CT1 Questions

Question No	Question
1	Identify the number of tokens in the following C statement. for(i=0,j=100;i<j;i++,j--);
2	For the given infix expressions write down the corresponding postfix expressions. Assume the operator precedence and associativity is same as C programming language. DO NOT add any space at the beginning or end or in between the operands/operators. (a) a*(b+c)+(d-e+a)/b
3	For the given infix expressions write down the corresponding postfix expressions. Assume the operator precedence and associativity is same as C programming language. DO NOT add any space at the beginning or end or in between the operands/operators. (b) (a*b+c)/(d-e)
4	For the given infix expressions write down the corresponding postfix expressions. Assume the operator precedence and associativity is same as C programming language. DO NOT add any space at the beginning or end or in between the operands/operators. (c) (a*(a/(b+c)-d)+e)/c
5	Identify the correct regular expression for a floating point constant.
6	E -> TE’ E’ -> +TE’ | ε T -> FT’ T’ -> *FT’ | ε F -> (E) | id Terminal symbols: +, *, (,), id, $. Rest is non terminal symbols. $ is the end of string terminal symbol. For the above grammar, what will be First(E)? DO NOT add any space at the beginning or end or in between of your answer. No need to give curly braces. Multiple entries must be separated by comma with no space in between.

7	E -> TE’ E’ -> +TE’ | ε T -> FT’ T’ -> *FT’ | ε F -> (E) | id Terminal symbols: +, *, (,), id, $. Rest is non terminal symbols. $ is the end of string terminal symbol. For the above grammar, what will be Follow(F)? DO NOT add any space at the beginning or end or in between of your answer. No need to give curly braces. Multiple entries must be separated by comma with no space in between.
8	For the given CFG, if we draw an LR(0) parsing table then how many states will be there. Augment your grammar before you procced. S -> aAd | bBd | aBe | bAe A -> c B -> c Capital letters indicates non-terminal symbols and small letters indicates terminal symbols.
9	State whether the statement is True/False. For a given k, LL(k) is more powerful than LR(k)
10	State whether the statement is True/False. The following grammar is LALR(1). S -> Aa | bAc | dc | bda A -> d Capital letters indicates non-terminal symbols and small letters indicates terminal symbols.
11	State whether the statement is True/False. Recursive descent parser can handle left recursion.
12	State whether the statement is True/False. The following grammar is LALR(1). S -> Aa | bAc | Bc | bBa A -> d B -> d Capital letters indicates non-terminal symbols and small letters indicates terminal symbols.

13	State whether the statement is True/False. In the context of LR class of parsers, merging the states may lead to reduce-reduce conflict.
14	Construct the LALR parsing table for the following grammar S -> SS+ | SS * | a Capital letters indicates non-terminal symbols and small letters including + and * indicates terminal symbols. Note: You can type your solution in the text box OR take the photo of your workout and attach.