LAB SHEET 3

Implement a Recursive Descent Parser for the following Grammars.

1. S->cAd

   A->ab|d

2. S->bAc

   A->re|r

3. S ->a|(L)
   L->L,S|S

4. E->E+T|T
   T->T*F|F
   F->(E) |id

The following conventions are used to specify the grammar rules.

i) The Non-Terminals are denoted by upper-case strings.
ii) The Terminals are denoted by lower-case strings.

IMPLEMENTATION GUIDELINES

1. For every non-terminal, you need to implement a method by the same name. i.e. S (), A (), L () etc. which returns a Boolean value.
2. Terminals can be checked by directly comparing the next token with the expected token. The method getToken () should be implemented fetch the next token.

3. TESTING:
   Given an input, the recursive descent program should output "ACCEPTED" if the input adheres to the grammar spec. If not, print "REJECTED". For example, if the input string is: id * id + id for question no 4; the program should return ACCEPTED since the input adheres to the grammar rules. On the contrary, if the input string is id * (id + id; - missing right parenthesis - the output should be REJECTED.