Language for propositional language

In this homework you are to implement lexer and parser for a simple language.

Token definitions:

ID = [A-Z]+
LAPR = (
RPAR =)
NOT = !
AND = /\
OR = \/
IMPLIES = ‘=>’
IFF = ‘<=>’

Grammar:

propositions → *proposition more-proposition*
more-proposition → , *propositions* | ε
proposition → *atomic* | *compound*
atomic → 0 | 1 | ID
compound → *atomic connective proposition* | *LPAR proposition RPAR* | *NOT proposition*
connective → AND | OR | IMPLIES | IFF

The start variable is *propositions.*

Your implementation should receive a file, if the input file is a valid program, it should print the parse tree in prefix order, otherwise it should return “Syntax Error” message along with the line and column numbers of the first error.

Instructions:

- Use Python 2.7 for implementation
- Deadline: Feb 11 at 9 AM