<assign> -> <id> = <expr>
{id> -> A | B | C
<expr> -> <expr> + <term> | <term>
<term> -> <term> * <factor> | <factor>
<factor> -> ( <expr> ) | <id>

1. Rewrite the BNF grammar above to give + precedence over * and force + to be right associative.

2. Using the grammar provided above, show a parse tree and a leftmost derivation for each of the following statements:
   
   a. A = ( A + B ) * C
   b. A = B * ( C * ( A + B ) )

3. Prove that the following grammar is ambiguous:

   <S> -> <A>
   <A> -> <A> + <A> + <A> | <id>
   <id> -> a | b | c

4. Consider the following grammar:

   <S> -> <A> a <B> b
   <A> -> <A> b | b
   <B> -> a <B> | a

   Which of the following sentences are in the language generated by this grammar?

   a. baab
   b. bbbab
   c. bbaaaaa
   d. bbaab

5. Write a grammar for the language consisting of strings that have \( n \) copies of the letter a followed by the same number of copies of the letter b, where \( n > 0 \).
6. Write an attribute grammar whose BNF basis is the grammar below but whose language rules are as follows: Data types cannot be mixed in expressions, but assignment statements need not have the same types on both sides of the assignment operator.

   <assign>  -->  <var> = <expr>
   <var>  -->  A | B | C

7. Compute the weakest precondition for each of the following:
   a. \( a = 2 \times (b - 1) - 1 \) \( \{ a > 0 \} \)
   b. \( a = 2 \times b + 1; \)
      \( b = a - 3; \)
      \( \{ b < 0 \} \)