For each grammar and the given string, draw the parse tree showing how the string may be derived. Whitespace is used for clarity but is not meaningful in these exercises. Non-terminals are usually written with an initial capital letter.

1. $S \to 0S1|01$
   Generate the string `000111`

2. $S \to +SS|*SS|a$
   Generate the string `+*aaa`

3. $S->(S)S|\epsilon$
   Generate the string `( ( ) ( ) )`

4. $S->SS|(S)|\epsilon$
   Come up with at least two ways to generate the string `(())()`

5. $S \to S+S|SS|(S)|S*|a$
   Generate `(a + a) * a`

6. $S \to aSbS|bSaS|\epsilon$
   Generate `aabbab`

7. $Id \to CharRest$
   $Rest \to SymbolRest|\epsilon$
   $Symbol \to Char|Digit|Underscore$
   $Char \to a|b|c|\ldots|z$
   $Digit \to 0|1|\ldots|9$
   $Underscore \to \_$

   Generate:

   (a) `a`

   (b) `a1`

   (c) `a_1`

   (d) `ax_1`

8. Assume Id is defined as in Question 7, then add:
   $Call \to Id(Optparams)$
   $Optparams \to Params|\epsilon$
   $Params \to Params, Param|Param$
   $Param \to Id$

   Generate:

   (a) `f()`

   (b) `f(x)`

   (c) `f(x, y)`

Most of these are based on examples in Compilers: Principles, Techniques, and Tools by Aho, Lam, Sethi, and Ullman.