

```

public class CuteScanner {
    public enum TokenType {
        //괄호안에 TokenType에 맞는 state number 입력
        INT(1), ID(4), MINUS(2), PLUS(3), L_PAREN(5), R_PAREN(6), TRUE(8),
FALSE(
        9), TIMES(10), DIV(11), LT(12), GT(13), EQ(14),
APOSTROPHE(15), QUESTION(
        16), // bug

        // not present in automata
        DEFINE(-1), LAMBDA(-1), COND(-1), QUOTE(-1), NOT(-1), CAR(-1), CDR(-
1), CONS(
        -1),

        ATOM_Q(-1), NULL_Q(-1), EQ_Q(-1);

        //.....이하생략

    private void init_TM() {
        //.....

        // '(', ')', '+', '-', '*', '/', '<', '=', '>', '\\', '#T', '#F', #에
대해서
        // 작동하도록 작성
        transM[4]['?'] = 16; // null?, atom?...

        transM[0]['-'] = 2; // '-'
        transM[0]['+'] = 3; // '+'
        transM[0]['('] = 5; // '('
        transM[0][')'] = 6; // ')'
        transM[0]['#'] = 7; // #
        transM[7]['T'] = 8; // #T
        transM[7]['F'] = 9; // #F
        transM[0]['*'] = 10; // *
        transM[0]['/'] = 11; // /
        transM[0]['<'] = 12; // <
        transM[0]['>'] = 13; // >
        transM[0]['='] = 14; // =
        transM[0]['\\'] = 15; // '

    }

    private Token nextToken() {
        //.....

        for (int i = 0; i < temp.length(); i++) {
            StateNew = transM[StateOld][temp.charAt(i)]; // 입력문자로
새로운 상태 판별

            if (StateNew == -1) {// 입력된 문자의 상태가 reject 이므로
에러메세지 출력후 return함
                System.out.println(String
                    .format("acceptState error %s\\n", temp));
                return null;
            }

            StateOld = StateNew;
        }
    }
}

```

```
        //.....
    }

    public List<Token> tokenize() {
        // Token 리스트반환
        List<Token> tokens = new ArrayList<Token>();
        Token t = null;

        while (true) {
            t = nextToken();

            if (t == null)
                break;

            tokens.add(t);
        }

        return tokens;
    }
}
```