2. (10%) (a) Is the following grammar a LL(1) grammar?
S, E, and F are nonterminals, and ``(``, ``)", ``+", and ``a" are terminals in the grammar below.

S -> E
E -> (a a a)
| F

F -> (a)

(b) Convert the grammar in (a) into LL(1) if it's not a LL(1) yet.

3. (15%) Explain the following concepts?

(a) Why is a left-recursion grammar not in LL(1)?

(b) Discuss the difference among LL(0), LL(1), and LL(2).

(c) Explain how to decide if a grammar is a LL(1) grammar.

4. (20%) If we use BNF form to write a grammar for an arithmetic expression includes ``*" (multiplication), ``#" (exponential operators), ``+" (addition), and parenthesis. We get a grammar below:

E -> E * E
E -> E # E
E -> E + E
E -> (E)
E -> Number

Assume the precedence order from the highest to the lowest is parenthesis, ``\#", ``*", ``+". The exponential operation is right associate, and all other operators are left associate.

(a) Re-Write the above grammar into an un-ambiguous grammar following the given precedence and associativity.