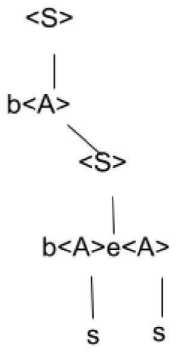
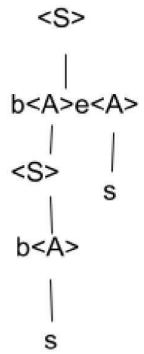
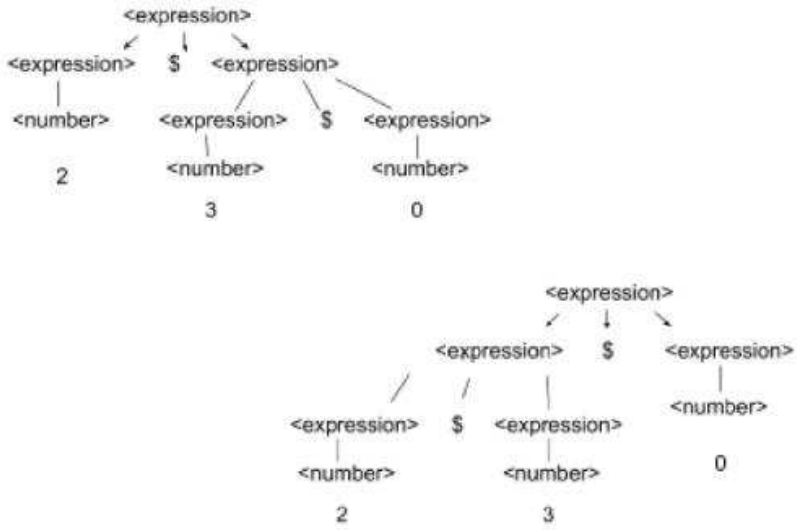
Homework #3

Name _____ Sec ____

Questions:	Answers:
1. Numbers can be real numbers, with a decimal point, as well as integers. Modify the BNF grammar below to allow reals as operands (i.e. define <number> as <integer> or <real> and then define integers and reals). <number> ::= <digit> | <number><digit> <digit> ::= 0|1|2|3|4|5|6|7|8|9	
2. Sometimes an expression can have two or more kinds of balanced parentheses. For example, Java expressions can have both round and square parentheses and both must be balanced; that is, every "(" must match ")", and every "[" must match"]". Write a grammar for strings of balanced parentheses of these two types. For example ([ ] ([( )])) is in the language but [( ]) is not.	S-> (S) | [S] | SS | ε S-> b (terminal)
3. Demonstrate that the following grammar is ambiguous. Create your own ambiguous case. (<S> is the start symbol.) <S> → b <A> <S> → b <A> e <A> <A> → <S> <A> → s	 

4. The following grammar for a fictitious operator '$' is ambiguous. <number> ::= 0\|1\|2\|3 <expression> ::= <number> <expression> ::= <expression>$<expression> Demonstrate the ambiguity by creating two parse trees for the expression 2 $ 3 $ 0.	
5. Fix the grammar in Problem #4 so that it is not ambiguous. Is your grammar left associative or right associative?	BNF Notation: <expression>::= <T> $ <B> | <T> <T>::= <T> $ <B> | <number> | <B> <B>::= <number> <number>::= 0|1|2|3 CFT Notation: E-> T$B \| T T-> T$B | N | B B-> N N->0|1|2|3 Left Associative
6. Suppose we want to implement a DDC (decimal digit calculator) compiler for the DDC language which performs arithmetic operations on integer arguments. The BNF grammar description below was written to describe the DDC language syntactically. Unfortunately, the grammar is ambiguous. <expr> ::= <term> | <expr> <op1> <expr> <term> ::= <decimal arg> | <term> <op2> <decimal arg> <decimal arg> ::= <digit> | <decimal arg> <digit> <digit> ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 <op1> ::= + | - <op2> ::= * | /	

Correct the grammar so it's unambiguous.	
7. Consider again the grammar in Problem 6. a) Give two different parse trees for 4 – 3 – 2. b) Evaluate the expression 4 – 3 – 2 according to your two parse trees. c) Which parse tree gives the right answer? d) Does the unambiguous grammar you gave as an answer to Problem 6 produce a parse tree that gives the right answer?	
