Assignment No:-1

1) Check whether this grammar is LL(1) or not? If yes then derive the parsing table for it

$S \rightarrow Aa \mid b$

$A \rightarrow Ac \mid Sd \mid \epsilon$

2) Design LL(1) parsing table for the following grammar.

$A \rightarrow AcB \mid cC \mid C$

$B \rightarrow bB \mid id$

$C \rightarrow CaB \mid BbB \mid B$

3) Consider the following grammar

$S' \rightarrow S\#$

$S \rightarrow ABC$

$A \rightarrow a \mid bbD$

$B \rightarrow a \mid \epsilon$

$C \rightarrow b \mid \epsilon$

$D \rightarrow c \mid \epsilon$

Where # is a terminal symbol

construct the FIRST & FOLLOW for the grammar & also design a LL(1) parsing table for the grammar.

4) Show the following grammar

$S \rightarrow (S) \mid \epsilon$

is LL(1) and also derive the parsing table for this grammar?

5) Check whether the given grammar is LL(1) or not, if yes then derive the parsing table for it.

$E \rightarrow TX$

$T \rightarrow (E) \mid int\ Y$

$X \rightarrow +E \mid \epsilon$

$Y \rightarrow *T \mid \epsilon$

Here "int" is the terminal symbol.

Assignment-No:-2

(Compiler Design)

1. Write a CFG for the regular expression
$R = 0^* 1 (0+1)^*$

2. Write a CFG for the language $L(G) = \{ww^r : w \in (0,1)^*\}$

3. Design a CFG for the language
$L = \{0^n 1^n \mid n \geq 0\} \cup \{1^n 0^n \mid n \geq 0\}$

4. Consider the following CFG

$E \rightarrow E+T \mid T$
$T \rightarrow T*F \mid F$
$F \rightarrow (E) \mid I$
$I \rightarrow a \mid b \mid c$

and remove left recursion

5. Write the code for the Recursive - decent parsing of the following grammar

expr → term rest
rest → + term rest | - term rest | ∈
term → 0 | 1 | 2 | ---- | 9

6. Consider the following grammar

$E \rightarrow TA$
$A \rightarrow +TA \mid \epsilon$
$T \rightarrow FB$
$B \rightarrow *FB \mid \epsilon$
$F \rightarrow (E) \mid id$

Write the algorithm of recursive decent parser for given grammar.

7. Consider the following grammar

$S \rightarrow 1AB \mid \epsilon$

$A \rightarrow 1AC \mid 0C$

$B \rightarrow 0S$

$C \rightarrow 1$

and test whether the grammar is LL(1) or not.

8. Calculate the First, Follow and Select the following given grammar :-

$S \rightarrow A$

$A \rightarrow aB \mid Ad$

$B \rightarrow bBC \mid f$

$C \rightarrow g$

9. The following is an LL(1) grammar for regular expression over alphabet {a, b}

$E \rightarrow TE'$

$E' \rightarrow +E \mid \epsilon$

$T \rightarrow FT'$

$T' \rightarrow T \mid \epsilon$

$F \rightarrow PF'$

$F' \rightarrow *F' \mid \epsilon$

$P \rightarrow (E) \mid a \mid b \mid \epsilon$

(i) Show the grammar is LL(1)

(ii) Construct the predictive parsing table for the grammar

(iii) Construct a recursive-descent parser for the grammar.