(tokens);
                } else { // OR
                    check = check || valueOfBoolExpr(tokens);
                }
                tokens.dequeue(); // for )
            }
        }
        return check;
    }

    /**
     * Main method.
     *
     * @param args
     *            the command line arguments; unused here
     */
    public static void main(String[] args) {
        SimpleWriter out = new SimpleWriter1L();
        //out.println("Hello World!");
        out.close();
    }

}

```