1. Write an unambiguous grammar for each of the following languages:

   (a) Statement blocks in ML where semicolons separate statements. For instance:
   `((s;s) ; (s;s;s) ; s) ; (s;s)`

   (b) Statement blocks in C where semicolons terminate the statements. For instance:
   `{{s;} {s; s;} s;} s; {s; s;}`

2. Consider the following grammar:

$$
\begin{array}{lcl}
S & \rightarrow & u\ B\ D\ z \\
B & \rightarrow & B\ v \mid w \\
D & \rightarrow & E\ F \\
E & \rightarrow & y \mid \varepsilon \\
F & \rightarrow & x \mid \varepsilon
\end{array}
$$

   (a) Calculate nullable, FIRST, and FOLLOW sets.

   (b) Construct the LL(1) parsing table and give evidence that this grammar is not LL(1).

   (c) Give an LL(1) grammar which accepts the same language and build the LL(1) parsing table for that grammar.

3. Consider the following grammar for expressions, consisting of array accesses, addition, and assignments:

$$E \rightarrow E\ [\ E\ ] \mid E + E \mid E = E \mid (E) \mid \mathsf{id}$$

   This is an ambiguous grammar. We would like to write an unambiguous grammar for the same language, such that array accesses have higher precedence than assignments, and both array accesses and assignments have higher precedence than addition.

   (a) Write an LL(1) grammar which accepts the same language and has the desired operator precedence. Show the LL(1) parsing table for this grammar.

   (b) Write an LR(1) grammar which accepts the same language, respects the desired operator precedence, and is such that addition is left-associative, and assignments are right-associative.

   (c) Write the parse tree for the expression `a+b=c[d]=e+f`, using the LR(1) grammar.

4. Consider the following grammar:

$$E \rightarrow \mathsf{id} \mid \mathsf{id}\ (\ E\ ) \mid E + \mathsf{id}$$

   (a) Build the LR(0) automaton for this grammar.

   (b) Is this an LR(0) grammar? Give evidence.

   (c) Is this an SLR grammar? Give evidence.

   (d) Is this an LR(1) grammar? Give evidence.

5. Consider the following grammar:

$$
\begin{array}{lcl}
S & \rightarrow & A\ a \mid b\ A\ c \mid B\ c \mid b\ B\ a \\
A & \rightarrow & d \\
B & \rightarrow & d
\end{array}
$$

   (a) Construct the LR(1) automaton and the LR(1) parsing table.

   (b) Explain why the grammar is LR(1), but not LALR(1).