EXERCISES ON BNF, EBNF, AND PARSE TREES

1. Consider the following grammar:

<exp> ::= <atom> | <list>
<atom> ::= <digit> | <string>
<list> ::= (<expr-list>)
<expr-list> ::= <exp> <expr-list> | ε
<digit> ::= 0 | 1 | 2 | ... | 9
<string> ::= a | b | c

   What change would you make to transform it into EBNF?

2. Given the following grammar with tokens **n**, **l**, **o**, **c**, and <G> as the start symbol:

<G> ::= <A> | <L>
<A> ::= **n** | **l**
<L> ::= **o** <S> **c**
<S> ::= <S> <G> | <G>

   Convert it to EBNF.

3. Write a BNF description for an identifier sequence which is a sequence of identifiers separated by commas. Assume that identifiers are denoted by the token id. In addition, translate your BNF into EBNF.

4. You are given the following grammar, with tokens **a**, **b**, **c**, **d**, and <S> as the start symbol:

<S> ::= **a** <L> **b** | **c**
<L> ::= <L> **d** <S> | <S>

   a. Give a leftmost derivation for the string **acdc.**
   b. Give a rightmost derivation for the same string.
   c. Draw a parse tree for that string.

5. You are given the following grammar, with tokens **a**, **b**, and <S>, <A>, and <B> as the non-terminals <S> as the start symbol:

<S> ::= <A> <B> | ε
<A> ::= **a** <B>
<B> ::= <S> **b**

   a. Give a leftmost derivation for the string **abb**. You may use S, A, and B instead of <S>, <A>, and <B> respectively.
   b. Give a rightmost derivation for the same string.

5. You are given the following grammar, with tokens **a**, **b**, and <S> as the start symbol:

<S> ::= **a** <A> <B>
<A> ::= **b** <B> **b**
<B> ::= <A> | ε

   a. Give a leftmost derivation for the string **abbb**
   b. Give a rightmost derivation for the same string
   c. Draw a parse tree for that string.